Operator Learning for Inference in Dynamical Systems

Thesis by
Edoardo Calvello

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy in Applied and Computational Mathematics

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2026
Defended April 7, 2026

Edoardo Calvello
ORCID: 0009-0002-6722-823X

ACKNOWLEDGEMENTS

I am immensely grateful to my advisor Andrew Stuart. His mentorship, guidance and friendship have made graduate school at Caltech an incredible experience. He has taught me everything I know about what it means to be a mathematician and an academic. His patience, encouragement to freely explore research topics ranging from analysis to machine learning, and support in traveling to conferences all across the world, have allowed me to grow as a researcher and as an individual. I will carry his example with me throughout my career.

I am grateful to Houman Owhadi, Franca Hoffmann, and Georgia Gkioxari for serving on my committee and for their guidance and friendship. I also thank Houman and Franca for welcoming me into their research groups.

This thesis would not have been possible without the numerous academic mentors I was lucky to have over the years. While at Imperial, I was fortunate to be introduced to research in applied mathematics by Grigorios Pavliotis, Nikolas Kantas, and Jonathan Weare. At Caltech, I was incredibly lucky to work with Nikola Kovachki and Ricardo Baptista; their mentorship has had a lasting impact on my work, and I am grateful to call them great friends.

I am also indebted to the many colleagues and friends I have had the privilege to work with, including Matthew Levine, Elizabeth Carlson, Matthieu Darcy, Théo Bourdais, Urbain Vaes, Bohan Chen, Eviatar Bach, Sebastian Reich, Nicholas Nelsen, and many others. I am grateful to Youssef Marzouk and Matthias Morzfeld for their hospitality at MIT and Scripps; their kindness and guidance have helped shape my research.

I am deeply grateful to all my friends, across the globe, for their unwavering support. A special thanks to those who have shared this journey with me in California: Alessio Tamborini, Attilio Lattanzi, Théo Bourdais, Ioannis Mandralis, Isabel Franco Garrido, Matthieu Darcy, Alessandro Bullitta, Giacomo Prandelli, and Valeria Cardazzi.

A tutta la mia famiglia. Mamma e papà, questo traguardo è tanto mio quanto vostro.

ABSTRACT

This dissertation advances the mathematical and algorithmic foundations of data assimilation (DA) for inference in nonlinear stochastic dynamical systems. DA is a cornerstone of modern computational science, enabling the integration of observational data with dynamical models for state estimation and uncertainty quantification. Classical algorithmic approaches, such as the ensemble Kalman filter (EnKF), rely on Gaussian approximations to estimate the conditional distribution of state given observations, which limit their fidelity in scientific and engineering applications.

The first part of this work develops a probabilistic mean-field formulation of the filtering problem, providing a rigorous description of the evolution of the conditional distribution of state given observations. This framework leads to new analytical tools to quantify the uncertainty represented by ensemble filters, culminating in the first proof of the accuracy of the EnKF as a quantifier of uncertainty for near-linear dynamics.

The second part develops theoretical and methodological advances in scientific machine learning. A continuum formulation of the attention mechanism enables the design of transformer neural operators together with the first universal approximation theorem for this class of architectures. This line of work further leads to measure neural mappings, the first neural operators defined on spaces of probability measures, opening a new avenue for Bayesian inference. With this novel methodology we introduce two new strategies for enhancing data assimilation via operator learning: (i) a fully data-driven approach involving direct approximation of conditional states via operator learning and (ii) a model-driven approach involving learning the DA analysis map via measure neural mappings. Together, these results pave the way towards theoretical and computational foundations for accurate uncertainty quantification in nonlinear, high-dimensional systems, with implications for climate science, engineering design, and digital twins.

PUBLISHED CONTENT AND CONTRIBUTIONS

Bach, Eviatar, Ricardo Baptista, Edoardo Calvello, Bohan Chen, and Andrew Stuart (2026). "Learning enhanced ensemble filters." In: *Journal of Computational Physics* 547, p. 114550. DOI: https://doi.org/10.1016/j.jcp.2025.114550.

E.C. was responsible for the theoretical analysis in Subsection 6.3 and collaborated on the development of the framework, as well as the writing.

Calvello, Edoardo, Elizabeth Carlson, Nikola Kovachki, Michael M. Manta, and Andrew M. Stuart (2026). "Operator Learning for Smoothing and Forecasting." In: *arXiv preprint arXiv:2603.20359*. URL: https://arxiv.org/pdf/2603.20359.

E. Cal. co-led the development of the theoretical framework and analysis, and was responsible for the numerical experiments and writing.

Calvello, Edoardo, Pierre Monmarché, Anadrew M. Stuart, and Urbain Vaes (2026). "Accuracy of the Ensemble Kalman Filter in the Near-Linear Setting." In: *SIAM Journal on Numerical Analysis* 64.2, pp. 391–429. DOI: https://doi.org/10.1017/S0962492924000060.

E.C. co-led the development of the theoretical framework and analysis, and was responsible for the writing.

Calvello, Edoardo, Nikola B. Kovachki, Matthew E. Levine, and Andrew M. Stuart (2025). "Continuum Attention for Neural Operators." In: *Journal of Machine Learning Research* 26.300, pp. 1–52. URL: https://jmlr.org/papers/volume26/24-0879/24-0879.pdf.

E.C. co-led the development of the theoretical framework, design of architectures, and theoretical analyis, and was responsible for most numerical experiments as well as the writing.

Calvello, Edoardo, Sebastian Reich, and Andrew M. Stuart (2025). "Ensemble Kalman Methods: A Mean Field Perspective." In: *Acta Numerica* 34, pp. 123–291. DOI: https://doi.org/10.1017/S0962492924000060.

E.C. co-led the development of the theoretical framework, analysis, and writing, and was responsible for the numerical experiments.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The field of data assimilation (DA) is focused on developing methodology for optimally integrating observational data with models of possibly stochastic dynamical systems. The central problem is one of inference under uncertainty: model error, chaotic dynamics, and observational noise each introduce significant challenges to obtaining accurate estimates of the system state. This thesis is concerned with three canonical inference problems that arise in this setting. Given partial observations of a dynamical system over a time window, the *smoothing* problem asks for the state estimate or conditional distribution of state given data, of the unobserved state trajectory given all observations within that window. The *filtering* problem is the online counterpart: it seeks to estimate the state or conditional distribution of the current state given all observations up to the present time. Finally, the *forecasting* problem aims to predict the future partially observed state beyond the observation window. Together, these three problems underpin a broad range of applications, from numerical weather prediction and climate modeling to subsurface flow and engineering control.

Traditional methodology for performing inference in dynamical systems has relied on linear-Gaussian approximations. The Kalman filter provides the exact solution when both the state dynamics and the observation operator are linear and all noise is Gaussian. For nonlinear systems, practitioners have relied on extensions such as the ensemble Kalman filter (EnKF), which propagates an ensemble of particles and approximates the filtering distribution, employing moment matching conditions with Gaussian projections. Despite their widespread adoption in the geosciences and engineering, these methods are fundamentally constrained by the Gaussian approximations inherent to their design, which limits their fidelity in the highly nonlinear, non-Gaussian regimes where they are applied. In this thesis we analyze the limitations of the ensemble Kalman filter while also exploring how machine learning can fill this gap, pursuing two complementary avenues. The first is an *end-to-end data-driven learning approach* for smoothing and forecasting, which seeks to learn a direct mapping from observations to unobserved state quantities, bypassing the need for an explicit dynamical model. The second is a *model-driven approach* for filtering, in which the model-driven forecast is retained but the analysis

step, the correction of the forecast by the incoming observation, is parametrized by a learnable operator. Both paradigms are developed with the goal of overcoming the linear-Gaussian constraints inherent to the design of classical ensemble methods and to enable accurate inference in these settings.

The data-driven methodology developed in this thesis rests on the paradigm of *operator learning*, the task of learning mappings between infinite-dimensional function spaces from finite collections of input-output pairs. A key desideratum for operator learning architectures in scientific computing is *discretization invariance*: the learned operator should be independent of the particular discretization of the underlying continuum problem, permitting training and deployment at arbitrary resolutions. Transformers, the architectural backbone of modern large language models, have shown remarkable capacity for modeling long-range correlations, and their attention mechanism admits a natural continuum generalization. Starting from a continuum formulation of attention, we introduce transformer neural operators, attention-based architectures mapping between infinite-dimensional spaces of functions that are inherently discretization invariant. A central theoretical contribution of this thesis is the first universal approximation theorem for this class of architectures: we prove that transformer neural operators can approximate, to arbitrary accuracy, a broad class of nonlinear operators between Banach spaces. Universal approximation is a necessary theoretical condition for any learning paradigm intended for deployment in scientific domains, as it guarantees the existence of network parameters achieving the desired operator approximation.

In the data assimilation context, transformer neural operators can be deployed in several distinct roles. They may serve as cheap-to-evaluate surrogate models for the forward dynamics, replacing expensive numerical solvers during inference. They may be used to learn the smoothing or forecasting map directly from data, in a fully data-driven fashion. Or they may parametrize the analysis step of a model-driven ensemble filter, using a learned nonlinear operator defined on spaces of probability measures. Indeed, another main contribution is the introduction of the first neural operator defined on the space of probability measures. This thesis develops the theoretical and algorithmic underpinnings of each of these approaches, establishing rigorous foundations and demonstrating practical performance on a range of nonlinear dynamical systems.

1.1 Outline and contributions

The thesis is organized in two parts. Part I, consisting of Chapter 2 and Chapter 3, is focused on the development of a probabilistic mean-field formulation of the filtering problem, and its application to the analysis of ensemble Kalman filters. Part II, consisting of Chapter 4, Chapter 5, and Chapter 6, is focused on the development of transformer-based operator learning methodology for nonlinear data assimilation. In this thesis we make the following main contributions.

(C1) We develop a probabilistic mean-field formulation of the filtering problem, which provides a rigorous description of the evolution of the conditional distribution of state given observations. This framework leads to new analytical tools to quantify the uncertainty represented by ensemble filters.

(C2) We introduce the first proof of the accuracy of the EnKF as a quantifier of uncertainty for near-linear state dynamics.

(C3) We develop a continuum formulation of the attention mechanism, which enables the design of transformer neural operators together with the first universal approximation theorem for this class of architectures.

(C4) We introduce a fully data-driven approach to smoothing and forecasting employing neural operators. We formulate and prove the first universal approximation theorem for the smoothing and forecasting maps, and demonstrate the practical performance of this approach on a range of nonlinear dynamical systems.

(C5) We introduce measure neural mappings, the first neural operators defined on spaces of probability measures, opening a new avenue for Bayesian inference. We use this methodology to learn the DA analysis map, and enhance traditional model-driven ensemble filters.

In Chapter 2 we work in the context of the state observation model, and introduce the general form of the state estimation problem, and the associated filtering problem of recovering the conditional distribution of state given observations. We develop a mean-field formulation of the filtering problem, which provides a rigorous description of the evolution of the conditional distribution. We use this perspective to introduce the ensemble Kalman methodology, hence Contribution (C1). In Chapter 3 we use the mean-field formulation and the tools developed in Chapter 2 to

introduce a novel analytical framework to quantify the uncertainty represented by ensemble Kalman filters, and use this framework to prove the first result on the accuracy of the EnKF as a quantifier of uncertainty for near-linear state dynamics, thus addressing Contribution (C2). This analysis and the contruction of ensemble Kalman filters using linear-Gaussian approximations suggests an inherent limitation of the EnKF in quantifying uncertainty for highly nonlinear dynamics, which motivates the development learning based methodology that overcomes this constraint. In Chapter 4 we introduce the transformer neural opperator architecture and universal approximation theory for this class of methods. This chapter addresses Contribution (C3) and underpins the algorithmic developments of the subsequent chapters. In Chapter 5 we introduce a fully data-driven approach to smoothing and forecasting employing neural operators. Leveraging the universal approximation theory developed in Chapter 4, we prove the first universal approximation result for the smoothing and forecasting maps, thus addressing Contribution (C4). We also demonstrate the practical performance of a transformer based approach on a range of nonlinear dynamical systems. We conclude in Chapter 6 by introducing a transformer operator mapping between spaces of probability measures. We use this approach to learn the analysis map in an ensemble filtering framework, beating traditional model-driven approaches on a range of experimental settings, hence Contribution (C5). This represent the first step towards building learning-based algorithms for accurate and scalable probabilistic conditioning.

1.2 Notation

Throughout we denote the positive integers and non-negative integers respectively by $\mathbb{N} = \{1, 2, \cdots\}$ and $\mathbb{Z}^+ = \{0, 1, 2, \cdots\}$, and the notation $\mathbb{R} = (-\infty, \infty)$ and $\mathbb{R}^+ = [0, \infty)$ for the reals and the non-negative reals. For a set D, we denote by $\overline{D}$ the closure of the set, i.e., the union of the set itself and the set of all its limit points. We will occasionally write $[\![1, n]\!] \subset \mathbb{N}$ to denote the set $\{1, \ldots, n\}$. We use $\mathcal{F}(A)$ to denote the set of all nonempty finite subsets of a set A.

1.2.1 Vectors, Matrices, Euclidean Spaces

In the following let A, B, C be symmetric matrices. We write $A \succ B$ when $A - B$ is positive definite and $A \succeq B$ when $A - B$ is positive semi-definite. We will also write $A \prec B$ when $B - A \succ 0$ and $A \preceq B$ when $B - A \succeq 0$. We let $\langle \cdot, \cdot \rangle, |\cdot|$ denote the Euclidean inner-product and norm, noting that $|v|^2 = \langle v, v \rangle$. We write $\langle \cdot, \cdot \rangle_{\mathbb{R}^d}, |\cdot|_{\mathbb{R}^d}$ if we want to make explicit the dimension of the space on which the Euclidean inner

product is computed. We also use $|\cdot|$ to denote the resulting induced norm on matrices. We may also use $|\cdot|$ to denote the cardinality of a set, but the distinction will be clear from context. We will also denote by $|\cdot|_1$ the norm on the vector space ℓ^1. For $C \succ 0$ (and therefore a covariance matrix) we define $\langle \cdot, \cdot \rangle_C$ and $|\cdot|_C$, the covariance-weighted Euclidean inner-product and norm, by $\langle u, v \rangle_C = \left\langle u, C^{-1} v \right\rangle$ and $|v|_C^2 = \langle v, v \rangle_C$. We use : to denote the Frobenius inner-product between matrices, and $|\cdot|_{\mathrm{F}}$ the induced norm on matrices.

1.2.2 Normed Spaces and Operators

We denote by $C(D; \mathbb{R}^d)$ the infinite dimensional Banach space of continuous functions mapping the set D to the d-dimensional vector space $\mathbb{R}^d$. The space is endowed with the supremum norm $\|\bullet\|_\infty$. Similarly, we denote by $L^2(D; \mathbb{R}^d)$ the infinite dimensional space of square integrable functions mapping D to $\mathbb{R}^d$. The space is endowed with the L^2 inner product, which induces the norm on the space. We will sometimes use the shorthand notation $C(D)$ and $L^2(D)$ to denote continuous functions and square integrable functions defined on the domain D, respectively, when the image space is irrelevant. Similarly, we sometimes drop the domain notation completely, when it is clear from context. For integers $s \geqslant 0$ we denote by C^s the space of continuously differentiable functions up to order s. Let C^∞ denote the infinite dimensional real vector space of all infinitely differentiable functions. For any function $f \in C^s\left(\mathbb{R}^{d_1}; \mathbb{R}^{d_2}\right)$ and any $0 \leqslant k \leqslant s$, we define by $\mathsf{D}^k f(v)$ the kth derivative at $v \in \mathbb{R}^{d_1}$ viewed as the linear operator $\mathsf{L}\left(\bigotimes_{i=1}^k \mathbb{R}^{d_1}; \mathbb{R}^{d_2}\right)$. We endow the space C^s with the norm $\|\bullet\|_{C^s}$ defined as

$$\|f\|_{C^s} = \sum_{j=0}^{s} \|\mathsf{D}^j f\|_\infty, \tag{1.1}$$

for any function $f \in C^s$. We also use ∇, $\nabla\cdot$ and Δ to denote the gradient, divergence and Laplacian operations respectively. Given $s \geqslant 1$, for any function $f \in C^s\left(\mathbb{R}^{d_1}; \mathbb{R}^{d_2}\right)$ and a vector field $w \in C^{s-1}\left(\mathbb{R}^{d_1}; \mathbb{R}^{d_1}\right)$, we also define $\mathcal{L}_w f(v)$, the Lie derivative of f along w, as

$$\mathcal{L}_w f(v) = \mathsf{D} f(v) w(v), \qquad v \in \mathbb{R}^{d_1}.$$

Note that $\mathcal{L}_w f \in C^{s-1}\left(\mathbb{R}^{d_1}; \mathbb{R}^{d_2}\right)$. For $k \leqslant s$ we also denote by $\mathcal{L}_w^k f$ the Lie derivative of f along w of order k, defined by $k-$fold application of $\mathcal{L}_w$. We will occasionally drop the vector field from the notation when it is clear from the context.

For $p \in [1, \infty)$, we denote by $W^{s,p}$ the Sobolev space of functions possessing weak derivatives up to order s of finite L^p norm. We use the notation H^s to denote $W^{s,2}$.

Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and $r \geqslant 0$, we let $B_{L^\infty}(f, r)$ denote the L^∞ ball of radius r centered at f. Similarly, for functions f, g and $r \geqslant 0$, we denote by $B_{L^\infty}\big((f, g), r\big)$ the L^∞ ball of radius r centered at (f, g), on the product space, i.e. the set of all $(\mathfrak{f}, \mathfrak{g}) \in L^\infty(\mathbb{R}^n) \times L^\infty(\mathbb{R}^d)$ satisfying

$$\|\mathfrak{f} - f\|_{L^\infty(\mathbb{R}^n)} \leqslant r, \quad \|\mathfrak{g} - g\|_{L^\infty(\mathbb{R}^d)} \leqslant r.$$

We write $|\bullet|_{C^{0,1}}$ for the $C^{0,1}$ semi-norm, referring in particular to the Lipschitz constant.

Throughout, general vector spaces will be written using a calligraphic font $\mathcal{U}$. We denote by $\mathsf{L}(\mathcal{U}; \mathcal{V})$ the infinite-dimensional space of linear operators mapping the vector space $\mathcal{U}$ to the vector space $\mathcal{V}$. We use $\mathcal{F}$ to denote the Fourier transform, while $\mathcal{F}^{-1}$ will denote the inverse Fourier transform. Throughout the thesis we will often distinguish operators acting on infinite-dimensional spaces by using the mathsf font. For example, we distinguish operators acting on finite dimensional spaces, such as Q, and operators acting on infinite dimensional spaces, such as Q.

1.2.3 Probability Spaces

We denote by $\mathcal{P}(\Omega)$ the space of probability measures defined on set Ω. We use $\mathbb{E}$ and $\mathbb{P}$ to denote expectation and probability under the prevailing probability measure; if we wish to make clear that measure π is the prevailing probability measure then we write $\mathbb{E}_\pi$ and $\mathbb{P}_\pi$. We denote by $\mathrm{Law}(v)$ the law of a random variable v. We apply the notation $\perp\!\!\!\perp$ to denote independence of two random variables.

We let δ_u denote the Dirac mass on $\mathbb{R}^d$, centered at point $u \in \mathbb{R}^d$. The notation $\mathsf{N}(m, C)$ denotes the distribution of a Gaussian random variable with mean m and covariance C. We let $\mathcal{G} = \mathcal{G}(\mathbb{R}^d)$ denote the set of Gaussian probability measures on $\mathbb{R}^d$ (including indefinite covariances and hence all Dirac masses). We also use the notation $\mathsf{N}(m, C)$ in the infinite dimensional context do denote a Gaussian measure with covariance operator C; whether we are in the finite dimensional setting or the infinite dimensional one will be clear from context. We also use the notation $\mathsf{unif}(D)$ to denote a uniform distribution defined over the domain D.

We will denote by $\mathcal{P}^p(\mathbb{R}^n)$ the set of probability measures over $\mathbb{R}^n$ with finite moments up to order p. With the exception of empirical measures formed from finite ensembles, this manuscript primarily deals with probability measures that have a Lebesgue density, because of the assumptions concerning the noise structure in the dynamics model and the data acquisition model. In this setting we occasionally abuse

notation by using the same symbols for probability measures and their densities. For $\mu \in \mathcal{P}(\mathbb{R}^n)$, the notation $\mu(x)$ for $x \in \mathbb{R}^n$ refers to the Lebesgue density of μ evaluated at x, while $\mu[f]$ for a function $f\colon \mathbb{R}^n \to \mathbb{R}$ will be notation for $\int_{\mathbb{R}^n} f \,\mathrm{d}\mu$. For function $f : \mathbb{R}^n \to \mathbb{R}^m$ and measure $\mu \in \mathcal{P}(\mathbb{R}^n)$, we denote by $f_\sharp\mu \in \mathcal{P}(\mathbb{R}^m)$ the measure given by the pushforward, under f, of μ.

For a probability measure $\mu \in \mathcal{P}(\mathbb{R}^n)$, the notation $\mathcal{M}_q(\mu)$ denotes the qth polynomial moment under the measure μ, defined as

$$\mathcal{M}_q(\mu) \coloneqq \int_{\mathbb{R}^n} |x|^q \mu(\mathrm{d}x). \tag{1.2}$$

We denote by $\mathcal{M}(\mu)$ and $C(\mu)$ the mean and covariance under μ, respectively:

$$\mathcal{M}(\mu) = \mu[x], \qquad C(\mu) = \mu\Big[\big(x - \mathcal{M}(\mu)\big) \otimes \big(x - \mathcal{M}(\mu)\big)\Big].$$

We use $\mathcal{P}_R(\mathbb{R}^n)$ with $R \geqslant 1$ to denote the subset of $\mathcal{P}(\mathbb{R}^n)$ of probability measures with mean and covariance satisfying the bounds

$$|\mathcal{M}(\mu)| \leqslant R, \qquad \frac{1}{R^2} I_n \preccurlyeq C(\mu) \preccurlyeq R^2 I_n, \tag{1.3}$$

where I_n denotes the identity matrix in $\mathbb{R}^{n\times n}$, and $\preccurlyeq$ is the partial ordering defined by the convex cone of positive semi-definite matrices. Additionally, $\mathcal{G}_R(\mathbb{R}^n)$ is the subset of $\mathcal{G}(\mathbb{R}^n)$ of probability measures satisfying (1.3).

For $\pi \in \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ defined on the joint space of state and data associated with random variable $(u, y) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_y}$ we denote by $\mathcal{M}^u(\pi)$, $\mathcal{M}^y(\pi)$ the means of the marginal distributions, and by $C^{uu}(\pi)$, $C^{uy}(\pi)$ and $C^{yy}(\pi)$ the blocks of the covariance matrix $C(\pi)$. In particular, we write

$$\mathcal{M}(\pi) = \begin{pmatrix} \mathcal{M}^u(\pi) \\ \mathcal{M}^y(\pi) \end{pmatrix}, \qquad C(\pi) = \begin{pmatrix} C^{uu}(\pi) & C^{uy}(\pi) \\ C^{uy}(\pi)^\mathsf{T} & C^{yy}(\pi) \end{pmatrix}. \tag{1.4}$$

For $h\colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ we also define $C^{hh}(\pi)$ to be the covariance of vector $h(u)$, for u distributed according to the marginal of π on u, and $C^{uh}(\pi)$ to be the covariance between u and $h(u)$. Given this notation, we also introduce the set $\mathcal{P}^2_{\succ 0}(\mathbb{R}^{d_u \times d_y})$ of probability measures π with finite second moment satisfying $C^{yy}(\pi) \succ 0$.

We introduce the Gaussian projection operator $\mathsf{G}\colon \mathcal{P}^2(\mathbb{R}^n) \to \mathcal{G}(\mathbb{R}^n)$ defined by $\mathsf{G}\mu = \mathsf{N}\big(\mathcal{M}(\mu), C(\mu)\big)$. We refer to G as a projection because $\mathsf{G}\mu$ is the Gaussian distribution closest to μ with respect to $\mathrm{KL}(\mu\|\bullet)$ (C. M. Bishop, 2006), where $\mathrm{KL}(\mu\|\pi)$ denotes the Kullback–Leibler (KL) divergence of μ from π. We note that G defines a nonlinear mapping. Throughout Chapter 3 we make use of the following weighted total variation distance as in J. A. Carrillo et al., 2024:

Definition 1.2.1. *Let* $g\colon \mathbb{R}^n \to [1,\infty)$ *define the function* $g(v) := 1 + |v|^2$. *The weighted total variation metric* $d_g\colon \mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^n) \to \mathbb{R}$ *is defined by*

$$d_g(\mu_1,\mu_2) = \sup_{|f|\leqslant g} \left|\mu_1[f] - \mu_2[f]\right|,$$

where the supremum is taken over all functions $f\colon \mathbb{R}^n \to \mathbb{R}$ *bounded from above by g pointwise and in absolute value.*

We denote by $T_\#\pi$ the pushforward measure of π by the map T: if $u \sim \pi$ then $T(u) \sim T_\#\pi$. In Chapter 6 if an operator B on probability measures is defined as the pushforward by a map, we denote the associated transport map with the font $\mathfrak{mathfrak}$, i.e. $\mathsf{B}(\cdot) = \mathfrak{B}_\sharp(\cdot)$ to make this connection clear.

1.2.4 Data Assimilation

In the context of DA, we use v to denote the system states and y to denote the observations. The superscript $\dagger$ indicates true values and the subscript $j \in \mathbb{Z}^+$ denotes the time step. For ensemble methods, we use the parenthesized superscript (n) for the ensemble index.

Part I

Ensemble Data Assimilation

Chapter 2

ENSEMBLE KALMAN METHODS: A MEAN FIELD PERSPECTIVE

2.1 Introduction

The ensemble Kalman methodology comprises an innovative and flexible set of tools which can be used for both state estimation in dynamical systems and parameter estimation for generic inverse problems. It has primarily been developed by practitioners in the geophysical sciences, with notable impact on the fields of oceanography, oil reservoir simulation and weather forecasting. Despite its widespread adoption in the geosciences over several decades, firm theoretical foundations are only recently starting to emerge; the methodology is hard to analyze. The purpose of this chapter is twofold: a) to introduce a mathematical framework for the analysis of ensemble Kalman methods, describing what is known and highlighting the many open mathematical challenges in the field; and b) to provide a literature survey which bridges the domain-specific development of the methodology with emerging mathematical analyses. In so doing we will also highlight the flexibility of the methodology for use in widespread applications, beyond its historical development in the geosciences.

The novel perspective which underlies all of this material is the derivation of ensemble Kalman methods as particle approximations of carefully designed mean field models. The relationship of these mean field models to exact transport models, for Gaussian problems, serves to motivate their form.

In Subsection 2.1.1 we overview the history of ensemble Kalman methods. Subsection 2.1.2 describes the organization of the chapter.

2.1.1 Historical Context

The Kalman filter (KF) is arguably the first setting in which the systematic integration of observational data with a dynamical system was considered, leading to both discrete time (Kalman, 1960) and continuous time (Kalman and Bucy, 1961) formulations; see Welch, G. Bishop, et al. (1995) for an overview. The Kalman filter applies only in the setting of linear Gaussian dynamics and observations. In this setting it computes the distribution of the state of the dynamical system, given observations, exactly; this Bayesian perspective on the filter was highlighted in Ho

and R. Lee (1964) subsequent to the original derivation in Kalman (1960) which proceeded by computing the best linear predictor of the state, given data. The extended Kalman filter (historically denoted EKF; however the acronym ExKF is also used and is useful) was introduced in order to extend Kalman's ideas to nonlinear problems; see the texts Andrew H Jazwinski (2007) and B. D. Anderson and Moore (2012) for overviews. The extended Kalman approach is based on a linearization approximation; it hence fails to exactly compute the distribution of the state of the dynamical system, given observations, in general. Furthermore it requires propagation of covariance matrices which can be very large for applications arising in the geosciences (Michael Ghil et al., 1981).

The ensemble Kalman filter (EnKF) was introduced in the celebrated paper Evensen (1994) which made the consequential observation that, if an *ensemble* of state estimators is employed then it can also be used to make an approximation of the covariance. In geosciences applications this circumvents the computation of large covariances, replacing them instead with low rank approximations, with rank determined by the number of ensemble members. The original paper developed the idea in the context of ocean models, but was rapidly and concurrently developed in a variety of geoscience application domains (Van Leeuwen and Evensen, 1996; G. Burgers, van Leeuwen, and G. Evensen, 1998; P.L. Houtekamer and M.L. Mitchell, 1998); the paper Van Leeuwen (2020b) provides a historical overview. These methods are sometimes referred as the *stochastic EnKF*: they require simulation of random variables to implement. A different class of ensemble methods, known collectively as *ensemble square root filters*, was subsequently developed (Jeffrey L Anderson, 2001; J.S. Whitaker and T.M. Hamill, 2002; C.H. Bishop, B.J. Etherton, and S.J. Majumdar, 2001; B.R. Hunt, E.J. Kostelich, and I. Szunyogh, 2007; M.K. Tippett et al., 2003; P. Sakov and P.R. Oke, 2008); these methods are a form of *deterministic EnKF*: they do not require simulation of random variables to implement.

Central to our mathematical presentation of Kalman-based methods is the adoption of mean field and transport perspectives on the subject. The incorporation of data within filtering constitutes (possibly approximate) application of Bayes' theorem; the papers Daum, Huang, and Noushin (2010), Reich (2011), El Moselhy and Y. M. Marzouk (2012), and the survey C. Cotter and Reich (2013) introduce novel approaches to Bayesian inversion, rooted in transport and mean field models. While El Moselhy and Y. M. Marzouk (2012), Spantini, Baptista, and Youssef Marzouk (2022a) propose a direct numerical approximation of the underlying optimal

transport problem, Daum, Huang, and Noushin (2010) and Reich (2011) pursue a homotopy approach. We note that the homotopy approach is closely related to iterative implementations of the EnKF, as first considered by G. Li and Albert Coburn Reynolds (2009), Gu and Dean S Oliver (2007), and P. Sakov, D. S. Oliver, and L. Bertino (2012); these homotopy approaches lead to continuous time formulations of the EnKF in the limit of infinitely many iterations, as first considered by Bergemann and Reich (2010a) and Bergemann and Reich (2010b). Directly starting from the continuous time filtering perspective, mean field models have been introduced independently in Crisan and Xiong (2010) and T. Yang, Prashant G. Mehta, and Meyn (2013).

The key connection between mean field models and ensemble methods is that the latter can be derived as particle approximations of the mean field limit; this viewpoint will play a guiding role in our presentation of the subject of ensemble methods in this chapter. In this context it is notable that the field of optimization, which is linked to Bayesian sampling through MAP estimation (Kaipio and Somersalo, 2006), has also seen recent development using mean field models — see José A Carrillo et al. (2018) for an overview and unifying mathematical framework.

The methods covered in this survey provide only approximate solutions to the underlying filtering, inference and/or optimization problem, in the mean field limit. The approximations invoked are based on assuming linear Gaussian structure, where the mean field models are exact, but applying the resulting methodology outside this regime. In the context of the optimization and Bayesian approaches to inversion, affine invariant algorithms (introduced in Goodman and Weare (2010)) play an important conceptual role in understanding the power of ensemble Kalman methods: affine invariance can be used to show universal convergence rates for linear Gaussian problems. Empirically the affine invariance confers advantages when ensemble Kalman methods are applied beyond the linear Gaussian setting. In practice this benefit must be weighed against the error resulting from using ensemble Kalman methods outside the linear Gaussian setting.

Alternative methods, such as sequential Monte Carlo, can be designed to be consistent with the underlying nonlinear filtering problem, and do not rely on being exact only for linear Gaussian problems. The books Doucet, De Freitas, and Gordon (2001) and Chopin and Papaspiliopoulos (2020) overview use of sequential Monte Carlo methods for general discrete time filtering and inference problems; the papers Pierre Del Moral (1997) and Pierre Del Moral and Guionnet (2001) prove

convergence of sequential Monte Carlo methods, including in some specific cases over long time horizons. However, sequential Monte Carlo methods suffer from a curse of dimensionality and are presently not directly applicable to high dimensional problems as arising, for example, from geophysical applications. This issue with the curse of dimensionality provides an important motivation for the ensemble methods covered in this survey. See Snyder et al. (2008), Bickel, B. Li, and Bengtsson (2008), Rebeschini and Van Handel (2015) and Agapiou et al. (2017) for detailed discussion of these issues. Related issues also arise for Monte Carlo Markov chain (MCMC) when studying Bayesian inverse problems (Kaipio and Somersalo, 2006). See Hairer, Andrew M Stuart, and Vollmer (2014) for an analysis of the degeneration of performance of standard MCMC methods in high dimensions, as well as analysis of special MCMC methods tailored to infinite dimensional problems; the subject is overviewed in S. L. Cotter et al. (2013).

This completes our chronological overview of the historical context for the development of ensemble Kalman methods, and the specific mathematical context which will be our focus. Each of the four subsequent sections concludes with a bibliographic subsection in which a deeper literature review is given.

2.1.2 Overview

Section 2.2 is devoted to the problem of state estimation for discrete time (possibly stochastic) dynamical systems, given noisy observations. We formulate the problem from the perspectives of both control theory and probability, and we provide a unifying approach to algorithms for these problems; the approach rests on transport of measures and mean field stochastic dynamical systems. Ensemble Kalman methods are then derived as particle approximations of the mean field models.

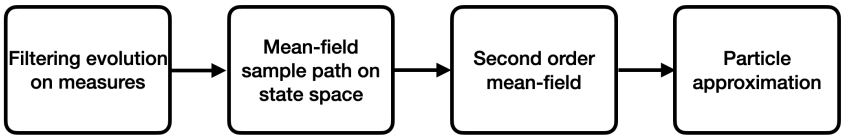


Figure 2.1: Organizational flow employed in Section 2.2 concerning state estimation in discrete time.

Section 2.2 is organized according to the flow of ideas displayed in Figure Figure 2.1. Indeed, starting from a probabilistic perspective, that describes the evolution of the filtering distribution, it is possible to formulate mean field maps whose sample paths on state space are equal in law to the filtering evolution. Since it is typically not tractable to identify these maps explicitly, it is of interest to determine mean

field maps whose sample paths are only *approximately* equal in law to the filtering evolution. This leads to the idea of second order mean field models whose sample paths have law with first and second order moments which match those of a Gaussian approximation of the Bayesian data incorporation step. Particle approximations of these second order mean field maps are then used to derive implementable numerical algorithms which are the ensemble Kalman methods implemented in practice.

2.2 State Estimation: Discrete Time

In Subsection 2.2.1 we provide the set-up for the problem of state estimation in discrete time. Subsection 2.2.2 introduces an algorithm for this problem based on a control theoretic perspective. Subsection 2.2.3 describes the Bayesian probabilistic perspective; in this subsection algorithms are not presented but foundations for the creation of algorithms are laid through the decomposition of the filtering cycle into an iteration which alternates prediction and the assimilation of data. In Subsection 2.2.4 we introduce the important idea of Gaussian projection, and the resulting Gaussian projected filter. Subsection 2.2.5 introduces various mean field dynamical systems which approximate the filtering cycle; a unifying transport map perspective is adopted. This leads, in Subsection 2.2.6, to definitions of the ensemble Kalman filter, the ensemble adjustment filter and the ensemble transform filter, all derived as particle approximations of specific mean field models. We conclude with bibliographic notes in Subsection 2.2.7 in which we review relevant literature, and include discussion of a variety of other algorithms for state estimation, and relate them to the perspective we adopt here.

We emphasize that all the mean field methods we derive in this section for the solution of filtering problems are exact only in the linear Gaussian setting. They may be implemented beyond this setting, however, and are empirically found to work well for many problems. Theory to justify this observation is very much needed. Our exposition provides a framework for such a theory.

2.2.1 Set-Up

The objective of *sequential data assimilation* is to iteratively update the state of a (possibly stochastic) dynamical system based on observations and knowledge of the dynamical and observational processes; we refer to this as *state estimation*. The typical setting is one in which the initial condition is uncertain, but this uncertainty is compensated for by (typically noisy) observations of a (possibly nonlinear and indirect) function of the state. These observations often live in a space of lower

dimension than the dimension of the state space meaning that the goal of state estimation goes beyond denoising, and into the realm of control theoretic considerations such as observability.

A useful starting point is to consider a stochastic dynamical system in which the evolution of the state, and the relationship between the observations (which we also refer to as data) and the state, are defined, respectively, by the equations

$$v_{n+1} = \Psi(v_n) + \xi_n, \tag{2.1a}$$

$$y_{n+1} = h(v_{n+1}) + \eta_{n+1}, \tag{2.1b}$$

taken to hold for all $n \in \mathbb{Z}^+$. We assume that, for each fixed $n \in \mathbb{Z}^+$, the *state* $v_n \in \mathbb{R}^{d_v}$ and the *observations* $y_n \in \mathbb{R}^{d_y}$. The maps $\Psi(\cdot)$ and $h(\cdot)$ describe the systematic, deterministic components of the dynamics and observation processes, and are assumed to be known measurable functions (with respect to the Borel algebra), bounded on compact sets. The initial condition for v_0 is assumed random and the systematic components of the model are subjected to mean zero noise, ξ_n and η_{n+1}. To be concrete we assume v_0, $\{\xi_n\}_{n\in\mathbb{Z}^+}$ and $\{\eta_n\}_{n\in\mathbb{N}}$ are mutually independent Gaussians defined by

$$v_0 \sim \mathsf{N}(m_0, C_0), \quad \xi_n \sim \mathsf{N}(0, \Sigma) \text{ i.i.d.}, \quad \eta_n \sim \mathsf{N}(0, \Gamma) \text{ i.i.d.} . \tag{2.2}$$

In practice we will have available to us the observation coordinates of a specific true realization of the random dynamical system (2.1), from which we wish to recover the specific true realization of the state that gave rise to these observations. We denote the true realizations of the state of the system by sequence $\{v_n^\dagger\}_{n\in\mathbb{Z}^+}$, and the observed data by $\{y_n^\dagger\}_{n\in\mathbb{N}}$. These are generated by $v_0^\dagger$, $\{\xi_n^\dagger\}_{n\in\mathbb{Z}^+}$ and $\{\eta_n^\dagger\}_{n\in\mathbb{N}}$, specific realizations of the initial condition and state and observational noise from the distribution defined by (2.2).

We let $Y_n^\dagger = \{y_\ell^\dagger\}_{1\leqslant\ell\leqslant n}$. Our objective is to estimate the state $v_n^\dagger$ at time n from data $Y_n^\dagger$. Specifically, it is natural to think about this design objective in two different ways:

- Objective 1: design an algorithm producing output v_n from $Y_n^\dagger$ so that $\{v_n\}_{n\in\mathbb{Z}^+}$ estimates $\{v_n^\dagger\}_{n\in\mathbb{Z}^+}$, the true state generated by (2.1a).
- Objective 2: design an algorithm which estimates the distribution of random variable $v_n|Y_n^\dagger$.

In both cases we are interested in *Markovian* formulations which update the estimate v_n, or the distribution $v_n|Y_n^\dagger$, sequentially as the data is acquired. In the next two subsections we describe control theoretic and probabilistic approaches to this problem which, respectively, provide the basis for algorithms addressing Objectives 1 and 2. We note that fulfilling Objective 2 immediately implies resolution of Objective 1, for example by taking the mean of $v_n|Y_n^\dagger$ as state estimator; but the reverse is not typically true. However, Objective 1 is easier to address and is especially relevant when noise levels are small; furthermore, its failure modes serve to motivate approaches which are used to address Objective 2.

2.2.2 Control Theory Perspective

A very natural idea from control theory underlies ensemble Kalman filtering and is encapsulated in the following algorithmic approach. This way of attacking state estimation is most appropriate when $|C_0|$, $|\Gamma|$ and $|\Sigma|$ are small so that the state and observations are close to deterministic. To understand this setting we will first study algorithms in which the covariances Γ and Σ are set to zero, and then return to the inclusion of noise later.

The algorithmic idea works as follows: from current state estimate v_n, given $Y_n^\dagger$, *predict* the outcome of the model and data, denoted by $(\widehat{v}_{n+1}, \widehat{h}_{n+1})$, using the update equations (2.1), but ignoring the noise; then *correct* the state estimate by nudging the prediction using the mismatch between observed and predicted data $(y_{n+1}^\dagger, \widehat{h}_{n+1})$. This results in a deterministic map $v_n \mapsto v_{n+1}$, assumed to hold for all $n \in \mathbb{Z}^+$, of the following form:

$$\widehat{v}_{n+1} = \Psi(v_n), \tag{2.3a}$$

$$\widehat{h}_{n+1} = h(\widehat{v}_{n+1}), \tag{2.3b}$$

$$v_{n+1} = \widehat{v}_{n+1} + K(y_{n+1}^\dagger - \widehat{h}_{n+1}). \tag{2.3c}$$

Equations (2.3a, 2.3b) create predicted state and data from current estimate v_n of the state. The difference between the predicted data $\widehat{h}_{n+1}$ and the true data $y_{n+1}^\dagger$, the latter found from a fixed realization of (2.1), is then used to correct the predicted state resulting in (2.3c). We can write the algorithm compactly in the form

$$v_{n+1} = \Psi(v_n) + K\Big(y_{n+1}^\dagger - h\big(\Psi(v_n)\big)\Big). \tag{2.4}$$

Choice of the *gain matrix* K completes definition of an algorithm which we refer to as 3DVAR.

Remark 2.2.1. *The nomenclature "3DVAR" was coined in the geophysics community, and stands for three dimensional variational data assimilation. This is natural for algorithms which incorporate spatially distributed data sequentially in time, in the context where the state variable v varies across three spatial coordinates. In our setting the state variable v is not required to have any spatial structure, but the control formulation (2.4) reproduces the 3DVAR algorithm from the geophysics community when it does. Hence we still refer to it as 3DVAR. We also note that the method is perhaps more properly termed Cycled 3DVAR — "3DVAR" refers to the assimilation of data at each observation time, and "Cycled" to doing this sequentially in time as successive observations are acquired. The variant on 3DVAR which uses data distributed over several times steps is known as 4DVAR; see bibliography Subsection 2.2.7.*

We note that the difference between the observed value $y^\dagger_{n+1}$ *and its estimator* $\widehat{h}_{n+1} = h\big(\Psi(v_n)\big)$ *is often referred to as the* innovation*, and for this we introduce the notation*

$$\mathfrak{I}_n = y^\dagger_{n+1} - \widehat{h}_{n+1}. \tag{2.5}$$

■

To illustrate 3DVAR we consider the linear setting:

Example 2.2.2. *Assume that, for matrices* M, H *of appropriate dimensions,*

$$\Psi(\cdot) := M\cdot, \quad h(\cdot) = H\cdot \tag{2.6}$$

and consider the setting in which there is no noise present. The 3DVAR algorithm (2.4) *is appropriate in this setting and reduces to*

$$v_{n+1} = Mv_n + K(y^\dagger_{n+1} - HMv_n). \tag{2.7}$$

Since there is no noise present it follows that $y^\dagger_{n+1} = Hv^\dagger_{n+1} = HMv^\dagger_n$ *and hence that*

$$v^\dagger_{n+1} = Mv^\dagger_n + K(y^\dagger_{n+1} - HMv^\dagger_n). \tag{2.8}$$

Subtracting (2.8) *from* (2.7) *and defining* $e_n = v_n - v^\dagger_n$ *we find that*

$$e_{n+1} = (I - KH)Me_n.$$

Since the goal of data assimilation is to recover the true state from partial observations we wish to drive e_n *to zero as* $n \to \infty$*. Thus a key question, for given forward*

dynamical model M and observation operator H, is whether it is possible to design K to ensure that the spectrum of $(I - KH)M$ is inside the unit circle. Such questions are at the heart of the theory of linear control. Thus the subject of control theory is fundamentally aligned with Objective 1. ■

In the following example we illustrate similar ideas to those from the preceding, linear, example but within the nonlinear context. In subsequent subsections we will show how the ideas can be generalized to an adaptive gain matrix K_n, leading us to the ensemble Kalman methodology and to addressing both Objectives 1 and 2.

Example 2.2.3. *To illustrate the 3DVAR algorithm* (2.4) *for state estimation we consider the Lorenz '96 (singlescale) model from Section 2.A. The unknown $v \in C(\mathbb{R}^+, \mathbb{R}^L)$ satisfies the equations*

$$\dot{v}_\ell = -v_{\ell-1}(v_{\ell-2} - v_{\ell+1}) - v_\ell + F + h_v m\big(v_\ell\big), \quad \ell = 1 \dots L\,, \tag{2.9a}$$

$$v_{\ell+L} = v_\ell, \quad \ell = 1 \dots L\,. \tag{2.9b}$$

Here we set $L = 9$, $h_v = -0.8$ and $F = 10$. Function m is shown in Figure Figure 2.1. [1]

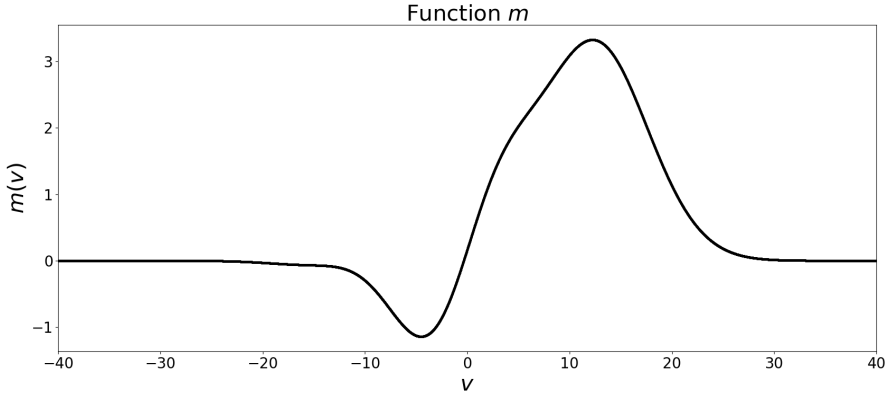


Figure 2.1: Graph of function m appearing in Lorenz '96 model (2.9).

We let Ψ denote the solution operator for (2.9) *over the observation time interval τ, suppresing the explicit dependence on τ for notational convenience. We emphasize that, at the parameter values we have chosen, the solution to* (2.9) *is chaotic and*

[1] We note that setting $h_v = 0$ in (2.9) leads to the standard singlescale Lorenz'96 model. We have $h_v \neq 0$ leading to a nonstandard version of the model. However, the specific choice of function $m(\cdot)$ does not make any material difference to what is presented in this example; any function $m(\cdot)$ for which the equation is well-posed will lead to similar conclusions.

exhibits sensitivity to perturbations. Prediction is thus challenging. But we will show that use of data enables accurate prediction.

We consider observations $\{y_n^\dagger\}_{n\in\mathbb{Z}^+}$ *arising from the model*

$$\begin{aligned} v_{n+1}^\dagger &= \Psi(v_n^\dagger) + \xi_n^\dagger, \\ y_{n+1}^\dagger &= h(v_{n+1}^\dagger) + \eta_{n+1}^\dagger, \end{aligned}$$

where $\{\xi_n^\dagger\}_{n\in\mathbb{Z}^+}$, $\{\eta_n^\dagger\}_{n\in\mathbb{N}}$ *are mutually independent Gaussian sequences defined by*

$$\xi_n^\dagger \sim \mathsf{N}(0,\sigma^2 I) \text{ i.i.d.}, \quad \eta_n^\dagger \sim \mathsf{N}(0,\gamma^2 I) \text{ i.i.d. .}$$

Because of the chaotic nature of the dynamical system defined by iteration of Ψ*, a key question concerning the problem of determining the state* $v_n^\dagger$ *from* $Y_n^\dagger$ *is whether the observations compensate for the sensitive dependence of the state evolution, enabling accurate recovery of the state; and whether there is then a choice of* K *in 3DVAR which enables this data to be used to accurately recover the state.*

We assume that the observation function is linear: $h(v) = Hv$ *for matrix* $H : \mathbb{R}^9 \to \mathbb{R}^6$ *defined by*

$$Hv = (v_1, v_2, v_4, v_5, v_7, v_8)^\top. \tag{2.10}$$

The 3DVAR algorithm (2.3) *reduces, in the setting of this example, to the mapping*

$$v_{n+1} = (I - KH)\Psi(v_n) + Ky_{n+1}^\dagger. \tag{2.11}$$

We define the filter by choosing gain $K : \mathbb{R}^6 \to \mathbb{R}^9$ *to be*

$$Kw = (w_1, w_2, 0, w_3, w_4, 0, w_5, w_6, 0)^\top. \tag{2.12}$$

To interpret the algorithm, and motivate the choice of K*, given* H*, notice that*

$$KHv = (v_1, v_2, 0, v_4, v_5, 0, v_7, v_8, 0)^\top, \tag{2.13a}$$

$$(I - KH)v = (0, 0, v_3, 0, 0, v_6, 0, 0, v_9)^\top, \tag{2.13b}$$

$$HK = I. \tag{2.13c}$$

Applying the observation map H *to the recursion* (2.11) *and using (2.13c) we find that*

$$Hv_{n+1} = y_{n+1}^\dagger = H\big(\Psi(v_n^\dagger) + \xi_n^\dagger\big) + \eta_{n+1}^\dagger.$$

Assuming that σ^2 *and* γ^2 *are small and neglecting the noise contributions shows that*

$$y_{n+1}^\dagger \approx H\Psi(v_n^\dagger) \approx Hv_{n+1}^\dagger.$$

Thus, ignoring small noise perturbations, $y_{n+1}^{\dagger} \approx Hv_{n+1}^{\dagger}$. *Then, using (2.13a) and (2.13b) we see that the algorithm* (2.11) *has the very natural (approximate) interpretation of iterating using the model* Ψ *to update the unobserved components and using the observed true state to update the observed components. This explains why the specific choice of* K *is reasonable.*

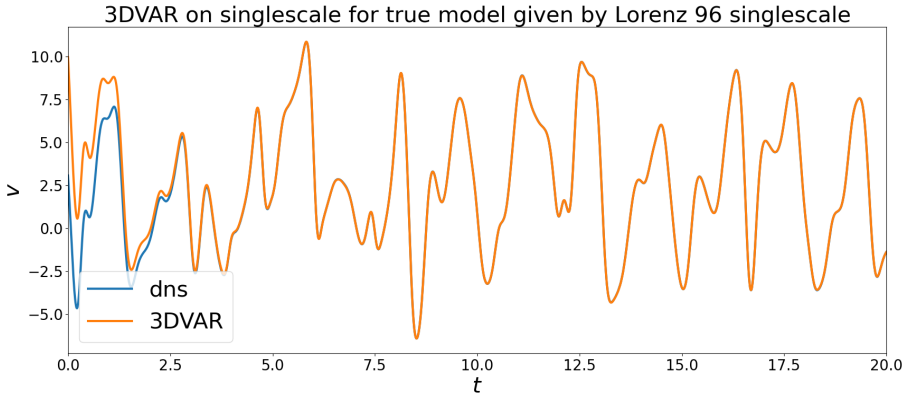
Because of this interpretation it is natural to study the 3DVAR algorithm, for this example, by displaying the output of 3DVAR on one of the unobserved components, and comparing with the truth; the key question is whether the observed components induce synchronization of 3DVAR with the truth in the unobserved components. Thus, in the following numerical experiments, we display component v_3.

Figure Figure 2.2 illustrates the foregoing intuition about the behavior of 3DVAR in experiments conducted with the choice $\tau = 10^{-3}$. *For small noise with standard deviations of size* 10^{-3} *we observe in Figure Figure 2.2a the phenomenon of near perfect synchronization of the 3DVAR output with the truth. Although not shown here, this synchronization occurs in all components of the solution, observed and unobserved, and thus the entire state of 3DVAR synchronizes with the truth, up to a small error on the scale of the noise. The algorithm thus produces an accurate estimate of the true state. In Figure Figure 2.2b larger state and observational noise, of standard deviation* 10^{-1}, *is considered. In this scenario 3DVAR still captures the correct trend of the true dynamics, but there are clear overshoots and undershoots in the estimates; this occurs because the noise is larger than in Figure Figure 2.2a and because noise is not accounted for in the 3DVAR algorithm.*

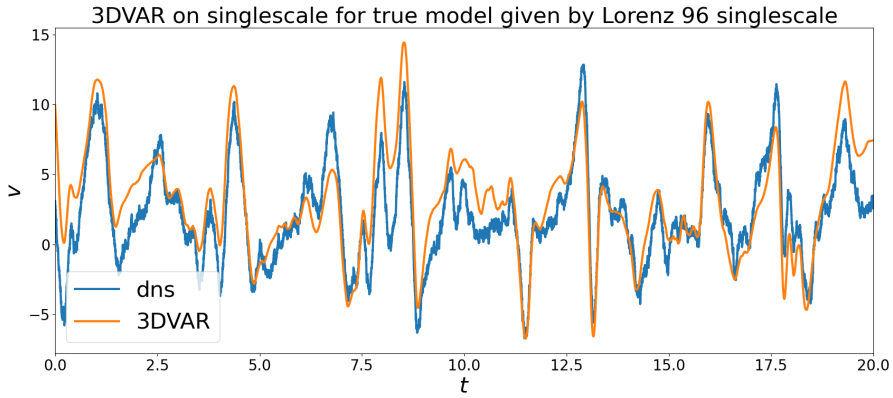
It is intuitive that the synchronization phenomena studied above will depend not only on the size of the noise, but also on the observation time intervals τ. *Figure Figure 2.3 illustrates the effect of varying* τ, *in the setting where the noise standard deviation is* 10^{-3}. *The simulations show that, for the larger value of* τ, *3DVAR does not estimate the true state as accurately as for the smaller value.* ■

Remark 2.2.4. *A common source of error in application of data assimilation algorithms in practice arises from the fact that the data is not produced by the mathematical model used for assimilation. This is known as* model misspecification. *In the context of this section we will make the* perfect model *assumption, avoiding the model misspecification issue.*

The control theoretic approach of 3DVAR addresses Objective 1. However, when



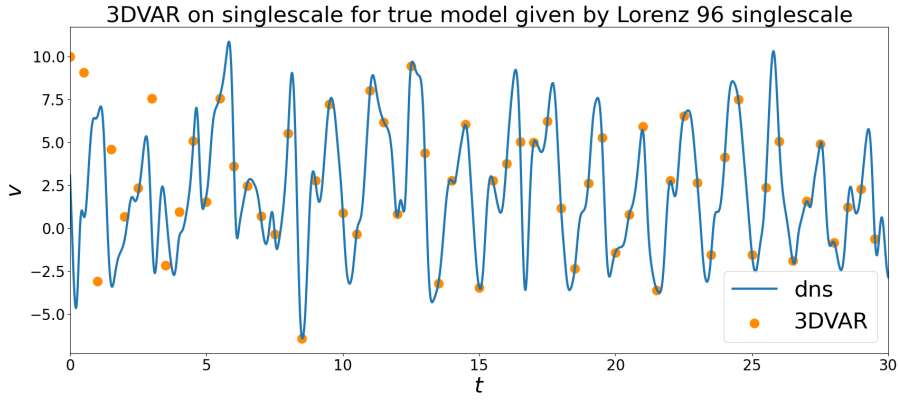
(a)



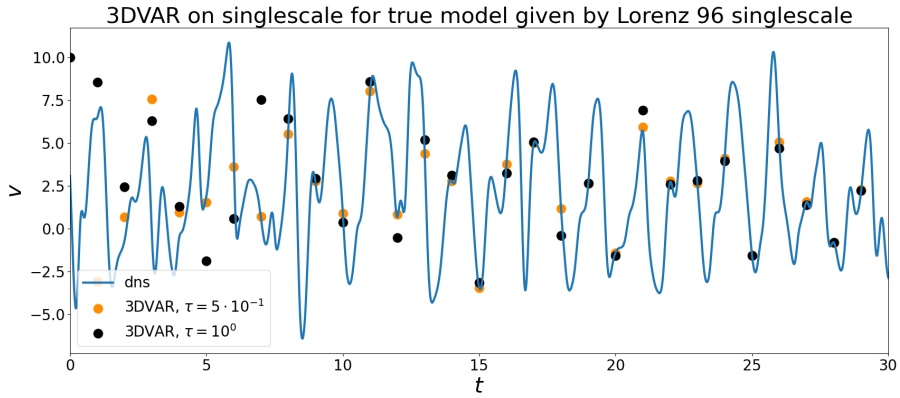
(b)

Figure 2.2: In both (a) and (b) the estimates of v_3 in time produced by 3DVAR using observation time interval $\tau = 10^{-3}$ are displayed and compared with dns. In (a) σ and γ are set to 10^{-3}, while in (b) to 10^{-1}. The acronym "dns" refers to direct numerical simulation of a true trajectory of the chaotic dynamical system. In both cases, it is noteworthy that 3DVAR is able to synchronize with the dns even though it is initialized far from the true initial condition. This is an example of data assimilation overcoming sensitive dependence in a chaotic system.

$|C_0|$, $|\Gamma|$ and $|\Sigma|$ are no longer necessarily small, so that the state and observations are subject to noise, it is natural to try and generalize the approach to address Objective 2, and include non-zero covariances; Figure Figure 2.2b from Example 2.2.3 shows that this may indeed be needed when the noise is larger. A natural stochastic generalization of the observer approach is as follows: from current state estimate v_n, given $Y_n^\dagger$, predict the outcome of the model and data from the update equations (2.1), which we denote by $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$; then correct the state estimate by nudging the mean of the prediction using the mismatch between observed and predicted data $(y_{n+1}^\dagger, \widehat{y}_{n+1})$. Given v_n computed from $Y_n^\dagger$ this results in state estimate v_{n+1} from



(a)



(b)

Figure 2.3: In both (a) and (b) the noise standard deviations σ and γ are set to 10^{-3}. In (a) we display the estimates of v_3 in time produced by 3DVAR using observation time interval $\tau = 5 \cdot 10^{-1}$, compared with dns; (b) displays the estimates obtained at unit time using assimilation at $\tau = 5 \cdot 10^{-1}$ and $\tau = 10^0$. Again the acronym "dns" refers to direct numerical simulation of the true chaotic dynamics. 3DVAR successfully synchronizes with the dns at the smaller value of τ but fails to do as well when the observation time interval τ is larger.

$Y_{n+1}^{\dagger}$ defined through the following stochastic dynamical system, assumed to hold for all $n \in \mathbb{Z}^+$:

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.14a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}, \tag{2.14b}$$

$$v_{n+1} = \widehat{v}_{n+1} + K_n(y_{n+1}^{\dagger} - \widehat{y}_{n+1}); \tag{2.14c}$$

here v_0, ξ_n and η_{n+1} are random variables given by the known distributions in (2.2). Note that the innovation $\mathfrak{I}_n$ is now modified from (2.5) to read

$$\mathfrak{I}_n = y_{n+1}^{\dagger} - \widehat{y}_{n+1}. \tag{2.15}$$

The data $\{y^\dagger_{n+1}\}_{n\in\mathbb{Z}^+}$ is a fixed realization of (2.1). Given this data, equations (2.14) then define a random map $v_n \mapsto v_{n+1}$ which uses knowledge of the model and the observed data to update our state estimate. The predicted state and data $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ are also referred to as the *simulated state and data*. Choice of the *gain matrix* K_n will complete definition of an algorithm. The key question we proceed to study in subsequent sections concerns the choice of $\{K_n\}_{n\in\mathbb{Z}^+}$ when addressing Objective 2. Before doing so we build on Example 2.2.3, studying the effect of noise on the 3DVAR algorithm, which is aimed at addressing Objective 1, and for which $K_n = K$ is constant. The resulting Example 2.2.5 demonstrates the need for an adaptive choice of K_n when noise is significant; it serves to motivate algorithms which address Objective 2.

Example 2.2.5. *We return to the setting of Example 2.2.3 and consider the effect of including noise in 3DVAR as in* (2.14)*. We again study the Lorenz '96 singlescale model* (2.9) *for unknown* $v \in C(\mathbb{R}^+, \mathbb{R}^L)$*, with* $L = 9, h_v = -0.8$ *and* $F = 10$*. We let* Ψ *denote the solution operator for* (2.9) *over the observation time interval* τ *and consider observations* $\{y^\dagger_n\}_{n\in\mathbb{Z}^+}$ *defined by*

$$\begin{aligned} v^\dagger_{n+1} &= \Psi(v^\dagger_n) + \xi^\dagger_n, \\ y^\dagger_{n+1} &= h(v^\dagger_{n+1}) + \eta^\dagger_{n+1}, \end{aligned}$$

where $\{\xi^\dagger_n\}_{n\in\mathbb{Z}^+}$, $\{\eta^\dagger_n\}_{n\in\mathbb{N}}$ *are mutually independent Gaussian sequences*

$$\xi^\dagger_n \sim \mathsf{N}(0, \sigma^2 I) \text{ i.i.d.}, \quad \eta^\dagger_n \sim \mathsf{N}(0, \gamma^2 I) \text{ i.i.d.},$$

with $\sigma = 0.1$ *and* $\gamma = 0.1$*. We again assume that the observation function is linear:* $h(v) = Hv$ *for matrix* $H : \mathbb{R}^9 \to \mathbb{R}^6$ *defined by* (2.10)*. As in Example 2.2.3 we choose fixed gain* $K_n \equiv K$ *with* $K : \mathbb{R}^6 \to \mathbb{R}^9$ *defined by* (2.12)*.*

Figure Figure 2.4 illustrates that this version of noisy 3DVAR produces trajectories that resemble the true signal better than noise-free 3DVAR (2.11)*. However the noisy 3DVAR does not perform better than the noise-free 3DVAR, in this setting where true state and true observation noise levels are high, in a quantitative sense. To demonstrate this, we compute the mean squared error between the estimates arising from the 3DVAR algorithm* (2.11) *and the true states, and the mean squared error between the estimates obtained using noisy 3DVAR algorithm* (2.14) *and the true states. Given either method the error e is computed using the following formula, in which, recall,* $\{v^\dagger_n\}$ *is the truth and* $\{v_n\}$ *is the output of the 3DVAR or noisy 3DVAR*

algorithm:

$$e = \frac{1}{N \cdot d_v} \sum_{n=1}^{N} \left| v^{\dagger}_{n^*+n} - v_{n^*+n} \right|^2. \tag{2.16}$$

Time $t^* = n^*\tau$ is chosen to remove error from the incorrect initialization, and focus on quantifying error in statistical steady state, after synchronization. Here N is such that $(n^* + N)\tau = T$ and $T - t^*$ is chosen large enough to allow time-averaging over a long enough window to capture the statistical steady state. In this case, we compute this average for the estimates obtained after time $t^* = 3$, up to $T = 20$, and find errors $e_{3DVAR} = 1.65 \cdot 10^0$ and $e_{noisy3DVAR} = 3.30 \cdot 10^0$. Clearly use of noisy 3DVAR does not improve the error, in comparison with noise-free 3DVAR. ■

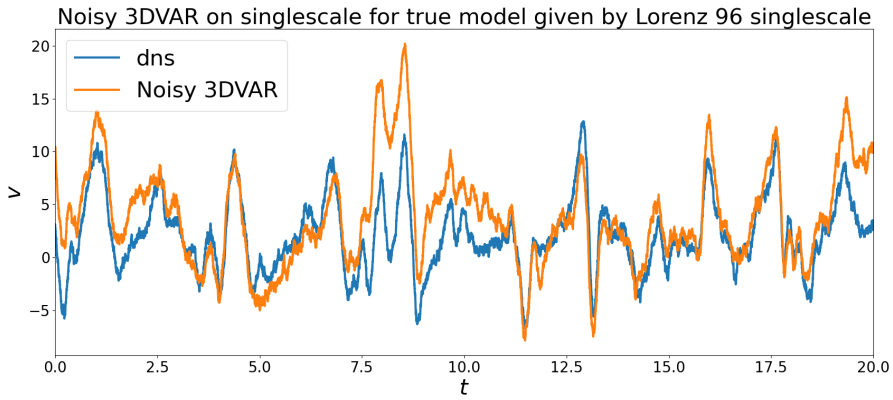


Figure 2.4: In this experiment we set the noise levels $\sigma = 10^{-1}, \gamma = 10^{-1}$. Again the acronym "dns" refers to direct numerical simulation. We display the estimates of v_3 in time produced by noisy 3DVAR against the true dynamics using observation time interval $\tau = 10^{-3}$. This should be compared with Figure Figure 2.2b in which the noise-free 3DVAR is deployed to solve the same problem. Notice that adding noise to 3DVAR has not improved the recovery of the true trajectory. However qualitatively the output of 3DVAR now resembles the true signal more closely.

Example 2.2.5 highlights the need to quantify uncertainty and pass to a probabilistic interpretation (Objective 2) of the filtering problem; and, in particular, to make an informed choice of adaptive gain matrices K_n. We turn to the probabilistic interpretation in the next subsection; in subsequent subsections we derive algorithms, leading in particular to a specific choice of adaptive gain matrices.

2.2.3 Probabilistic Perspective

We have shown that the 3DVAR methodology can recover the state of a (possibly chaotic) dynamical system, even though the initial condition is not known, by ex-

ploiting the observations. However 3DVAR does not quantify uncertainty in the state estimate; it is derived from a purely control theoretic perspective. We now introduce a probabilistic perspective which enables us to address the issue of uncertainty quantification. Subsection 2.2.3.1 discusses the unconditioned dynamics from the perspective of evolution of probability densities. In Subsection 2.2.3.2 we define the filtering distribution and describe this from the perspective of evolution of probability densities. Subsection 2.2.3.3 introduces the *sample path perspective* on algorithms for filtering, a central idea in this chapter. In Subsection 2.2.3.4 we establish some notation, used throughout the sequel, that is important for the reader to internalize.

2.2.3.1 Unconditioned Dynamics

To open our development of the probabilistic perspective we first consider the unconditioned dynamics on state $\{v_n\}_{n\in\mathbb{Z}^+}$ defined by (2.1a). We let r_n denote the probability density of random variable v_n and derive an evolution equation for r_n. Irrespective of whether the evolution of the state $\{v_n\}_{n\in\mathbb{Z}^+}$ defined by (2.1a) is linear, the evolution of $\{r_n\}_{n\in\mathbb{Z}^+}$ is linear. Furthermore the evolution of the state and its probability density are uncoupled from one another. The evolution of the probability density implied by (2.1a) is given by

$$r_{n+1} = \mathsf{P} r_n, \tag{2.17a}$$

$$(\mathsf{P}r)(\mathrm{d}v) = \Big(\int_{u\in\mathbb{R}^{d_v}} p(u,v) r(\mathrm{d}u)\Big)\mathrm{d}v, \tag{2.17b}$$

$$p(u,v) = \frac{1}{(2\pi)^{d_v/2}\sqrt{\mathrm{d}et(\Sigma)}} \exp\Big(-\frac{1}{2}|v-\Psi(u)|^2_\Sigma\Big). \tag{2.17c}$$

Thus, in particular, r_n evolves in time n through application of a linear integral operator. We refer to $\{v_n\}_{n\in\mathbb{Z}^+}$ defined by (2.1a) as a *Markov process.* The situation when we condition the state on observations is different, leading to nonlinear evolution of densities.

2.2.3.2 The Filtering Distribution

Here we introduce the *filtering distribution* with density μ_n: the distribution of the conditioned random variable $v_n|Y_n^\dagger$. This captures the knowledge of the state of the system, and uncertainties in the state, given the observations. To understand how uncertainty in estimates of the state evolves it is thus important to understand how μ_n evolves with n. Unlike the unconditioned dynamics, this conditioned dynamics has

a *nonlinear* structure. Nonlinear evolution equations arise in filtering through the incorporation of data. This nonlinearity renders filtering a challenging mathematical and computational problem. To define this evolution we first define the linear operator

$$(\mathsf{Q}\mu)(\mathrm{d}v, \mathrm{d}y) = q(v, y)\mu(\mathrm{d}v)\mathrm{d}y, \tag{2.18a}$$

$$q(v, y) = \frac{1}{(2\pi)^{d_y/2}\sqrt{\mathrm{d}et(\Gamma)}} \exp\Big(-\frac{1}{2}|y - h(v)|^2_\Gamma\Big). \tag{2.18b}$$

Then we define the $n-$dependent family of two nonlinear operators

$$(\mathsf{B}_n\pi)(\mathrm{d}v) = \int_{y\in\mathbb{R}^{d_y}} \delta_{y^\dagger_{n+1}}(y)\pi(\mathrm{d}v, \mathrm{d}y) \Big/ \Big(\int_{(v,y)\in\mathbb{R}^{d_v}\times\mathbb{R}^{d_y}} \delta_{y^\dagger_{n+1}}(y)\pi(\mathrm{d}v, \mathrm{d}y)\Big), \tag{2.19}$$

$$\mathsf{L}_n(\mu)(\mathrm{d}v) = q(v, y^\dagger_{n+1})\mu(\mathrm{d}v) \Big/ \Big(\int_{v\in\mathbb{R}^{d_v}} q(v, y^\dagger_{n+1})\mu(\mathrm{d}v)\Big). \tag{2.20}$$

The nonlinear map $\mu_n \mapsto \mu_{n+1}$ is most easily described by first introducing $\widehat{\mu}_{n+1}$, the distribution of $v_{n+1}|Y^\dagger_n$ and π_{n+1}, the distribution of $(v_{n+1}, y_{n+1})|Y^\dagger_n$. The map from μ_n to $\widehat{\mu}_{n+1}$ is determined by equation (2.1a) and is *linear* as a map from the space of probability measures defined on $\mathbb{R}^{d_v}$ into itself as discussed in the preceding subsection; the map from $\widehat{\mu}_{n+1}$ to π_{n+1} is defined by (2.1b), and is also *linear*, now as a map from the space of probability measures defined on $\mathbb{R}^{d_v}$ into the space of probability measures defined on $\mathbb{R}^{d_v+d_y}$; the map from π_{n+1} to μ_{n+1} is defined by conditioning π_{n+1} on $y^\dagger_{n+1}$ and is *nonlinear* as a map from the space of probability measures defined on $\mathbb{R}^{d_v+d_y}$ into the space of probability measures defined on $\mathbb{R}^{d_v}$. Using the preceding definitions of linear and nonlinear operators we have

$$\widehat{\mu}_{n+1} = \mathsf{P}\mu_n, \tag{2.21a}$$

$$\pi_{n+1} = \mathsf{Q}\widehat{\mu}_{n+1}, \tag{2.21b}$$

$$\mu_{n+1} = \mathsf{B}_n(\pi_{n+1}). \tag{2.21c}$$

Concatenating we find that

$$\mu_{n+1} = \mathsf{B}_n(\mathsf{QP}\mu_n), \quad \mu_0 = \mathsf{N}(m_0, C_0). \tag{2.22}$$

This map defines an inhomogeneous nonlinear map on the space of probability measures on $\mathbb{R}^{d_v}$. The map $\mathsf{B}_n(\mathsf{QP}\cdot)$ may be decomposed into two maps: (i) the *prediction* P, which represents application of the dynamical model (2.1a); and (ii)

application of *Bayes Theorem* [2] through operator $\mathsf{L}_n \cdot := \mathsf{B}_n(\mathsf{Q}\cdot)$, which corresponds to use of likelihood defined by the observation model (2.1b). With this notation we thus obtain

$$\widehat{\mu}_{n+1} = \mathsf{P}\mu_n, \tag{2.23a}$$

$$\mu_{n+1} = \mathsf{L}_n(\widehat{\mu}_{n+1}). \tag{2.23b}$$

We refer to iteration of (2.23) as the *filtering cycle*. The cycle involves iterative interleaving of prediction, using the dynamical model, a linear operation on measures, and Bayes Theorem, using the observation model, a nonlinear operation on measures.

It is important to appreciate that there is, in general, no closed form expression for μ_n defined by the iteration (2.23); thus (2.23) does not constitute an algorithm. However if Ψ and h are linear then, since $\mu_0 = \mathsf{N}(m_0, C_0)$ is Gaussian, it follows that $\widehat{\mu}_{n+1}, \pi_{n+1}, \mu_{n+1}$ are all Gaussian for all $n \in \mathbb{Z}^+$, and closed form expressions, based on dynamical updates of means and covariances, are available. This linear Gaussian setting is discussed in the following example and linear Gaussian examples will be used throughout the chapter. However the main thrust of the chapter concerns nonlinear and non-Gaussian problems; for these problems further ideas, which we will explain in subsequent subsections, are required to make actionable algorithms from the iteration (2.23).

Example 2.2.6. *To define the* Kalman filter *we consider the setting* (2.6)*, where* $\Psi(\cdot)$ *and* $h(\cdot)$ *are both linear. Then* (2.1) *becomes*

$$v_{n+1} = Mv_n + \xi_n, \tag{2.24a}$$

$$y_{n+1} = Hv_{n+1} + \eta_{n+1}. \tag{2.24b}$$

For this problem the mapping (2.23) *may be solved explicitly. In fact* μ_n *and* $\widehat{\mu}_n$ *are both Gaussian and we write their mean-covariance pairs as* (m_n, C_n) *and* $(\widehat{m}_n, \widehat{C}_n)$ *respectively. Then* $\widehat{\mu}_{n+1}$ *is determined from* μ_n *by the formulae*

$$\widehat{m}_{n+1} = Mm_n, \tag{2.25a}$$

$$\widehat{C}_{n+1} = MC_nM^\top + \Sigma, \tag{2.25b}$$

the prediction step. Measure μ_{n+1} *is determined from* $\widehat{\mu}_n$ *by the application of a Bayesian update, solving the inverse problem defined by (2.24b); this inverse*

[2] Often referred to as the *analysis* step in the geophysical data assimilation community.

problem is for v_{n+1} given fixed realization of data $y_{n+1} = y^\dagger_{n+1}$ generated by (2.24)*. Since the prior and posterior are Gaussian we may complete the square to solve the Bayesian inverse problem to give the following update formulae for the mean and precision:*

$$C_{n+1}^{-1} m_{n+1} = \widehat{C}_{n+1}^{-1} \widehat{m}_n + H^\top \Gamma^{-1} y^\dagger_{n+1}, \tag{2.26a}$$

$$C_{n+1}^{-1} = \widehat{C}_{n+1}^{-1} + H^\top \Gamma^{-1} H. \tag{2.26b}$$

By use of the Woodbury matrix identity we obtain the following formulae expressed in terms of covariances rather than precisions:

$$m_{n+1} = \widehat{m}_{n+1} + \widehat{C}_{n+1} H^\top (H \widehat{C}_{n+1} H^\top + \Gamma)^{-1} (y^\dagger_{n+1} - H \widehat{m}_{n+1}), \tag{2.27a}$$

$$C_{n+1} = \widehat{C}_{n+1} - \widehat{C}_{n+1} H^\top (H \widehat{C}_{n+1} H^\top + \Gamma)^{-1} H \widehat{C}_{n+1}. \tag{2.27b}$$

The update equations (2.25) *simply represent propagation of a Gaussian under the linear dynamics defined by (2.24a);* (2.27) *corresponds to application of Bayes theorem, and in this particular linear setting to conditioning a Gaussian, using the observations defined by (2.24b) with $y_{n+1} = y^\dagger_{n+1}$. The Kalman filter update equations* (2.25)*,* (2.27) *are well-defined if $\Gamma \succ 0$.* ■

The Gaussian setting of the preceding example is very special. But the idea of making a *Gaussian approximation* will play a central role in ensemble Kalman methods and as a consequence the explicit calculations in the example will be generally useful. The perspective of invoking Gaussian approximations is introduced in Subsection 2.2.4; it is subsequently developed to form the backbone of the methodology highlighted in this chapter.

2.2.3.3 The Sample Path Perspective

In Subsection 2.2.3.1 we demonstrated that the (typically nonlinear) state space evolution of $\{v_n\}_{n \in \mathbb{Z}^+}$ defined by (2.1a) may be alternatively viewed in terms of the linear evolution of probability density functions r_n defined by (2.17). In Subsection 2.2.3.2 we showed that, when conditioned on observations, the evolution of the probability density function becomes nonlinear and is defined by (2.21) or (2.23). In this section we address the question of finding an evolution in state space that is consistent with this nonlinear evolution of densities defined by (2.21) or (2.23). Our aim is to generalize the control theoretic data assimilation algorithm given by (2.14) in order to find such an evolution.

Throughout this subsection let v_n be a random variable distributed according to $\mathrm{Law}(v_n)$, let $\widehat{v}_{n+1}$ be random variable with $\mathrm{Law}(\widehat{v}_{n+1}) = \mathsf{P}\,\mathrm{Law}(v_n)$ and let $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ be random variable with law $\mathsf{Q}\,\mathrm{Law}(\widehat{v}_{n+1})$. Using Law avoids a proliferation of notation for the different measures arising from the use of various different algorithms to approximate the filtering cycle.

All of the algorithms that we will introduce in what follows are based on a prediction of state, and possibly observation, from $\mathrm{Law}(v_n)$. To this end recall (2.14) and for all $n \in \mathbb{Z}^+$,

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.28a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}. \tag{2.28b}$$

The distributions of $\widehat{v}_{n+1}$ and $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ given by these equations define $\mathsf{P}\,\mathrm{Law}(v_n)$ and $\mathsf{Q}\,\mathrm{Law}(\widehat{v}_{n+1})$, respectively. A common theme in this chapter is to augment these equations for the predicted state and data with a final map, to generalize (2.14c), of the form

$$(\widehat{v}_{n+1}, \widehat{y}_{n+1}) \mapsto v_{n+1}, \tag{2.29}$$

or

$$\widehat{v}_{n+1} \mapsto v_{n+1}. \tag{2.30}$$

These maps are chosen, respectively, to mimic (2.21) or (2.23). Then, if $v_n \sim \mu_n$ the maps are designed so that $v_{n+1} \sim \mu_{n+1}$ where μ_n and μ_{n+1} are related by the filtering update (2.21) or, equivalently, (2.23). A key observation is that these maps will necessarily depend on the distributions of $\widehat{v}_{n+1}$ and $\widehat{y}_{n+1}$ rendering the associated Markov processes of mean field type. Thus, in contrast to the unconditioned dynamics, the state space evolution does not decouple from the evolution of the associated probability density function; rather it depends on it. The resulting state space evolution is said to define a *nonlinear Markov process.*

Note that, once (2.28) is augmented with either (2.29) or (2.30), we have a state space evolution that describes, through the probability distribution of v_n, the filtering process; the state space evolution can then be used as the basis for algorithms. Thus the key question for such a program is the identification of either (2.29) or (2.30). The existence of *transport maps* or, more generally, *couplings*, which define the steps (2.29) or (2.30), follows under quite general conditions. But finding them explicitly is generally difficult. Furthermore, the maps (2.29) and (2.30) are not uniquely

defined, in general. As a consequence of the difficulty in identifying mean field transport maps, the algorithms we study will be based on identifying maps (2.29) or (2.30) which only *approximately* achieve the filtering update (2.21). However, all formulations considered in this survey will be of mean field type.

In summary, we refer to equations (2.28,2.29) or (2.28a,2.30) as providing a *sample path perspective*. The resulting algorithms update state $v_n \mapsto v_{n+1}$ and are designed to exactly (in theory), and approximately (in practice), reproduce the *probabilistic perspective* encapsulated in the three steps of (2.21), or the two steps of (2.23). These algorithms result in a sample path $\{v_\ell\}_{\ell=0}^n$ with property that (possibly only approximately) $\text{Law}(v_\ell) = \mu_\ell$. This idea is central to the algorithmic developments in the chapter.

2.2.3.4 Important Repeatedly Used Notation

The approximations we develop will be based on matching first and second order moments. In service of designing these approximations, it is useful to define various first and second order statistics computed under the law of $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$. First define mean and covariance of $\widehat{v}_{n+1}$:

$$\widehat{m}_{n+1} = \mathbb{E}\widehat{v}_{n+1}, \tag{2.31a}$$

$$\widehat{C}_{n+1} = \mathbb{E}\Big((\widehat{v}_{n+1} - \widehat{m}_{n+1}) \otimes (\widehat{v}_{n+1} - \widehat{m}_{n+1}) \Big). \tag{2.31b}$$

Then define mean of the predicted data, cross-covariance from predicted data to state and covariance of the data:

$$\widehat{o}_{n+1} = \mathbb{E}\widehat{y}_{n+1}, \tag{2.32a}$$

$$\widehat{C}^{vy}_{n+1} = \mathbb{E}\Big((\widehat{v}_{n+1} - \widehat{m}_{n+1}) \otimes (\widehat{y}_{n+1} - \widehat{o}_{n+1}) \Big), \tag{2.32b}$$

$$\widehat{C}^{yy}_{n+1} = \mathbb{E}\Big((\widehat{y}_{n+1} - \widehat{o}_{n+1}) \otimes (\widehat{y}_{n+1} - \widehat{o}_{n+1}) \Big). \tag{2.32c}$$

From these covariances we define the matrix

$$K_n = \widehat{C}^{vy}_{n+1} (\widehat{C}^{yy}_{n+1})^{-1}. \tag{2.33}$$

This particular choice of K_n, known as the *Kalman gain*, plays a central role in the mean field maps which underpin ensemble Kalman methods through their particle approximations. Note that, if $\widehat{C}^{yy}_{n+1}$ is not invertible, then its action may still be defined through a pseudo-inverse.

It is sometimes useful to express the Kalman gain K_n in terms of the variable $\widehat{h}_{n+1} = h(\widehat{v}_{n+1})$ and without reference to predicted data $\widehat{y}_{n+1}$. For this purpose we define the following correlation matrices, computed under the law of $\widehat{v}_{n+1}$:

$$\widehat{o}_{n+1} = \mathbb{E}h(\widehat{v}_{n+1}), \tag{2.34a}$$

$$\widehat{C}^{vh}_{n+1} = \mathbb{E}\Big(\big(\widehat{v}_{n+1} - \widehat{m}_{n+1}\big) \otimes \big(\widehat{h}_{n+1} - \widehat{o}_{n+1}\big) \Big), \tag{2.34b}$$

$$\widehat{C}^{hh}_{n+1} = \mathbb{E}\Big(\big(\widehat{h}_{n+1} - \widehat{o}_{n+1}\big) \otimes \big(\widehat{h}_{n+1} - \widehat{o}_{n+1}\big) \Big). \tag{2.34c}$$

Then, in place of (2.32) and (2.33), we have

$$\widehat{C}^{vy}_{n+1} = \widehat{C}^{vh}_{n+1}, \quad \widehat{C}^{yy}_{n+1} = \widehat{C}^{hh}_{n+1} + \Gamma, \tag{2.35a}$$

$$K_n = \widehat{C}^{vh}_{n+1} \left(\widehat{C}^{hh}_{n+1} + \Gamma \right)^{-1}. \tag{2.35b}$$

Note that, if $\Gamma \succ 0$, K_n is well-defined without recourse to the use of pseudo-inverse.

Example 2.2.7. *In the setting of the linear and Gaussian Example 2.2.6 we have*

$$\widehat{C}^{vy}_{n+1} = \widehat{C}_{n+1} H^\top,$$
$$\widehat{C}^{yy}_{n+1} = H \widehat{C}_{n+1} H^\top + \Gamma,$$

and the mean update (2.27a) may be written

$$m_{n+1} = \widehat{m}_{n+1} + K_n (y^\dagger_{n+1} - H\widehat{m}_{n+1}).$$

In particular only the single covariance $\widehat{C}_{n+1}$ needs to be computed. Note the similarity of the resulting algorithm with the 3DVAR algorithm (2.8)*, in the linear Gaussian setting, since in this linear case $\widehat{m}_{n+1} = Mm_n$. It differs only through having a time-varying gain matrix K_n.* ■

This linear Gaussian setting provides some motivation for the Kalman gain. The origin of this key concept in the more general nonlinear and non-Gaussian setting will be described in the next subsection.

2.2.4 Gaussian Projected Filtering Distribution

The *Gaussian projected filter* gives a Gaussian approximation of the true filtering distribution. It is defined by the following three steps:

- (i) taking input Gaussian at time n as $\text{Law}(v_n)$ and pushing this measure forward under (2.28) to find (typically non-Gaussian) measure $\text{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1})$;

- (ii) projecting this joint measure onto the nearest Gaussian (in a sense that we will make precise);
- (iii) conditioning this Gaussian on the data $y_{n+1}^{\dagger}$ to find output Gaussian at time $n+1$.

We note that conditioning a Gaussian random variable on linear functionals of the random variable returns another Gaussian. Thus the algorithm maps Gaussians to Gaussians. The resulting approximation of the filtering distribution plays an important role in motivating the mean field maps, introduced in Subection Subsection 2.2.5, that underly ensemble Kalman methods. It is also of interest as a method in its own right.

In what follows in this subsection we introduce the Gaussian projected approximation to the evolution (2.22). In the case where $\Psi(\cdot)$ and $h(\cdot)$ are linear the resulting formulae deliver exact solutions of the filtering cycle (2.22), leading to the Kalman filter as given in Example 2.2.6. The Gaussian projected filter also leads to a derivation of the Kalman gain (2.33), beyond the linear Gaussian setting.

To describe the Gaussian projected approximation to the filtering distribution, we define the map G, definition of which uses $\mathcal{P} = \mathcal{P}(\mathbb{R}^d)$ and $\mathcal{G} = \mathcal{G}(\mathbb{R}^d)$ defined in Section 1.2.

Definition 2.2.8. *Define* $\mathsf{G} : \mathcal{P} \mapsto \mathcal{G}$ *by*

$$\mathsf{G}\mu = \mathsf{N}(m^\mu, C^\mu), \qquad m^\mu = \mathbb{E}_\mu u, \quad C^\mu = \mathbb{E}_\mu\big((u - \mathbb{E}_\mu u) \otimes (u - \mathbb{E}_\mu u)\big),$$

where $u \sim \mu$. ■

Thus the map G applied to measure μ simply computes the Gaussian with mean and covariance calculated with respect to the, typically non-Gaussian, measure μ. Notice that G is the identity on Gaussians. Furthermore $\mathsf{G} \circ \mathsf{G} = \mathsf{G}$. We refer to this as a *projection* onto Gaussians because it corresponds to finding the closest point to given measure μ, with respect to a Kullback-Leibler divergence[3]:

$$\mathsf{G}\mu = \mathrm{a}rgmin_{\pi \in \mathcal{G}} d_{\mathrm{KL}}(\mu||\pi). \tag{2.36}$$

We now use mapping G to find an approximation to the evolution (2.21) which generates measures remaining Gaussian; intuitively this will be a good approximation whilst the measures $\{\mu_n\}$ evolving under (2.22) remain close to Gaussian. To

[3]For details see the bibliography Subsection 2.2.7.

this end we consider random variable $v_n \sim \mu_n^G$ where the probability measure μ_n^G evolves according to

$$\mu_{n+1}^G = \mathsf{B}_n(\mathsf{GQP}\mu_n^G), \quad \mu_0^G = \mathsf{N}(m_0, C_0). \tag{2.37}$$

This may be decomposed as follows:

$$\widehat{\mu}_{n+1}^G = \mathsf{P}\mu_n^G, \tag{2.38a}$$

$$\pi_{n+1}^G = \mathsf{Q}\widehat{\mu}_{n+1}^G, \tag{2.38b}$$

$$\mu_{n+1}^G = \mathsf{B}_n(\mathsf{G}\pi_{n+1}^G). \tag{2.38c}$$

Map (2.37) defines a nonlinear Markov process, similarly to (2.22). It also maps Gaussians into Gaussians. This fact follows from the fact that the nonlinear map $\mathsf{B}_n(\cdot)$, which represents conditioning, maps Gaussians into Gaussians. The map $\mu_n^G \mapsto \mu_{n+1}^G$ hence defines a deterministic mapping from the mean m_n and covariance C_n of μ_n^G into the mean m_{n+1} and covariance C_{n+1} of μ_{n+1}^G. We now identify this map explicitly.

For $v_n \sim \mu_n^G$ we introduce the random variables $\widehat{v}_{n+1}, \widehat{y}_{n+1}$ defined by (2.28). It then follows that $\widehat{v}_{n+1} \sim \widehat{\mu}_{n+1}^G = \mathsf{P}\mu_n^G$ and that $(\widehat{v}_{n+1}, \widehat{y}_{n+1}) \sim \pi_{n+1}^G = \mathsf{QP}\mu_n^G$. Note also that $\widehat{\mu}_{n+1}^G$ and π_{n+1}^G are not Gaussian, but are defined by the Gaussian μ_n^G and hence completely determined by m_n and C_n. Thus the mean and covariance under $\widehat{\mu}_{n+1}^G$ are $(\widehat{m}_{n+1}, \widehat{C}_{n+1})$ given by (2.31). Furthermore, $\mathsf{G}\pi_{n+1}^G$ is defined by

$$\mathsf{G}\pi_{n+1}^G = \mathsf{N}\Bigg(\begin{bmatrix}\widehat{m}_{n+1} \\ \widehat{o}_{n+1}\end{bmatrix}, \begin{bmatrix}\widehat{C}_{n+1} & \widehat{C}_{n+1}^{vy} \\ (\widehat{C}_{n+1}^{vy})^\top & \widehat{C}_{n+1}^{yy}\end{bmatrix}\Bigg), \tag{2.39}$$

where all relevant quantities are defined in Subsection 2.2.3.4.

We now condition the Gaussian $\mathsf{G}\pi_{n+1}^G$, on the second component of the vector taking value $y_{n+1}^\dagger$. From this we find the Gaussian measure $\mu_{n+1}^G = \mathsf{B}_n(\mathsf{G}\pi_{n+1}^G)$ characterized by mean m_{n+1} and covariance C_{n+1} given by the following lemma.

Lemma 2.2.9. *Assume that $\Gamma \succ 0$. Let m_n and C_n denote the mean and covariance under the Gaussian projected filter. Consider equations* (2.28) *initialized at $v_n \sim \mathsf{N}(m_n, C_n)$, and then $(\widehat{m}_{n+1}, \widehat{C}_{n+1})$ defined by* (2.31)*; furthermore, define the mean of the observed data and covariances given by* (2.32)*. Then $\widehat{C}_{n+1}^{yy} \succ 0$ for all $n \in \mathbb{Z}^+$ and*

$$m_{n+1} = \widehat{m}_{n+1} + \widehat{C}_{n+1}^{vy}(\widehat{C}_{n+1}^{yy})^{-1}\big(y_{n+1}^\dagger - \widehat{o}_{n+1}\big), \tag{2.40a}$$

$$C_{n+1} = \widehat{C}_{n+1} - \widehat{C}_{n+1}^{vy}(\widehat{C}_{n+1}^{yy})^{-1}\big(\widehat{C}_{n+1}^{vy}\big)^\top, \tag{2.40b}$$

where $\{y_n^\dagger\}$ arises from a fixed realization of (2.1). $\diamond$

Proof. We first note that $\widehat{C}^{yy}_{n+1} \succ 0$. Indeed, since by assumption $\Gamma \succ 0$ and by definition $\widehat{C}^{hh}_{n+1} \succeq 0$, then $\widehat{C}^{hh}_{n+1} + \Gamma \succ 0$. Hence by (2.35) we have that $\widehat{C}^{yy}_{n+1} \succ 0$. Now consider the distribution of the Gaussian $\mathsf{G}\pi^G_{n+1}$ given by (2.39). Conditioning the resulting joint random variable on $(v, y) \in \mathbb{R}^{d_v} \times \mathbb{R}^{d_y}$ on $y = y^\dagger_{n+1}$, it is possible to conclude that from standard formulae for conditioned Gaussians that m_{n+1} and C_{n+1} are given by the expressions in (2.40). □

Equations (2.28), (2.31), (2.32), and (2.40) define a mapping from μ^G_n, characterized by (m_n, C_n), into μ^G_{n+1}, characterized by (m_{n+1}, C_{n+1}). They comprise an explicit set of formulae for the mapping (2.37): since Gaussians are determined by mean and covariance, the map on measures is completely determined by the map from (m_n, C_n) to (m_{n+1}, C_{n+1}).

We note that (2.40) can also be rewritten as

$$m_{n+1} = \widehat{m}_{n+1} + \widehat{C}^{vh}_{n+1}(\widehat{C}^{hh}_{n+1} + \Gamma)^{-1}\big(y^\dagger_{n+1} - \widehat{o}_{n+1}\big), \tag{2.41a}$$

$$C_{n+1} = \widehat{C}_{n+1} - \widehat{C}^{vh}_{n+1}(\widehat{C}^{hh}_{n+1} + \Gamma)^{-1}\big(\widehat{C}^{vh}_{n+1}\big)^\top. \tag{2.41b}$$

Equations (2.28), (2.31), (2.34) and (2.41) then also define the mapping from μ^G_n into μ^G_{n+1} and also comprise an explicit set of formulae for the updates of the mean and covariance which characterize mapping (2.37).

Finally we note that, using the definition (2.33) of Kalman gain, we can rewrite (2.40) as

$$\begin{aligned} m_{n+1} &= \widehat{m}_{n+1} + K_n\big(y^\dagger_{n+1} - \widehat{o}_{n+1}\big), \\ C_{n+1} &= \widehat{C}_{n+1} - K_n\big(\widehat{C}^{vy}_{n+1}\big)^\top. \end{aligned}$$

Thus we see how the Kalman gain arises naturally within the context of this Gaussian projected filter.

Remark 2.2.10. *The preceding explicit formulae for* (2.37) *involve the computation of expectations under the non-Gaussian measure* $\mathrm{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1})$. *For this reason they do not constitute an algorithm. A possible approach to algorithmic implementations involves quadrature to approximate the expectations, leading, for example, to the* unscented Kalman filter *approach; details may be found in the bibliography Subsection 2.2.7. However the formulation of explicit maps on Gaussians plays another, important, role in this chapter: we use it as a way of explaining the sense in which the distribution of our mean field models approximate evolution of measures*

under the true filtering distribution; see Subsection 2.2.5.6. The explicit map on means and covariances can be used to derive the Kalman filter, which applies in the linear Gaussian setting, and is presented in Example 2.2.6. ■

2.2.5 Mean Field Maps

In the previous section we did not adopt the sample path perspective, but rather chose to represent the evolution of the filtering distribution, approximately, as the evolution of Gaussians. In this subsection we introduce our first instance of the sample path perspective, finding an evolution in state space which approximately captures the evolution of the filtering distribution. Like the Gaussian projected filter it uses a Gaussian ansatz, but in a different way, leading to a state space evolution which is not Gaussian.

Note that elements of $\mathcal{P}(\mathbb{R}^d)$ are *infinite dimensional* objects, for any d. Thus the filtering distribution defines an evolution in an infinite dimensional space. This fact goes to the heart of the computational challenges faced when solving the filtering problem. These computational challenges are further exacerbated when $d \gg 1$. The manifold of Gaussians $\mathcal{G}(\mathbb{R}^d)$ is finite dimensional, because it is parameterized by the mean and covariance and hence has dimension $\frac{1}{2}d(d+3)$. The preceding subsection provides explicit *finite dimensional maps* for the mean and covariance which characterize the Gaussian projected filter $\mu_n^G \mapsto \mu_{n+1}^G$, an approximation which is (intuitively) accurate when the true filter is close to Gaussian. Nonetheless if $d \gg 1$ this method can still be prohibitive because the algorithm acts on a space of dimension that grows quadratically in d.

To address the issue that the Gaussian projected filter may not be efficient if $d \gg 1$ we introduce, in this section, a more ambitious aim: to find maps on finite dimensional spaces of dimension d with the property that (possibly only approximately) their output is equal in law to the map on measures $\mu_n \mapsto \mu_{n+1}$ given by the filtering cycle. We achieve this by studying transport maps that achieve (2.29) or (2.30); we then weaken this requirement and ask only for transport maps that approximately achieve (2.29) or (2.30) in a manner that we will make precise. When combined with (2.28), either (2.29) or (2.30) gives a sample path evolution which can be used as the basis of algorithms to solve the filtering problem. This transport map viewpoint leads us to the subject of mean field maps, namely random maps that depend on the law of the state being mapped. When approximated by particle methods these maps lead to methods that scale linearly with d, in contrast to the Gaussian projected filter

which scales quadratically.

This section is organized as follows. We start in Subsection Subsection 2.2.5.1 with an introduction to transport maps. Subsection Subsection 2.2.5.2 describes two distinct transport approaches which effect exact filtering: one based on the conditioning component of the overall Bayesian inference step, a transport between probability measures on different spaces; and the other based on the prior to posterior map that constitutes the Bayesian inference step of filtering, a transport between probability measures on the same space. We refer to these maps which effect exact filtering as *perfect transport.*[4] Subsections Subsection 2.2.5.4 and Subsection 2.2.5.5 are concerned with approximations of these two perfect transports, and are motivated in Subsection 2.2.5.3 with an explicit example. In these approximations the pushforward under the transport map is designed to match only the first and second order moments of the target measure. Hence these approximations are closely related to, but they are different from, the previously defined Gaussian projected filter; we elaborate on this connection in the summary Subsection 2.2.5.6. That subsection also includes Example 2.2.15 in which we identify mean field formulations of the Kalman filter; recall that this filter applies only to linear Gaussian systems, and is defined in Example 2.2.6.

2.2.5.1 Transport Maps

We start by setting-up notation used throughout. Consider probability measures π and π' on $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ respectively, and recall the definition of *pushforward* of a measure under under $T : \mathbb{R}^d \to \mathbb{R}^{d'}$: the statement $\pi' = T_\sharp \pi$ is a succinct way of stating that, if $\text{Law}(v) = \pi$ and $v' = T(v)$, then $\text{Law}(v') = \pi'$. Given probability measures π and π' on $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ respectively, a *transport* $T : \mathbb{R}^d \to \mathbb{R}^{d'}$ from π to π' is a map with property that the pushforward of probability measure π under T, $T_\sharp \pi$, is equal to probability measure π'. In the following we refer to π as the *source measure* and π' as the *target measure* defining the transport. In our setting π' will be uniquely determined by π and an observed piece of finite dimensional data. Thus T depends on π and so we may view T as a mapping $\mathbb{R}^d \times \mathcal{P} \to \mathbb{R}^{d'}$, suppressing, for the moment, explicit dependence on the observed data. We can compute the pushforward under T on any measure in $\mathcal{P}$; but when we compute the pushforward

[4]Perfect transport should not be confused with *optimal transport* which identifies among all (perfect) transport maps the one minimizing a certain cost functional such as that leading to the Wasserstein distance. We use *perfect* here to distinguish from the *approximate* transport maps, based only on matching first and second moment; these approximate transport maps underpin ensemble Kalman methods.

on π we obtain π'. We emphasize that transport maps are not uniquely defined by their source and target measures and require certain conditions for their existence, which we assume here to be satisfied. The underlying mathematical concept is that of *coupling of measures*. See bibliography Subsection 2.2.7 for discussion of transport, optimal transport and coupling.

We will also consider classes of *approximate transport maps* which do not achieve transport from π to π', but instead match first and second order moment information (we will be precise below). Such maps will also depend on π.

We now clarify an important notational issue. The dependence of (possibly approximate) transport T on a measure in $\mathcal{P}$ does not affect the definition of pushforward; we employ the following general definition of pushforward for measure-dependent maps, taken to hold for all π_1, π_2 regardless of any assumed relationship between them:

$$T(\cdot;\pi_1)_\sharp\pi_2 = T(\cdot;\widetilde{\pi})_\sharp\pi_2\Big|_{\widetilde{\pi}=\pi_1}; \tag{2.42}$$

in particular,

$$T(\cdot;\pi)_\sharp\pi = T(\cdot;\widetilde{\pi})_\sharp\pi\Big|_{\widetilde{\pi}=\pi}, \tag{2.43a}$$

$$T(\cdot;\pi)_\sharp(\mathsf{G}\pi) = T(\cdot;\widetilde{\pi})_\sharp(\mathsf{G}\pi)\Big|_{\widetilde{\pi}=\pi}. \tag{2.43b}$$

In the preceding, pushforward under $\widetilde{\pi}-$dependent map $T(\cdot,\widetilde{\pi})$ denotes regular pushforward with no relationship assumed between $\widetilde{\pi}$ and measure being pushed forward. Note that we may define $\mathsf{T} : \mathcal{P}(\mathbb{R}^d) \to \mathcal{P}(\mathbb{R}^d)$ by $\mathsf{T}(\pi) = T(\cdot;\pi)_\sharp\pi$. The (approximate) transport maps just identified can be recast as mean field maps when used in the context where source π is the distribution of the input to T.

2.2.5.2 Perfect Transport

Consider the idea of finding a transport map that acts on the joint space of state and data, to effect conditioning with respect to the observed data. To this end we consider the dynamical system, assumed to hold for all $n \in \mathbb{Z}^+$:

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.44a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}, \tag{2.44b}$$

$$v_{n+1} = T^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi_{n+1}, y^\dagger_{n+1}), \tag{2.44c}$$

where $\{y^\dagger_n\}$ arises from a fixed realization of (2.1). Recall that $\mu_n = \mathrm{Law}(v_n)$, $\widehat{\mu}_{n+1} = \mathrm{Law}(\widehat{v}_{n+1})$ and $\pi_{n+1} = \mathrm{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1})$.

This is an example of the sample path perspective, and (2.28, 2.29) in particular. The first two equations, which coincide with (2.28), effect the mappings from μ_n to $\widehat{\mu}_{n+1}$ and from $\widehat{\mu}_{n+1}$ to π_{n+1}. Map[5] $T_n^S(\cdot,\cdot) := T^S(\cdot,\cdot;\pi_{n+1},y_{n+1}^\dagger)$ is then an explicit example of (2.29) defined to effect the desired conditioning of π_{n+1} on $y_{n+1}^\dagger$ in order to obtain μ_{n+1}. The letter S in T^S connotes the dependence of the map on the *stochastic* data $\widehat{y}_{n+1}$. Thus equations (2.44) define a mean field stochastic dynamical system mapping v_n to v_{n+1}: stochastic because of the noise in (2.44a, 2.44b), mean field because the map T^S in (2.44c) depends on the law π_{n+1} of $(\widehat{v}_{n+1},\widehat{y}_{n+1})$, and hence on μ_n. The three update steps in this mean field stochastic dynamical system lead to the following maps on measures:

$$\widehat{\mu}_{n+1} = \mathsf{P}\mu_n, \tag{2.45a}$$

$$\pi_{n+1} = \mathsf{Q}\widehat{\mu}_{n+1}, \tag{2.45b}$$

$$\mu_{n+1} = (T_n^S)_\sharp \pi_{n+1}. \tag{2.45c}$$

This is simply a restatement of (2.21), noting that $(T_n^S)_\sharp$ has been chosen so that pushforward corresponds to conditioning π_{n+1} on data $y_{n+1}^\dagger$ to obtain μ_{n+1}. In particular $\mathsf{T}_n^S(\pi_{n+1}) := (T_n^S)_\sharp\pi_{n+1}$ has property $\mathsf{T}_n^S(\pi_{n+1}) = \mathsf{B}_n(\pi_{n+1})$. Note that the implied map from μ_n to μ_{n+1} is a nonlinear Markov process, because of the dependence of T_n^S on π_{n+1} and hence on μ_n. Furthermore, we have that $\mathrm{Law}(v_\ell) = \mu_\ell$ for all $\ell \in \mathbb{Z}^+$. The important takeaway is that (2.44) defines a sample-path picture of the evolution of the filtering distribution: it provides a map in state space with law governed by the filter. We reemphasize that such a sample-path representation is not uniquely defined.

Consider now a different approach to transport for filtering: we seek a transport map that acts on the state space only to effect Bayes Theorem, i.e. mapping prior $\widehat{\mu}_{n+1}$ to posterior μ_{n+1}. To this end we consider the dynamical system

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.46a}$$

$$v_{n+1} = T^D(\widehat{v}_{n+1};\widehat{\mu}_{n+1},y_{n+1}^\dagger), \tag{2.46b}$$

again assumed to hold for all $n \in \mathbb{Z}^+$. The first equation maps $v_n \sim \mu_n$ to $\widehat{v}_{n+1} \sim \widehat{\mu}_{n+1}$, thus giving a sample path realization of (2.23a). In the second equation, the map $T_n^D(\cdot) := T^D(\cdot;\widehat{\mu}_{n+1},y_{n+1}^\dagger)$ is chosen so that, if $\widehat{v}_{n+1} \sim \widehat{\mu}_{n+1}$ then $v_{n+1} \sim \mu_{n+1}$, thus

[5]It is convenient to use both the notation $T^S(\cdot,\cdot;\pi_{n+1},y_{n+1}^\dagger)$, to be explicit about important dependencies in T^S, and to use the notation T_n^S, for succinct statement of certain formulae when dropping explicit dependence of T^S on π_{n+1} and $y_{n+1}^\dagger$. We will use analogous notation for other mean field maps in what follows.

giving a sample path realization of (2.23b). Thus we have another instance of the sample path perspective, and (2.28, 2.30) in particular.

Equation (2.46) constitutes another mean field stochastic dynamical system: stochastic because of the noise in (2.46a); mean field because the map T^D in (2.46b) depends on the law of $\widehat{v}_{n+1}$ itself, and hence on μ_n. The symbol D distinguishes map T^D from map T^S: map T^D is *deterministic* in the sense that it does not require stochastic data $\widehat{y}_{n+1}$, in contrast to T^S. Therefore, we again have that $\text{Law}(v_\ell) = \mu_\ell$ for all $\ell \in \mathbb{Z}^+$, where the probability measure μ_n evolves according to

$$\widehat{\mu}_{n+1} = \mathsf{P}\mu_n, \tag{2.47a}$$

$$\mu_{n+1} = (T_n^D)_\sharp \widehat{\mu}_{n+1}. \tag{2.47b}$$

This is simply a restatement of (2.23), noting that $(T_n^D)_\sharp$ has been chosen so that the pushforward corresponds to the application of Bayes Theorem to incorporate data $y_{n+1}^\dagger$. In particular $\mathsf{T}_n^D(\widehat{\mu}_{n+1}) := (T_n^D)_\sharp \widehat{\mu}_{n+1}$ has property $\mathsf{T}_n^D(\widehat{\mu}_{n+1}) = \mathsf{L}_n(\widehat{\mu}_{n+1})$. The evolution (2.47) is another nonlinear Markov process, now because of the dependence of T_n^D on $\widehat{\mu}_{n+1}$. Again, the underlying sample-path representation (2.46) is not uniquely defined.

The two transport maps T^S and T^D introduce an important conceptual approach to algorithms for filtering, but determining the maps can be as hard, or harder, than solving the filtering problem itself. Thus, in the next two subsections, we turn to relaxations of the perfect transport effected by T^S and T^D. We instead seek mean field maps which match only first and second order moment information; this relaxation allows for approximate transport maps with simple affine (in senses to be made precise) forms. The perspective of matching first and second order moments naturally suggests working with Gaussians and hence we also relate the approximate transport to the Gaussian projected filter.

2.2.5.3 Second Order Transport – Motivation

To motivate the more general ideas behind second order transport, we first study an explicit example. It is well known how to transform samples from a unit centered Gaussian random variable on $\mathbb{R}$ into samples from a Gaussian random variable with mean $m \neq 0$ and variance $\sigma \neq 1$ by a simple scaling and shifting operation. An

appropriate generalization of such a procedure suggests consideration of the map

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.48a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}, \tag{2.48b}$$

$$v_{n+1} = m_{n+1} + C_{n+1}^{\frac{1}{2}} \widehat{C}_{n+1}^{-\frac{1}{2}} (\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}), \tag{2.48c}$$

where m_{n+1}, $\widehat{C}_{n+1}$, and C_{n+1} are determined by (2.31), (2.32), and (2.40), using (2.48a) and (2.48b). Here $\{y_n^\dagger\}$ arises again from a fixed realization of (2.1). This is a specific instance of the sample path perspective, and (2.28, 2.30) in particular. In this case the sample path evolution of v_n has law which only approximates the true filtering law.

The map $v_n \mapsto v_{n+1}$ defined by (2.48) is a mean field map because of the dependence of m_{n+1}, C_{n+1} and $\widehat{C}_{n+1}$ on $\mathsf{Q}\ \mathrm{Law}(v_n)$. It may be viewed as an approximation to (2.44) which is exact when $\mathsf{Q}\ \mathrm{Law}(v_n)$ is Gaussian. To demonstrate exactness on Gaussians it suffices to show that we obtain the desired mean and covariance after application of the map. It is clear from (2.48c) that $\mathbb{E}v_{n+1} = m_{n+1}$ and also that

$$\begin{aligned}
&\mathbb{E}\big((v_{n+1} - m_{n+1}) \otimes (v_{n+1} - m_{n+1})\big) \\
&\qquad = C_{n+1}^{\frac{1}{2}} \widehat{C}_{n+1}^{-\frac{1}{2}} \mathbb{E}\big((v_{n+1} - m_{n+1}) \otimes (v_{n+1} - m_{n+1})\big) \widehat{C}_{n+1}^{-\frac{1}{2}} C_{n+1}^{\frac{1}{2}} \\
&\qquad = C_{n+1}^{\frac{1}{2}} \widehat{C}_{n+1}^{-\frac{1}{2}} \widehat{C}_{n+1} \widehat{C}_{n+1}^{-\frac{1}{2}} C_{n+1}^{\frac{1}{2}} \\
&\qquad = C_{n+1}.
\end{aligned}$$

It is important to recognize that, in general, v_{n+1} defined by (2.48c) will not be Gaussian distributed since $\widehat{v}_{n+1}$, defined by (2.48a), will not be Gaussian either. However, although (2.48) does not provide a closed iteration on Gaussians, the map from $\widehat{v}_{n+1}$ to v_{n+1} agrees with the same step in the Gaussian projected filter, at the level of first and second order moments. But, because it is not a closed iteration on $\mathcal{G}(\mathbb{R}^{d_v})$, it is clearly not the same as the Gaussian projected filter.

Whilst the mean field map from (2.48) is a relatively transparent way to achieve the goal of matching first and second order moments in the transport step there is an uncountable set of ways of achieving this objective; the next two subsections demonstrate this, identifying all mean field maps effecting approximate transport from within two specific classes of affine transformations. We will then highlight a small subset that have been used in practice, each of which is useful in certain specific contexts.

2.2.5.4 Second Order Transport – Stochastic Case

The first class of approximate filters determined by mean field maps have the sample path form

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.49a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}, \tag{2.49b}$$

$$v_{n+1} = \widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi_{n+1}, y^\dagger_{n+1}), \tag{2.49c}$$

where $\{y^\dagger_n\}$ arises from a fixed realization of (2.1). We identify $\widetilde{T}^S_n : \mathbb{R}^{d_v} \times \mathbb{R}^{d_y} \to \mathbb{R}^{d_v}$, where $\widetilde{T}^S_n(\cdot) := \widetilde{T}^S(\cdot\,; \pi_{n+1}, y^\dagger_{n+1})$ approximates an exact transport map $T^S_n(\cdot) = T^S(\cdot\,; \pi_{n+1}, y^\dagger_{n+1})$, defined previously, by matching the first and second order moments.

We next introduce[6] $\pi = \mathrm{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ and assume that the exact and approximate transport maps satisfy, respectively,

$$(T^S_n)_\sharp \pi = \mathsf{B}_n(\pi), \tag{2.50a}$$

$$\mathsf{G}\big((\widetilde{T}^S_n)_\sharp \pi\big) = \mathsf{B}_n(\mathsf{G}\pi), \tag{2.50b}$$

for all measures π on the product space $\mathbb{R}^{d_v} \times \mathbb{R}^{d_y}$. Of course T^S_n and $\widetilde{T}^S_n$ will depend on π and then pushforward is to be interpreted as in (2.42), (2.43). It is intuitive that (2.50b) enforces map $\widetilde{T}^S_n$ to satisfy (2.50a) when π is Gaussian; we prove this in Lemma 2.2.11 below.

Perfect transport corresponds to asking that, for all measures π on the space $\mathbb{R}^{d_v} \times \mathbb{R}^{d_y}$, (2.50a) holds; second order transport relaxes this and asks only that (2.50b) holds. Whilst achieving (2.50a) may be harder than solving the filtering problem directly, we will show that achieving (2.50b) is straightforward and leads to computationally tractable methods. This gain in tractability comes at the price of only achieving (2.50b) in place of (2.50a); however it is intuitive that this price will not be high for settings in which the filtering distribution, and the predictive distribution on state and data, is not too far from Gaussian; we flesh out this idea in Subsection 2.2.5.6 below.

Working to satisfy (2.50b) allows us to find tractable approximate second order transport maps by seeking $\widetilde{T}^S$ in the form

$$\widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi, y^\dagger) := A\widehat{v}_{n+1} + B\widehat{y}_{n+1} + a. \tag{2.51}$$

[6] We temporarily drop explicit notational dependence on $n+1$ in π and in $y^\dagger$. This should not cause confusion as the approximate map we derive is concerned simply with finding a push forward which approximates conditioning of $\mathrm{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ on $\widehat{y}_{n+1} = y^\dagger$.

We allow the matrices/vectors A, B, a to depend on $(\pi, y^\dagger)$; however, they are assumed to be independent of $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$. Making this assumption ensures that the transport map is affine with respect to the realization of $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ (but not their law). This in turn leads to tractable computations to determine A, B, a on the basis of matching second order moments of perfect transport. In addition to computational tractability, the affine form of the transport map $\widetilde{T}^S$ is motivated by the following which shows that the approximate transport is perfect when applied to Gaussian source.

Lemma 2.2.11. *Consider approximate transport map $\widetilde{T}_n^S = \widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi, y^\dagger)$ with the form* (2.51)*, assumed to satisfy (2.50b). Then $\widetilde{T}^S$ depends on π only through $\mathsf{G}\pi$. Furthermore,*

$$(\widetilde{T}_n^S)_\sharp(\mathsf{G}\pi) = \mathsf{B}_n(\mathsf{G}\pi);$$

thus, if π is Gaussian, equation (2.50b) implies (2.50a). $\diamond$

Proof. We first note that (2.50b) is equivalent to insisting that

$$\mathsf{G}\big((\widetilde{T}_n^S)_\sharp \mathsf{G}\pi\big) = \mathsf{B}_n(\mathsf{G}\pi) \tag{2.52}$$

for all measures π; this follows because, noting the definition (2.42) and consequence (2.43), the first and second moments of $(\widetilde{T}_n^S)_\sharp \mathsf{G}\pi$ and $(\widetilde{T}_n^S)_\sharp \pi$ agree, because of the affine form (2.51) assumed for $\widetilde{T}_n^S$. Recall that $(\widetilde{T}_n^S)$ depends on $(\pi, y^\dagger) = \big(\mathrm{Law}(\widehat{v}_{n+1}, \widehat{y}_{n+1}), y_{n+1}^\dagger\big)$. From the identity (2.52), it is clear that $\widetilde{T}_n^S$ only depends on π through $\mathsf{G}\pi$ because changing $\pi \to \mathsf{G}\pi$ leaves the identity invariant, as $\mathsf{G} \circ \mathsf{G} = \mathsf{G}$. Now note that, because Gaussians are preserved under affine transformations,

$$\big(\widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi, y^\dagger)\big)_\sharp(\mathsf{G}\pi) = \mathsf{G}\Big(\big(\widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi, y^\dagger)\big)_\sharp \pi\Big),$$

or, in compact notational form,

$$(\widetilde{T}_n^S)_\sharp(\mathsf{G}\pi) = \mathsf{G}\big((\widetilde{T}_n^S)_\sharp \pi\big). \tag{2.53}$$

The desired display in the lemma is then immediate from (2.50b). $\square$

An affine transport map of the form (2.51), when combined with particle approximations, leads to practical implementable algorithms and achieves (2.50b) by ensuring that $(\widetilde{T}_n^S)_\sharp \pi$ has first and second moments which agree with those of the Gaussian

projected filter; these are given by equations (2.31), (2.32), and (2.40) when π is the law of $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$.

In Appendix B, Subsection 2.B.1, we identify the (uncountable) set of all possible A, B, a which achieve the desired matching of first and second order moments. Here we focus on the two specific choices given in Example 2.B.5 from that Appendix. The first that we highlight corresponds to the choice

$$\widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \widetilde{\pi}_{n+1}, y^\dagger_{n+1}) := \widehat{v}_{n+1} + K_n(y^\dagger_{n+1} - \widehat{y}_{n+1}),$$

with $K_n = K(\pi_{n+1})$ given by (2.33). Thus we obtain the following mean field dynamical system, which corresponds to (2.14) in the setting where the Kalman gain K_n is defined by (2.33):

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.54a}$$

$$\widehat{y}_{n+1} = h(\widehat{v}_{n+1}) + \eta_{n+1}, \tag{2.54b}$$

$$v_{n+1} = \widehat{v}_{n+1} + \widehat{C}^{vy}_{n+1}(\widehat{C}^{yy}_{n+1})^{-1}(y^\dagger_{n+1} - \widehat{y}_{n+1}), \tag{2.54c}$$

where $\{y^\dagger_n\}$ arises from a fixed realization of (2.1) and equations (2.31), (2.32) define the Kalman gain $K_n = \widehat{C}^{vy}_{n+1}(\widehat{C}^{yy}_{n+1})^{-1}$. We refer to this as *Kalman transport*, noting that it serves as a derivation of the Kalman gain, beyond the linear Gaussian setting. This is a specific instance of the sample path perspective, and (2.28, 2.29) in particular. Again, this is a case in which the sample path evolution for v_n has law which only approximates the true filtering law.

The second transport map from Example 2.B.5 corresponds to the choice

$$\widetilde{T}^S(\widehat{v}_{n+1}, \widehat{y}_{n+1}; \pi_{n+1}, y^\dagger_{n+1}) := m_{n+1} + C^{\frac{1}{2}}_{n+1}\widehat{C}^{-\frac{1}{2}}_{n+1}(\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}),$$

leading to the mean field map (2.48), recalling that m_{n+1}, $\widehat{C}_{n+1}$, and C_{n+1} are determined by (2.31), (2.32), and (2.40), using (2.48a) and (2.48b).

Remark 2.2.12. *One important difference between the mean field models* (2.54) *and* (2.48) *is that the former involves inversion of matrices in data space, and the latter in state space. The relative dimensions of the two spaces plays a role in determining which mean field model is more appropriate as the basis of algorithms. A second noteable difference is that the mean field model* (2.48) *does not require generation of the stochastic data* $\widehat{y}_{n+1}$*. This is because we may employ the identity* $\mathbb{E}\widehat{y}_{n+1} = \mathbb{E}h(\widehat{v}_{n+1})$ *and use* (2.34), (2.35) *to compute* m_{n+1}, C_{n+1} *and* $\widehat{C}_{n+1}$*. Motivated by this observation, the next subsection studies a wide class of approximate transport maps with the property that they do not require generation of stochastic data.* ■

2.2.5.5 Second Order Transport – Deterministic Case

We now turn our attention to approximate filters defined by deterministic mean field maps. We seek to approximate the exact transport (2.46) by mean field maps with the sample path form

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.55a}$$

$$v_{n+1} = \widetilde{T}^D(\widehat{v}_{n+1}; \widehat{\mu}_{n+1}, y^\dagger_{n+1}), \tag{2.55b}$$

where $\{y^\dagger_n\}$ arises from a fixed realization of (2.1). As in the previous subsection we drop explicit n-dependence on the measure $\widehat{\mu}_{n+1}$ and on the data $y^\dagger_{n+1}$ when no confusion arises from doing so. To this end we define, for $\widehat{v}_{n+1}$ given by (2.28b), $\widehat{\mu} = \mathrm{Law}(\widehat{v}_{n+1})$ and $y^\dagger = y^\dagger_{n+1}$. In the following $T^D_n : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ is defined by $T^D_n(\cdot) = T^D(\cdot; \widehat{\mu}, y^\dagger)$ and $\widetilde{T}^D_n : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ is defined by $\widetilde{T}^D_n(\cdot) = \widetilde{T}^D(\cdot; \widehat{\mu}, y^\dagger)$; this is a useful notational convention for the reasons explained in the stochastic transport setting.

Analogously to the identities (2.50) in the previous subsection, we seek an approximation $\widetilde{T}^D$ which, in comparison with the true transport map T^D, satisfies

$$(T^D_n)_\sharp \widehat{\mu} = \mathsf{B}_n(\mathsf{Q}\widehat{\mu}), \tag{2.56a}$$

$$\mathsf{G}\big((\widetilde{T}^D_n)_\sharp \widehat{\mu}\big) = \mathsf{B}_n(\mathsf{G}\mathsf{Q}\widehat{\mu}), \tag{2.56b}$$

for all measures $\widehat{\mu}$ on the state space $\mathbb{R}^{d_v}$. As in the previous subsection, where we studied approximate stochastic transport, we again seek maps with a specific affine form. Concretely, the maps are assumed to be affine in the pair $\big(\widehat{v}_{n+1}, \widehat{h}_{n+1}\big)$, with $\widehat{h}_{n+1} = h(\widehat{v}_{n+1})$, leading to the assumed form $\widetilde{T}^D_n(\cdot) = \widetilde{T}^D(\cdot; \mu, y^\dagger)$ with

$$\widetilde{T}^D(\widehat{v}_{n+1}, \widehat{h}_{n+1}; \mu, y^\dagger) := R\widehat{v}_{n+1} + S\widehat{h}_{n+1} + r, \tag{2.57}$$

for $(\widehat{\mu}, y^\dagger)-$dependent matrices/vectors R, S, r of appropriate dimensions. Note, however, that R, S, r are assumed to be independent of the realization $(\widehat{v}_{n+1}, \widehat{h}_{n+1})$, depending only on its law, so that the transport map is affine in $(\widehat{v}_{n+1}, \widehat{h}_{n+1})$. With this restriction, which will lead to practical implementable algorithms, we simply ask that (2.56b) holds: the first and second moments of the output map agree with those of the Gaussian projected filter, given by equations (2.31), (2.34), and (2.41).

As in the previous subsection, there are uncountably many choices of R, S, r which we identify in Appendix B, Subsection 2.B.2; Example 2.B.12 highlights two important cases. The first coincides with (2.48) since $S = 0$, but the second leads to a

new mean field map. To formulate this new map we first define $\widetilde{K}_n = \widetilde{K}(\widehat{\mu})$ by

$$\widetilde{K}_n = \widehat{C}^{vh}_{n+1}\Big((\widehat{C}^{hh}_{n+1} + \Gamma) + \Gamma^{1/2}(\widehat{C}^{hh}_{n+1} + \Gamma)^{1/2}\Big)^{-1}. \tag{2.58}$$

We then make the choice

$$\widetilde{T}^D(\widehat{v}_{n+1}, \widehat{h}_{n+1}; \widehat{\mu}, y^\dagger) := \widehat{v}_{n+1} - \widetilde{K}_n(\widehat{h}_{n+1} - \mathbb{E}\widehat{h}_{n+1}) + K_n(y^\dagger - \mathbb{E}\widehat{h}_{n+1}),$$

with K_n given by (2.35) and repeated here for convenience:

$$K_n = \widehat{C}^{vh}_{n+1}\left(\widehat{C}^{hh}_{n+1} + \Gamma\right)^{-1}.$$

The second mean field map identified in Example 2.B.12 is

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.59a}$$

$$\widehat{h}_{n+1} = h(\widehat{v}_{n+1}), \tag{2.59b}$$

$$v_{n+1} = \widehat{v}_{n+1} - \widetilde{K}_n(\widehat{h}_{n+1} - \mathbb{E}\widehat{h}_{n+1}) + K_n(y^\dagger_{n+1} - \mathbb{E}\widehat{h}_{n+1}), \tag{2.59c}$$

where K_n and $\widetilde{K}_n$ are computed under $\mathrm{Law}(\widehat{v}_{n+1})$. This is another instance of the sample path perspective, and (2.28, 2.30) in particular. Again this sample path evolution for v_n has law which only approximates the true filtering law.

Remark 2.2.13. *If the ensemble spread is such that the size of $\widehat{C}^{hh}_{n+1}$ is much smaller than the size of the observational covariance Γ then we may invoke the approximation $\widehat{C}^{hh}_{n+1} + \Gamma \approx \Gamma$. With this approximation it follows that $\widetilde{K}_n \approx \frac{1}{2}K_n$ in (2.58). Some deterministic ensemble Kalman filters are derived from mean field dynamics which exploit this approximation by setting $\widetilde{K}_n = \frac{1}{2}K_n$ in (2.59). We then replace (2.59c) by the compact update step*

$$v_{n+1} = \widehat{v}_{n+1} + K_n\left(y^\dagger_{n+1} - \frac{1}{2}\big(\mathbb{E}\widehat{h}_{n+1} + \widehat{h}_{n+1}\big)\big)\right). \tag{2.60}$$

Such a formulation corresponds to the control-theoretic perspective of (2.14), with K_n given by (2.35) *and the innovation (2.15) replaced by*

$$\mathfrak{I}_n = y^\dagger_{n+1} - \frac{1}{2}\big(\mathbb{E}\widehat{h}_{n+1} + \widehat{h}_{n+1}\big). \tag{2.61}$$

Filters based on this mean field dynamics thus invoke an additional approximation of perfect transport, over and above that stemming from matching only first and second moments: they assume further that the observational noise dominates ensemble variation. We note that in the continuous time limit this form of the innovation arises naturally and does not constitute an additional approximation. ■

2.2.5.6 Second Order Transport – Summary

It is helpful at this point to take stock of two approximations to filtering that we have introduced, Gaussian projected filtering and approximate transport, and discuss their inter-relations. For simplicity we do this in the context of mean field stochastic maps, but similar considerations extend to mean field deterministic maps. In this subsection we also include Example 2.2.15, demonstrating the existence of mean field maps for the Kalman filter which, recall, applies only in the linear Gaussian setting.

Recall that

$$\mu_{n+1} = \mathsf{B}_n(\mathsf{QP}\mu_n), \quad \mu_0 = \mathsf{N}(m_0, C_0), \tag{2.62a}$$

$$\mu^G_{n+1} = \mathsf{B}_n(\mathsf{GQP}\mu^G_n), \quad \mu^G_0 = \mathsf{N}(m_0, C_0), \tag{2.62b}$$

With the goal of discussing the inter-relations between Gaussian projected filtering and approximate transport methods, we let μ^{MF} denote the measure associated with using the mean field map $\widetilde{T}^S_n$ to approximate the conditioning step in (2.21). Using (2.50b) to rewrite the Gaussian projected filter, and using the construction of the stochastic mean field model (2.49), we obtain

$$\mu^G_{n+1} = \mathsf{G}\big((\widetilde{T}^S_n)_\sharp(\mathsf{QP}\mu^G_n)\big), \quad \mu^G_0 = \mathsf{N}(m_0, C_0), \tag{2.63a}$$

$$\mu^{MF}_{n+1} = (\widetilde{T}^S_n)_\sharp(\mathsf{QP}\mu^{MF}_n), \quad \mu^{MF}_0 = \mathsf{N}(m_0, C_0). \tag{2.63b}$$

Remark 2.2.14. *Equations* (2.63) *show that* $\{\mu^{MF}_n\}$ *is close to* $\{\mu^G_n\}$*, if the Gaussian projection in (2.63a) is close to the identity where it acts on the output of onestep. Equations* (2.62) *show that* $\{\mu^G_n\}$ *is close to* $\{\mu_n\}$ *if the Gaussian projection in (2.62b) is close to the identity where it acts on the joint space of state and observation. Together these two facts suggest that* $\{\mu^{MF}_n\}$*,* $\{\mu^G_n\}$ *and* $\{\mu_n\}$ *are all close to one another if the two Gaussian projections can be viewed as being close to the identity map, where they appear in* (2.62) *and in* (2.63)*. This provides a potential path for analysis of the mean field model, away from the linear setting where it is exact. Note also that (2.63a) shows that the Gaussian projected filter evolves within the manifold of Gaussian probability measures; the mean field model (2.63b) does not.*

Example 2.2.15. *Assume that* $v_0 \sim \mathsf{N}(m_0, C_0)$*, that* $\Gamma \succ 0$ *and consider the Kalman filter setting of Example 2.2.6; in particular* (2.6) *prevails rendering* Ψ *and h linear.*

The mean field stochastic dynamical system (2.54) *then takes the form*

$$\widehat{v}_{n+1} = Mv_n + \xi_n, \tag{2.64a}$$

$$\widehat{y}_{n+1} = H\widehat{v}_{n+1} + \eta_{n+1}, \tag{2.64b}$$

$$v_{n+1} = \widehat{v}_{n+1} + \widehat{C}_{n+1}H^\top (H\widehat{C}_{n+1}H^\top + \Gamma)^{-1}(y^\dagger_{n+1} - \widehat{y}_{n+1}), \tag{2.64c}$$

where $\{y^\dagger_n\}$ *arises from a fixed realization of* (2.24) *and where* C_n *is the covariance of* v_n *and* $\widehat{C}_{n+1} = MC_nM^\top + \Sigma$ *is the covariance of* $\widehat{v}_{n+1}$*. The resulting dynamics give a sample path representation of the Kalman filter in that* $v_n \sim \mathsf{N}(m_n, C_n)$ *where* m_n, C_n *are as given in Example 2.2.6. This follows because the map defined by* (2.64) *is well-defined, since* $\Gamma \succ 0$*. Lemma 2.2.11 shows that the approximate transport is exact in this Gaussian setting.*

Similar ideas can be applied to (2.48) *and* (2.59) *to determine other mean field models with law equal to that of the Kalman filter. Furthermore we observe that the formulation based on* (2.48) *can be symmetrized to obtain, in the linear Gaussian setting of* (2.6)*,*

$$\widehat{v}_{n+1} = Mv_n + \xi_n,\ n \in \mathbb{Z}^+, \tag{2.65a}$$

$$v_{n+1} = m_{n+1} + A_n(\widehat{v}_{n+1} - \widehat{m}_{n+1}), \tag{2.65b}$$

$$A_n = (C_{n+1})^{1/2}\left[(C_{n+1})^{1/2}\widehat{C}_{n+1}(C_{n+1})^{1/2}\right]^{-1/2}(C_{n+1})^{1/2}, \tag{2.65c}$$

with $(\widehat{m}_{n+1}, \widehat{C}_{n+1}, \widehat{C}^{vh}_{n+1}, \widehat{C}^{hh}_{n+1})$ *defined by* (2.31)*,* (2.34) *and* (m_{n+1}, C_{n+1}) *defined by* (2.40)*. We note that the second component of the map may be written in gradient form and corresponds to an optimal transport from* $\widehat{\mu}_{n+1}$ *into* μ_{n+1} *in the sense of the Euclidean Wasserstein distance of optimal transportation (see Subsection 2.2.7 for details); indeed this is true for an entire family of weighted Wasserstein distances. To recognize the gradient structure define*

$$\Phi_n(v) := \langle m_{n+1}, v\rangle + \frac{1}{2}\langle A_n(v - \widehat{m}_{n+1}), v - \widehat{m}_{n+1}\rangle$$

and note that then

$$\widehat{v}_{n+1} = Mv_n + \xi_n,\ n \in \mathbb{Z}^+, \tag{2.66a}$$

$$v_{n+1} = \nabla\Phi_n(\widehat{v}_{n+1}). \tag{2.66b}$$

■

2.2.6 Ensemble Kalman Methods

The mean field formulations from Subsection 2.2.5 provide clear insights into many of the design choices and mechanisms that underlie ensemble Kalman methods. In this subsection we take the mean field models and use particle approximations to derive implementable numerical algorithms. When approximated by interacting particle systems, the mean field formulations of ensemble Kalman methods lead to actionable algorithms.

We start, in Subsection 2.2.6.1, with the setting in which transport is perfect; these algorithms are not, in general, implementable since determining perfect transport is itself a difficult computational task and the subject of ongoing research; see the bibliography Subsection 2.2.7. Thus we turn to particle approximations of the transports designed to match first and second order statistics. This leads to the (stochastic) *ensemble Kalman filter* in Subsection 2.2.6.2, and to (deterministic) *ensemble square root filters* in Subsection 2.2.6.3. The methods derived in this subsection involve approximating the Kalman gain by computing covariances under the empirical measure defined by the ensemble of particles. To avoid overloading notation, in the rest of this subsection C_{n+1}, $\widehat{C}_{n+1}$, $\widehat{C}^{vy}_{n+1}$, and $\widehat{C}^{yy}_{n+1}$ will denote covariances computed with expectation under the empirical measure. With this notation in place, in the rest of this specific subsection, K_n will directly refer to the particle approximation of the Kalman gain, computed using the covariances with respect to the empirical measure, without further specification needed. Throughout this section $\{y_n^\dagger\}$ arises from a fixed realization of (2.1).

2.2.6.1 Perfect Particle Filters

The mean field equations (2.44) could, in principle, be approximated through a particle approximation of the mean field leading to the following conceptual (because map T_n^S is not known explicitly) algorithm: let $\mathsf{J} = \{1, \cdots, J\}$ and consider, for $(n, j) \in \mathbb{Z}^+ \times \mathsf{J}$, the interacting particle dynamical system

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j} + \xi_n^{(j)}, \tag{2.67a}$$

$$\widehat{y}_{n+1}^{(j)} = h(\widehat{v}_{n+1}^{(j} + \eta_{n+1}^{(j)}, \tag{2.67b}$$

$$v_{n+1}^{(j)} = T^S(\widehat{v}_{n+1}^{(j)}, \widehat{y}_{n+1}^{(j)}; \pi_{n+1}^{\mathsf{J}}, y_{n+1}^\dagger), \tag{2.67c}$$

$$\pi_{n+1}^{\mathsf{J}} = \frac{1}{J}\sum_{j=1}^{J} \delta_{(\widehat{v}_{n+1}^{(j)}, \widehat{y}_{n+1}^{(j)})}. \tag{2.67d}$$

This evolves the particles $\{v_n^{(j)}\}_{j\in\mathsf{J}}$ into $\{v_{n+1}^{(j)}\}_{j\in\mathsf{J}}$. Here the $\{\xi_n^{(j)}\}$ are, for each j, random variables given by the known distribution of ξ_n specified in (2.2) and, furthermore, are drawn independently with respect to each $(n, j) \in \mathbb{Z}^+ \times \mathsf{J}$. Similar considerations apply to the $\{\eta_n^{(j)}\}$ which, additionally, are independent of the $\{\xi_n^{(j)}\}$. It is intuitive that the large J limit of this system recovers the mean field dynamics (2.44) and, in particular,

$$\mu_n^{\mathsf{J}} = \frac{1}{J}\sum_{j=1}^{J} \delta_{v_n^{(j)}} \approx \mu_n. \tag{2.68}$$

Applying a similar idea to (2.46) leads to the following conceptual (because map T_n^D is not known explicitly) algorithm. Consider, for $(n, j) \in \mathbb{Z}^+ \times \mathsf{J}$, the interacting particle dynamical system

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j)}) + \xi_n^{(j)}, \tag{2.69a}$$

$$v_{n+1}^{(j)} = T^D(\widehat{v}_{n+1}^{(j)}; \widehat{\mu}_{n+1}^{\mathsf{J}}, y_{n+1}^{\dagger}), \tag{2.69b}$$

$$\widehat{\mu}_{n+1}^{\mathsf{J}} = \frac{1}{J}\sum_{j=1}^{J} \delta_{\widehat{v}_{n+1}^{(j)}}. \tag{2.69c}$$

This evolves the particles $\{v_n^{(j)}\}_{j\in\mathsf{J}}$ into $\{v_{n+1}^{(j)}\}_{j\in\mathsf{J}}$. The same assumptions are made about the $\{\xi_n^{(j)}\}$ as for the preceding interacting particle dynamical system. It is again intuitive that the large J limit of this system recovers the mean field dynamics (2.46), and the evolution (2.47). In particular it is intuitive that (2.68) holds for this particle approximation too. We reiterate that in practice these algorithms are, in general, not easy to use. More specifically, finding particle based approximations $T^{S,J}$ and $T^{D,J}$ to the desired transport maps such that

$$\lim_{J\to\infty} T^{S,J} = T^S, \qquad \lim_{J\to\infty} T^{D,J} = T^D$$

in an appropriate sense is a computationally challenging task and the subject of ongoing research. This leads to the next two subsections in which we replace T^S and T^D, in the interacting particles systems (2.67) and (2.69), by the previously introduced affine approximate transports $\widetilde{T}^S$ and $\widetilde{T}^D$, respectively.

2.2.6.2 Stochastic Ensemble Kalman Filters

Particle approximation of the mean field dynamical system (2.54), effecting Kalman transport, bring us to the stochastic EnKF (*ensemble Kalman filter*). This method may be derived by writing down a particle approximation of the mean field stochastic

dynamics defined by (2.54). We evolve the particles $\{v_n^{(j)}\}_{j\in\mathsf{J}}$ into $\{v_{n+1}^{(j)}\}_{j\in\mathsf{J}}$ according to the following stochastic interacting particle system, holding for $(n, j) \in \mathbb{Z}^+\times\mathsf{J}$:

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j)}) + \xi_n^{(j)},\ n \in \mathbb{Z}^+, \tag{2.70a}$$
$$\widehat{y}_{n+1}^{(j)} = h(\widehat{v}_{n+1}^{(j)}) + \eta_{n+1}^{(j)},\ n \in \mathbb{Z}^+, \tag{2.70b}$$
$$v_{n+1}^{(j)} = \widehat{v}_{n+1}^{(j)} + K_n\big(y_{n+1}^\dagger - \widehat{y}_{n+1}^{(j)}\big), \tag{2.70c}$$
$$\pi_{n+1}^{\mathsf{J}} = \frac{1}{J}\sum_{j=1}^{J} \delta_{(\widehat{v}_{n+1}^{(j)}, \widehat{y}_{n+1}^{(j)})}. \tag{2.70d}$$

Here the Kalman gain from (2.33) is approximated using the empirical measure π_{n+1}^{J}, but is still denoted by K_n to avoid proliferation of notation; details follow below. The same assumptions regarding $\{\xi_n^{(j)}\}$ and $\{\eta_{n+1}^{(j)}\}$ are made as for (2.67). We let $\mathbb{E}_n^{\mathsf{J}}$ denote expectation under π_n^{J}. For the basic implementation of EnKF (2.70) the desired covariance matrices, and Kalman gain (2.33), are then approximated by expectation under π_{n+1}^{J}, so that [7]

$$\begin{aligned}
\widehat{C}_{n+1}^{vy} &= \mathbb{E}_{n+1}^{\mathsf{J}}\Big(\big(\widehat{v}_{n+1} - \mathbb{E}_{n+1}^{\mathsf{J}}\widehat{v}_{n+1}\big) \otimes \big(\widehat{y}_{n+1} - \mathbb{E}_{n+1}^{\mathsf{J}}\widehat{y}_{n+1}\big)\Big),\\
\widehat{C}_{n+1}^{yy} &= \mathbb{E}_{n+1}^{\mathsf{J}}\Big(\big(\widehat{y}_{n+1} - \mathbb{E}_{n+1}^{\mathsf{J}}\widehat{y}_{n+1}\big) \otimes \big(\widehat{y}_{n+1} - \mathbb{E}_{n+1}^{\mathsf{J}}\widehat{y}_{n+1}\big)\Big),\\
K_n &= \widehat{C}_{n+1}^{vy}\big(\widehat{C}_{n+1}^{yy}\big)^{-1}.
\end{aligned}$$

Note that a pseudo-inverse may be required to define K_n. An alternative, avoiding pseudo-inverse, is to use a particle approximation in formula (2.35b), leading to the use of

$$\widehat{C}_{n+1}^{vh} = \mathbb{E}_{n+1}^{\mathsf{J}}\Big(\big(\widehat{v}_{n+1} - \mathbb{E}_{n+1}^{\mathsf{J}}\widehat{v}_{n+1}\big) \otimes \big(h(\widehat{v}_{n+1}) - \mathbb{E}_{n+1}^{\mathsf{J}}h(\widehat{v}_{n+1})\big)\Big), \tag{2.71a}$$
$$\widehat{C}_{n+1}^{hh} = \mathbb{E}_{n+1}^{\mathsf{J}}\Big(\big(h(\widehat{v}_{n+1}) - \mathbb{E}_{n+1}^{\mathsf{J}}h(\widehat{v}_{n+1})\big) \otimes \big(h(\widehat{v}_{n+1}) - \mathbb{E}_{n+1}^{\mathsf{J}}h(\widehat{v}_{n+1})\big)\Big), \tag{2.71b}$$
$$K_n = \widehat{C}_{n+1}^{vh}\big(\widehat{C}_{n+1}^{hh} + \Gamma\big)^{-1} \tag{2.71c}$$

to compute the gain K_n. The advantage of this latter formulation is that it ensures positivity, and hence invertibility, of the covariance in data space, if Γ is assumed positive-definite. It is hence typically preferred.

Example 2.2.16. *We return to the set-up of Example 2.2.3, and now demonstrate performance of the stochastic EnKF on the same Lorenz '96 model. Indeed, we again*

[7] The empirical covariance computations are often modified to accommodate the widely adopted convention of scaling by $1/(J-1)$, instead of $1/J$, in view of the matrix being computed from $J-1$ independent increments about the mean.

study the Lorenz '96 (singlescale) model for unknown $v \in C(\mathbb{R}^+, \mathbb{R}^L)$ *satisfying the equations* (2.9) *with* $L = 9, h_v = -0.8$ *and* $F = 10$ *and function* m *as shown in Figure Figure 2.1. We consider observations* $\{y_n^\dagger\}_{n\in\mathbb{Z}^+}$ *arising from the model*

$$v_{n+1}^\dagger = \Psi(v_n^\dagger) + \xi_n^\dagger,$$
$$y_{n+1}^\dagger = h(v_{n+1}^\dagger) + \eta_{n+1}^\dagger,$$

where Ψ *is the solution operator for* (2.9) *over the observation time interval* τ*, and* $\{\xi_n^\dagger\}_{n\in\mathbb{Z}^+}$*,* $\{\eta_n^\dagger\}_{n\in\mathbb{N}}$ *are mutually independent Gaussian sequences defined by*

$$\xi_n^\dagger \sim \mathsf{N}(0, \sigma^2 I) \text{ i.i.d.}, \quad \eta_n^\dagger \sim \mathsf{N}(0, \gamma^2 I) \text{ i.i.d.},$$

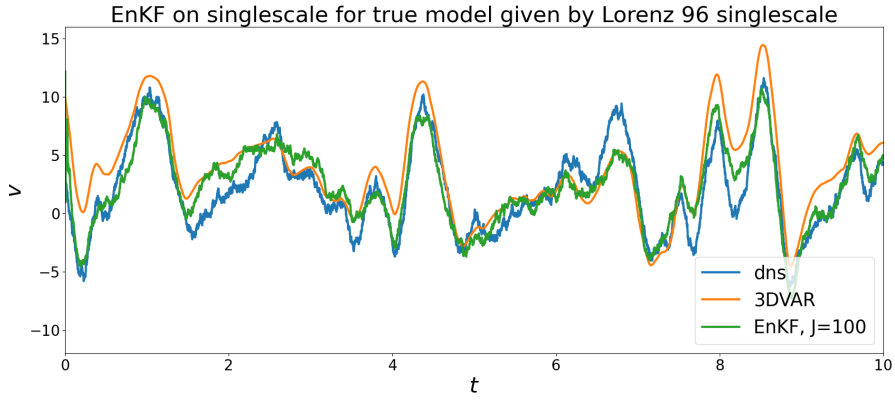
with $\sigma = 0.1$ *and* $\gamma = 0.1$*. We again assume that the observation function is linear:* $h(v) = Hv$ *for matrix* $H : \mathbb{R}^9 \to \mathbb{R}^6$ *defined by* (2.10)*.*

Figures Figure 2.5a and Figure 2.5b demonstrate the performance of stochastic EnKF in this experimental setting with $\tau = 10^{-3}$ *and using* $J = 10^2$ *and* $J = 5 \cdot 10^2$*, respectively, against the performance of 3DVAR with no noise; note that the EnKF uses a time-varying estimate of the gain* K_n*, whilst 3DVAR uses the fixed* K *given in Example 2.2.3. These experiments illustrate that using sufficiently large ensembles, the ensemble Kalman filter outperforms 3DVAR on such a nonlinear filtering problem where the true state and observational noise levels are high. Here* outperforms *refers to mean square error in recovery of the state. To quantitatively demonstrate this improvement, we compute: (a) the mean squared error between the estimates yielded by 3DVAR and the true states; and (b) the mean squared error between the ensemble mean of stochastic EnKF and the true states. In particular we report time-averaged mean squared errors obtained from both 3DVAR and stochastic EnKF given by use of formula* (2.16) *from Example 2.2.5 using* $t^* = 3$ *and* $T = 10$*. An ensemble size of* $J = 10^2$ *yields* $e_{EnKF} = 1.05 \cdot 10^0$*, while for* $J = 5 \cdot 10^2$ *we obtain* $e_{EnKF} = 5.24 \cdot 10^{-1}$*. For comparison, the error obtained using 3DVAR is* $e_{3DVAR} = 1.85 \cdot 10^0$*.* ■

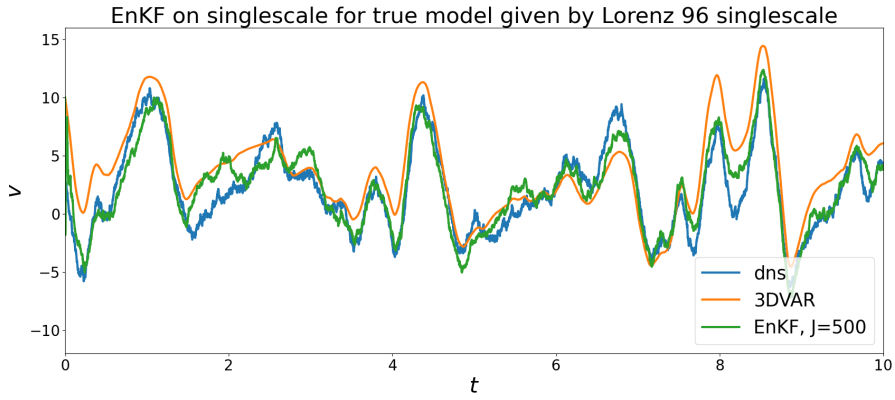
2.2.6.3 Ensemble Square Root Filters

The variants on the EnKF described in this subsection are known as ensemble square root filters; they are based on mean field maps (2.48) and (2.59), approximated by interacting particle systems.

Remark 2.2.17. *Square root filters are sometimes referred to as deterministic ensemble Kalman filters, to distinguish them from the ensemble Kalman filters described in*



(a) EnKF with ensemble size $J = 10^2$.



(b) EnKF with ensemble size $J = 5 \cdot 10^2$.

Figure 2.5: In this experiment we set the noise levels $\sigma = 10^{-1}, \gamma = 10^{-1}$. We display the estimates of v_3 in time produced by EnKF (using ensemble average) and 3DVAR against the true dynamics using observation time interval $\tau = 10^{-3}$. Again "dns" refers to direct numerical simulation. The results show that the EnKF provides a more accurate estimate of the trajectory, albeit at higher cost in terms of number of model evaluations.

the preceding subsection (see discussion of this, and bibliographic information, in Subsection 2.1.1.) However we have already used the terminology "stochastic" and "deterministic" to distinguish between different variants on the mean field models that we describe in Subsections Subsection 2.2.5.4 and Subsection 2.2.5.5 respectively. With the exception of the discussion in Subsection 2.1.1, we simply refer to square root filters for the methods introduced in this subsection. We note however that they are deterministic in the sense that they do not require generation of random variables.

We introduce two families of square root filters: of adjustment and transform type.

We emphasize again that the choice of which method to use in practice is determined by implementation details such as the number of particles J, the dimension of state space d_v, and the dimension of the observation space d_y. These implementation details, although important, are not the focus of this chapter. ■

Ensemble Adjustment Kalman Filters We introduce two different particle-based approximations of mean field models. Both methods are examples of a general class of algorithms known as *ensemble adjustment Kalman filters*: EAKF. The starting point for the first of these EAKF methods is the mean field map (2.48a), (2.48c), repeated here for convenience:

$$\widehat{v}_{n+1} = \Psi(v_n) + \xi_n, \tag{2.72a}$$

$$v_{n+1} = m_{n+1} + C_{n+1}^{\frac{1}{2}} \widehat{C}_{n+1}^{-\frac{1}{2}} (\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}), \tag{2.72b}$$

where m_{n+1}, $\widehat{C}_{n+1}$, and C_{n+1} are determined by (2.31), (2.34), (2.35), and (2.40).[8]

As in the previous subsection the methods evolve particle ensemble $\{v_n^{(j)}\}_{j\in\mathsf{J}}$ into $\{v_{n+1}^{(j)}\}_{j\in\mathsf{J}}$, via the predictive ensemble $\{\widehat{v}_{n+1}^{(j)}\}_{j\in\mathsf{J}}$. However they do not employ simulated data $\{\widehat{y}_{n+1}^{(j)}\}_{j\in\mathsf{J}}$, rather they make use of $\{\widehat{h}_{n+1}^{(j)}\}_{j\in\mathsf{J}}$, where $\widehat{h}_n^{(j)} = h(\widehat{v}_n^{(j)})$. To define the methods it helps to introduce new notation. Slightly modifying the notation in the preceding subsection, we now let $\mathbb{E}_n^{\mathsf{J}}$ denote expectation with respect to the empirical measure

$$\widehat{\mu}_n^{\mathsf{J}} = \frac{1}{J} \sum_{j=1}^{J} \delta_{\widehat{v}_n^{(j)}}.$$

We let $\widehat{v}_{n+1}$ denote the random variable with this distribution and, as before, $\widehat{h}_{n+1} = h(\widehat{v}_{n+1})$.

Next we define matrix $\widehat{V}_n$ comprising scaled ensemble deviations in state space:

$$\widehat{V}_n = \frac{1}{\sqrt{J}} \left(\widehat{v}_n^{(1)} - \mathbb{E}_n^{\mathsf{J}} \widehat{v}_n, \widehat{v}_n^{(2)} - \mathbb{E}_n^{\mathsf{J}} \widehat{v}_n, \cdots, \widehat{v}_n^{(J)} - \mathbb{E}_n^{\mathsf{J}} \widehat{v}_n \right) \in \mathbb{R}^{d_v \times J},$$

we then define the analogous matrix $\widehat{H}_n$ in observation space

$$\widehat{H}_n = \frac{1}{\sqrt{J}} \left(\widehat{h}_n^{(1)} - \mathbb{E}_n^{\mathsf{J}} \widehat{h}_n, \widehat{h}_n^{(2)} - \mathbb{E}_n^{\mathsf{J}} \widehat{h}_n, \cdots, \widehat{h}_n^{(J)} - \mathbb{E}_n^{\mathsf{J}} \widehat{h}_n \right) \in \mathbb{R}^{d_y \times J}.$$

[8] Although the identity (2.40) is derived in a subsection concerning the Gaussian projected filter, it is contained in Lemma 2.2.9 which simply concerns conditioning of Gaussians.

With these notations in hand, we have, with expectations computed under $\mathbb{E}_n^{\mathrm{J}}$,

$$\widehat{C}_n^{vh} = \widehat{V}_n \widehat{H}_n^\top, \tag{2.73a}$$

$$\widehat{C}_n^{hh} = \widehat{H}_n \widehat{H}_n^\top, \tag{2.73b}$$

and the ensemble-based approximation of the Kalman gain matrix is [9]

$$K_n = \widehat{V}_{n+1} \widehat{H}_{n+1}^\top (\widehat{H}_{n+1} \widehat{H}_{n+1}^\top + \Gamma)^{-1}. \tag{2.74}$$

This is a linear algebraic reformulation of the Kalman gain approximation resulting from (2.71). By making a particle approximation of (2.72), using (2.31), (2.34), (2.35), and (2.40), we obtain

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j)}) + \xi_n^{(j)},\ n \in \mathbb{Z}^+, \tag{2.75a}$$

$$\widehat{m}_{n+1} = \mathbb{E}_{n+1}^{\mathrm{J}} \widehat{v}_{n+1}, \tag{2.75b}$$

$$m_{n+1} = \widehat{m}_{n+1} + K_n (y_{n+1}^\dagger - \mathbb{E}_{n+1}^{\mathrm{J}} \widehat{h}_{n+1}), \tag{2.75c}$$

$$v_{n+1}^{(j)} = m_{n+1} + C_{n+1}^{\frac{1}{2}} \widehat{C}_{n+1}^{-\frac{1}{2}} \left(\widehat{v}_{n+1}^{(j)} - \widehat{m}_{n+1} \right), \tag{2.75d}$$

where K_n is given by (2.74); furthermore, $\widehat{C}_{n+1}$ and C_{n+1} are computed empirically using

$$\widehat{C}_{n+1} = \widehat{V}_{n+1} \widehat{V}_{n+1}^\top, \quad C_{n+1} = \widehat{V}_{n+1} \left(I + \widehat{H}_{n+1}^\top \Gamma^{-1} \widehat{H}_{n+1} \right)^{-1} \widehat{V}_{n+1}^\top. \tag{2.76}$$

The first of these two formulae follows similarly to (2.73); the second of these two formulae is derived as follows: [10]

$$C_{n+1} = \widehat{C}_{n+1} - \widehat{C}_{n+1}^{vh} (\widehat{C}_{n+1}^{hh} + \Gamma)^{-1} (\widehat{C}_{n+1}^{vh})^\top \tag{2.77a}$$

$$= \widehat{V}_{n+1} \widehat{V}_{n+1}^\top - \widehat{V}_{n+1} \widehat{H}_{n+1}^\top (\widehat{H}_{n+1} \widehat{H}_{n+1}^\top + \Gamma)^{-1} \widehat{H}_{n+1} \widehat{V}_{n+1}^\top \tag{2.77b}$$

$$= \widehat{V}_{n+1} \left(I - \widehat{H}_{n+1}^\top (\widehat{H}_{n+1} \widehat{H}_{n+1}^\top + \Gamma)^{-1} \widehat{H}_{n+1} \right) \widehat{V}_{n+1}^\top \tag{2.77c}$$

$$= \widehat{V}_{n+1} \left(I + \widehat{H}_{n+1}^\top \Gamma^{-1} \widehat{H}_{n+1} \right)^{-1} \widehat{V}_{n+1}^\top. \tag{2.77d}$$

The EAKF (2.75) takes as starting point (2.48). If instead we apply a particle approximation to the mean field dynamical system (2.59) we obtain a second version

[9] Here too, the empirical covariance computations are often modified to accommodate the widely adopted convention of scaling by $1/(J-1)$, instead of $1/J$.

[10] Using, in the last line, the identity $I - W^\top (WW^\top + I)^{-1} W = (I + W^\top W)^{-1}$ which holds for all (not necessarily square) matrices W.

of the EAKF:

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j)}) + \xi_n^{(j)},\ n \in \mathbb{Z}^+, \tag{2.78a}$$
$$\widehat{h}_{n+1}^{(j)} = h(\widehat{v}_{n+1}^{(j)}), \tag{2.78b}$$
$$\widehat{m}_{n+1} = \mathbb{E}_{n+1}^{\mathsf{J}} \widehat{v}_{n+1}, \tag{2.78c}$$
$$m_{n+1} = \widehat{m}_{n+1} + K_n(y_{n+1}^{\dagger} - \mathbb{E}_{n+1}^{\mathsf{J}} \widehat{h}_{n+1}), \tag{2.78d}$$
$$v_{n+1}^{(j)} = m_{n+1} + (\widehat{v}_{n+1}^{(j)} - \widehat{m}_{n+1}) - \widetilde{K}_n(\widehat{h}_{n+1}^{(j)} - \mathbb{E}_{n+1}^{\mathsf{J}} \widehat{h}_{n+1}). \tag{2.78e}$$

By making an empirical approximation of the formula (2.58) the matrix $\widetilde{K}_n$ is defined using the identification

$$\widetilde{K}_n = \widehat{V}_{n+1} \widehat{H}_{n+1}^{\top} \left[(\widehat{H}_{n+1} \widehat{H}_{n+1}^{\top} + \Gamma) + (\widehat{H}_{n+1} \widehat{H}_{n+1}^{\top} + \Gamma)^{1/2} \Gamma^{1/2} \right]^{-1}.$$

Remark 2.2.18. *The key difference between* (2.75) *and* (2.78) *is that the former involves inversion in state space, and the latter in data space. The relative size of the two dimensions dictates which is preferable.* ■

Ensemble Transform Kalman Filters The two EAKFs just defined both involve application, and inversion, of matrices which are applied on the left and act on state space. A different class of algorithms, known as *ensemble transform Kalman filters* (ETKF), involve matrix multiplication from the right and consequently inversions take place in the ensemble space of dimension J. In many applications this is far smaller than the dimension of the state or data spaces, and then use of this version of the methodology is preferred. The aim is to determine matrix $Z_n \in \mathbb{R}^{J \times J}$, and to derive Kalman gain K_n from Z_n, so that the following interacting particle system produces an ensemble of particles $\{v_{n+1}^{(j)}\}_{j \in \mathsf{J}}$ with empirical covariance C_{n+1} defined by the second item in display (2.76):

$$\widehat{v}_{n+1}^{(j)} = \Psi(v_n^{(j)}) + \xi_n^{(j)},\ n \in \mathbb{Z}^+, \tag{2.79a}$$
$$\widehat{m}_{n+1} = \mathbb{E}_{n+1}^{\mathsf{J}} \widehat{v}_{n+1}, \tag{2.79b}$$
$$m_{n+1} = \widehat{m}_{n+1} + K_n(y_{n+1}^{\dagger} - \mathbb{E}_{n+1}^{\mathsf{J}} \widehat{h}_{n+1}), \tag{2.79c}$$
$$v_{n+1}^{(j)} = m_{n+1} + \sum_{i=1}^{J} \left(\widehat{v}_{n+1}^{(i)} - \widehat{m}_{n+1} \right) (Z_n)_{ij}. \tag{2.79d}$$

To this end we define the matrix of ensemble deviations

$$V_n = \frac{1}{\sqrt{J}} \left(v_n^{(1)} - \mathbb{E}_{*,n}^{\mathsf{J}} v_n, v_n^{(2)} - \mathbb{E}_{*,n}^{\mathsf{J}} v_n, \cdots, v_n^{(J)} - \mathbb{E}_{*,n}^{\mathsf{J}} v_n \right) \in \mathbb{R}^{d_v \times J},$$

where, here, expectation $\mathbb{E}^{\mathsf{J}}_{*,n}$ is with respect to the empirical measure (2.68) and v_n is a random variable with this distribution. It then follows from (2.79d) that

$$V_{n+1} = \widehat{V}_{n+1} Z_n. \tag{2.80}$$

If we define

$$Z_n = \left(I + \widehat{H}_{n+1}^\top \Gamma^{-1} \widehat{H}_{n+1} \right)^{-1/2} \in \mathbb{R}^{J \times J}, \tag{2.81}$$

then, as desired, $V_{n+1} V_{n+1}^\top = C_{n+1}$ as defined by (2.76), by virtue of (2.77). The calculations in (2.77) can also be utilized to verify that the empirical Kalman gain matrix defined by (2.74) satisfies

$$K_n = \widehat{V}_{n+1} Z_n^2 \widehat{H}_{n+1}^\top \Gamma^{-1}.$$

Using this formula for K_n in (2.75) leads to an algorithm which matches first and second order statistics of the Gaussian projected filter, at $n+1$, requiring only matrix inversions in space of dimension defined by the number of particles J.

We finally note that (2.79c) and (2.79d) can be combined into a single transformation step of the form

$$v_{n+1}^{(j)} = \sum_{i=1}^{J} \widehat{v}_{n+1}^{(i)} (S_n)_{ij}\,, \tag{2.82}$$

where $S_n \in \mathbb{R}^{J \times J}$ replaces the matrix Z_n in (2.79d) such that

$$m_{n+1} = \sum_{j=1}^{J} v_{n+1}^{(j)} = \sum_{i,j=1}^{J} \widehat{v}_{n+1}^{(i)} (S_n)_{ij}$$

holds in addition to (2.80) with Z_n replaced by S_n.

Remark 2.2.19. *Formulation (2.82) has a number of attractive features. First, it clearly reveals that the analysis $\{v_{n+1}^{(j)}\}_{j \in \mathsf{J}}$ lies in the span of the space spanned by the predictions $\{\widehat{v}_{n+1}^{(j)}\}_{j \in \mathsf{J}}$, which is relevant whenever $J < d_v$. Second, all particle implementations of the ensemble Kalman filter and many of its extensions can be put into the framework (2.82) with the (possibly random) matrix S_n chosen appropriately. Third, it encodes a coupling between the prediction $\{\widehat{v}_{n+1}^{(j)}\}_{j \in \mathsf{J}}$ and the analysis $\{v_{n+1}^{(j)}\}_{j \in \mathsf{J}}$ at the level of their associated empirical measures $\widehat{\mu}_n^{\mathsf{J}}$ and μ_n^{J}, respectively. See the following bibliographic Subsection 2.2.7 for more details.* ■

2.2.7 Bibliographical Notes

Ideas from feedback control underlie the material in Subsection 2.2.2, addressing Objective 1. Control theory is an enormous subject in its own right and we cannot

do justice to it in this chapter. For study of linear control theory, as illustrated in Example 2.2.2, see Åström and Murray (2021) and Sontag (2013) for engineering and mathematical treatments respectively. For the control theoretic approach to the state estimation problem see D. G. Luenberger (1964) and D. Luenberger (1971).

Our study of control-theoretic methods has focused on 3DVAR. Recent analysis of the 3DVAR method rests heavily on ideas arising from determining modes for dissipative evolution equations, an idea with roots in the paper ClPRIAN Foias and Prodi (1967) and unified in the book Temam (2012). The use of these ideas in data assimilation was introduced in Olson and Titi (2003) and developed further in Hayden, Olson, and Titi (2011) and Ciprian Foias, Mondaini, and Titi (2016); the papers K. J. Law and Andrew M Stuart (2012), K. J. Law, Shukla, and Andrew M Stuart (2014), KJH Law et al. (2016) and Sanz-Alonso and Andrew M Stuart (2015) essentially establish the stability of these deterministic results to small noise perturbations; see also Moodey et al., 2013 for related analysis. In the context of using observations to control the instability of chaotic systems, all of this work may be seen as building on the study of synchronization (Pecora and Carroll, 1990), reviewed in Ashwin (2003).

In Subsection 2.2.3 we introduce the probabilistic approach to filtering, addressing Objective 2. In low dimensional systems particle filters provide a flexible and efficient tool for attacking probabilistic filtering; see Doucet, De Freitas, and Gordon, 2001. However in this chapter our focus is on high dimensional problems and Kalman-based methods specifically. The books Reich and Colin Cotter (2015), Asch, Marc Bocquet, and Nodet (2016), Kody Law, A. Stuart, and Kostas Zygalakis (2015), Harlim and Majda (2010), Abarbanel (2013), and G. Evensen, F.C. Vossepoel, and van Leeuwen (2022) provide overviews of a variety of filtering methods, and ensemble Kalman methods from Subsection 2.2.6 in particular.

The Kalman filter (Kalman, 1960) from Example 2.2.6 led to arguably the first systematic analysis of an algorithm for incorporation of discrete time data into estimation of a discrete time stochastic dynamical system; it applies only to linear Gaussian systems. The monograph Andrew H Jazwinski (2007) provides an introduction to nonlinear filtering both in discrete and continuous time; in particular it discusses the extended Kalman filter, found by applying the Kalman filter to a linearization of the state and data dynamics. However this method does not work well in high dimensions Michael Ghil et al., 1981, motivating the use of mean field maps, as introduced in Subsection 2.2.5, and the ensemble-based methods from

Subsection 2.2.6 which approximate them. The approximation of mean field maps by interacting particle systems is overviewed in Sznitman (1991).

We refer the reader to the excellent monographs by Asch, Marc Bocquet, and Nodet (2016) and G. Evensen, F.C. Vossepoel, and van Leeuwen (2022) for texts with emphasis on important implementation details, not covered in this chapter, relating to these ensemble Kalman methods; these include techniques such as inflation and localization that are central to the success of these methods in high dimensions. We also refer to the paper by Vetra-Carvalho et al. (2018) for a comprehensive review of the algorithmic details of ensemble Kalman methods. Here we point to two particular implementation details that are of particular practical importance. The first concerns the use of ensemble square-root filters from Subsection 2.2.6.3. The matrix Z_n in (2.80) is not uniquely defined by the requirement $V_{n+1}V_{n+1}^\top = C_{n+1}$. Formula (2.81) constitutes one possible choice, which leads to a symmetric Z_n; see Livings, Dance, and Nichols (2008) for more details. Second, finite particle implementations of the stochastic EnKF from Subsection 2.2.6.2 entail that the random realizations $\widehat{y}_{n+1}^{(j)}$ appear both in the Kalman gain K_n as well as in the innovation term in (2.70c). As first observed by Houtekamer and Mitchell (2005), this leads to a systematic underestimation of the ensemble spread, which vanishes in the $J \to \infty$ limit, but can affect the performance of the EnKF for small particle sizes.

Particle-based extensions of the classical Kalman filter to nonlinear filtering problems include the unscented Kalman filter and the ensemble Kalman filter. While this chapter focuses primarily on ensemble Kalman filter techniques, the unscented Kalman filter is an approach based on application of quadrature to the Gaussian projected filter from Subsection 2.2.4; see Julier, Uhlmann, and Durrant-Whyte (2000) as well as Särkkä and Svensson (2023). A discussion and evaluation in the context of ensemble square root filters may be found in X. Wang, C.H. Bishop, and S.J. Julier (2004).

Much of the development of ensemble Kalman methods reflects the historical roots of the subject in the geophysical sciences, the atmosphere-ocean sciences in particular, including Lagrangian data assimilation, and in the modeling of subsurface flow (G. Burgers, van Leeuwen, and G. Evensen, 1998; P.L. Houtekamer and M.L. Mitchell, 1998; Jeffrey L Anderson, 2001; C.H. Bishop, B.J. Etherton, and S.J. Majumdar, 2001; J.S. Whitaker and T.M. Hamill, 2002; M.K. Tippett et al., 2003; B.R. Hunt, E.J. Kostelich, and I. Szunyogh, 2007; G. Li and Albert Coburn

Reynolds, 2009; P. Sakov, D. S. Oliver, and L. Bertino, 2012; M. Bocquet and P. Sakov, 2014; Evensen, 2019; Marc Bocquet and Pavel Sakov, 2012; Marc Bocquet, Gurumoorthy, et al., 2017; Gurumoorthy et al., 2017; Sampson et al., 2021; Kuznetsov, Ide, and Jones, 2003; Salman et al., 2006). We also mention the randomized maximum likelihood (RML) approach to Bayesian inference which is closely related to the analysis step of a stochastic EnKF and has also been developed primarily through application in the geophysical sciences (Kitanidis, 1995; Dean S. Oliver, Cunha, and Albert C. Reynolds, 1997; D. Oliver, A. Reynolds, and N. Liu, 2008).

There is also a body of literature concerning the analysis and development of ensemble methods with an emphasis on applications in complex and turbulent flows: Grooms, Y. Lee, and Majda (2014), Grooms, Y. Lee, and Majda (2015), Robinson, Grooms, and Kleiber (2018), Y. Lee, Majda, and Qi (2017), Gottwald and Majda (2013), D. Kelly, Majda, and Tong (2015), Tong, Majda, and D. Kelly (2016a), Tong, Majda, and D. Kelly (2016b), D. Kelly, Majda, and Tong (2015), Harlim, Mahdi, and Majda (2014), Majda and Tong (2018), Fertig, Harlim, and Brian R Hunt (2007), Harlim and Brian R Hunt (2007a), Harlim and Brian R Hunt (2007b). The conceptual fluid dynamics models of Lorenz (Edward N Lorenz, 1996a) (often referred to, collectively, as Lorenz '96 models) have been particularly influential in germinating this body of work and we use them exclusively in our illustrative Examples 2.2.3, 2.2.5 and 2.2.16. Furthermore we will make use of the relationship between the multiscale and singlescale version of the model as developed in Fatkullin and Vanden-Eijnden (2004).

Recently ensemble Kalman methods have been developed for potential use in machine learning (Haber, Lucka, and Ruthotto, 2018; Nikola B Kovachki and Andrew M Stuart, 2019; Guth, Schillings, and Weissmann, 2022; Grooms, 2021; Gottwald and Reich, 2021; L. M. Yang and Grooms, 2021; J. Pidstrigach and S. Reich, 2023); see also Marc Bocquet, Gurumoorthy, et al. (2017), Yuming Chen, Sanz-Alonso, and Willett (2022) for research at the intersection of machine learning with ensemble Kalman methodology.

In this survey we have started from mean field equations, in Subsection 2.2.5, and then discretized the mean field limit using J particles in subsequent subsections. It is of interest to demonstrate that the discrete formulations which arise actually converge to the mean field equations in the $J \to \infty$ limit. This has indeed been established for the ensemble Kalman filter when applied in the linear Gaussian setting in which the mean field limit exactly recovers the filtering distribution (Le Gland, Monbet,

and Tran, 2011a; J. Mandel, L. Cobb, and J. Beezley, 2011; Kwiatkowski and J. Mandel, 2015) and indeed some results also apply in the nonlinear setting. In the continuous time-setting long-time error estimates, exploiting ergodicity of the Kalman-Bucy filter itself and propagation of chaos ideas to extend to the ensemble Kalman approximations, are developed in Del Moral and Tugaut (2018). The paper K. J. Law, Tembine, and Tempone (2016) studies related work concerning the mean field limit of ensemble Kalman methods in the context of non-Gaussian problems. The papers Ding, Q. Li, and Lu (2021), Ding and Q. Li (2021a) and Ding and Q. Li (2021b) study particle approximation of mean field limits beyond the Gaussian setting, primarily in the context of the solution of inverse problems. The papers Hoel, K. J. Law, and Tempone (2016) and Chernov et al. (2021) study the use of multilevel approximation of the mean field limit, coupling ensemble approximations at different levels of space or time discretization.

In Subsection 2.2.5.3 we introduce the idea of second-order transport: approximations of the perfect transport maps that effect filtering. The non-uniqueness of second order transport maps is studied in continuous time, for linear Gaussian stochastic differential equations, in Taghvaei and Prashant G Mehta (2020): this work is closely related to our analysis in the first two subsections of Section 2.B. Non-Gaussian extensions are discussed in Taghvaei and Hosseini (2022). We also highlight that it is possible to construct second order transport maps $\widetilde{T}$, which satisfy conditions different from (2.50b) and (2.56b), respectively. For example, one could request that

$$\mathsf{G}\big((\widetilde{T}_n^S)_\sharp \pi\big) = \mathsf{GB}_n(\pi).$$

This approximation has been utilized by J. Lei and P. Bickel (2011). The analogous deterministic approach has been put forward by J. Tödter and B. Ahrens (2015) and has been explored further, for example, in W. Acevedo, J. d. Wiljes, and S. Reich (2017). Alternatively, one can also replace the definition (2.36) of the Gaussian projection operator G. To this end, recall that the Kullback–Leibler divergence between probability measures π_1 and π_2 on $\mathbb{R}^d$ is defined as

$$d_{\mathrm{KL}}(\pi_1||\pi_2) = \int \pi_1(du) \log \frac{\mathrm{d}\pi_1}{\mathrm{d}\pi_2}(u);$$

in particular, it is not symmetric in its two arguments. Gaussian variational inference (C. Bishop, 2011) is for example based of the definition

$$\mathsf{G}\mu = \mathrm{a}rgmin_{\pi\in\mathcal{G}} d_{\mathrm{KL}}(\pi||\mu) \tag{2.83}$$

in place of (2.36); note, however, that the minimizer of (2.83) may not be unique, whilst the minimizer of (2.36) is always unique.

While we follow the moment matching perspective on the derivation of ensemble Kalman filter methods in this survey, we mention in passing that there is an alternative perspective based on linear minimum variance estimators. See Van Leeuwen (2020b) in the context of the stochastic ensemble Kalman filter and the Subsection 2.B.3 as well as J. Lei and P. Bickel (2011) for nonlinear extensions. The Bayes linear methodology is also an alternative approach of potential interest Goldstein and Wooff, 2007.

Even in the mean field limit $J \to \infty$, the ensemble Kalman filter provides approximations only to approximate transport based filters. They are only exact in the linear Gaussian setting (Le Gland, Monbet, and Tran, 2011a); recent works develop new tools of analysis to extend this to nonlinear filtering problems that are close to Gaussian (J. A. Carrillo et al., 2024; Calvello, Monmarché, et al., 2026). As mentioned in Subsection 2.1.1, sequential Monte Carlo methods can be designed to be consistent with the underlying nonlinear filtering problem as defined, for example, by perfect transport based filters. Foundational analysis of these particle methods is undertaken in Crisan, Pierre Del Moral, and Lyons (1999) and Del Moral (2004); but we reiterate that, in contrast to ensemble Kalman based methods, they do not scale well to high dimensions. We also point to Pierre Del Moral, Doucet, and Jasra (2006) for an application of the sequential Monte Carlo method to Bayesian inference problems; the approach therein is closely related to iterative implementations of the EnKF that were subsequently developed in the papers G. Li and Albert Coburn Reynolds (2009), Gu and Dean S Oliver (2007), P. Sakov, D. S. Oliver, and L. Bertino (2012).

Despite only providing approximations to the exact filtering distribution, i.e. from the perspective of Objective 2, accuracy and stability results for the ensemble Kalman filter, viewed as a state estimator and hence from the perspective of Objective 1, have been derived. See, for example, González-Tokman and Brian R Hunt (2013), D. T. Kelly, K. J. Law, and Andrew M Stuart (2014), Tong, Majda, and D. Kelly (2016b), Tong, Majda, and D. Kelly (2016a) and Pierre Del Moral and Horton (2023a). Mechanisms for finite time filter divergence have also been identified (Gottwald and Majda, 2013; D. Kelly, Majda, and Tong, 2015).

Extending the ensemble Kalman filter to strongly nonlinear and high dimensional state estimation problems constitute an area of active ongoing research. The current

state of the art has been summarized in Van Leeuwen, Hans R. Künsch, et al. (2019b) in the context of high dimensional geophysical applications. Extensions of the transport framework (2.46), which build on approximating the perfect transport maps T^D in (2.46b) in an asymptotically consistent manner, include the work by S. Reich (2013), Y. Cheng and S. Reich (2015), Spantini, Baptista, and Youssef Marzouk (2022a), and Zech and Y. Marzouk (2022). In an alternative line of research there have been several proposals to construct hybrid methods, which aim to adaptively bridge between the ensemble Kalman and particle filters, including the work by A.S. Stordal et al. (2011), Frei and Hans R Künsch (2013), N. Chustagulprom, S. Reich, and M. Reinhardt (2016), and Nerger (2022).

In this context, the transformation formula (2.82) proves to be rather useful since most existing particle based methods can be covered by appropriate choices of S_n, where S_n is typically the realization of a random matrix. See Reich and Colin Cotter (2015) for more details. For example, a resampling step in a sequential Monte Carlo method gives rise to a matrix S_n with a single non-zero entry equal to one in each of its columns. More generally it holds that, for all $j \in \mathsf{J}$,

$$\sum_{i\in\mathsf{J}} (S_n)_{ij} = 1.$$

In particular, the matrix S_n can be chosen to correspond to an optimal coupling between two discrete random variables (S. Reich, 2013), which builds a link between filtering and optimal transport also explored in Corenflos et al. (2021). The subject of optimal transport is given a comprehensive treatment in C. Villani (2008); see also Cédric Villani (2021). Computational aspects of the subject including entropy-regularized optimal transport are discussed in Cuturi (2013) and Peyré and Cuturi (2019). Entropy-regularization is linked to the Schrödinger bridge problem, and connections with data assimilation are developed in Reich (2019). See also Subsection 2.1.1 for discussion of transport-based methodologies within the context of ensemble Kalman methods. See Example 2.B.7 and Reich and Colin Cotter (2015) for the connection between the map (2.66) and optimal transport. For a derivation of the standard formulae for mean and covariance of conditioned Gaussians used in the proof of Lemma 2.2.9 see Eaton (2007).

Finally we observe that we do not discuss, in this chapter, the *smoothing* approach to state estimation from data in model (2.1). This approach aims at finding the entire sequence $\{v_\ell\}_{\ell=0}^n$ from the data $Y_n^\dagger$. Thus state estimates depend on data in their future. To read about smoothing see G. Evensen, F.C. Vossepoel, and van Leeuwen

(2022) and Sanz-Alonso, A. Stuart, and Taeb (2023b). We note here that there is a smoothing counterpart of the 3DVAR algorithm (see Remark 2.2.1) known as 4DVAR because, for physical systems, it uses data distributed in the three space and one time dimensions.

2.A Lorenz '96 Multiscale

To illustrate the problems of both state estimation and parameter estimation we use, throughout this chapter, variants on the Lorenz '96 model. In particular we use both the Lorenz '96 multiscale system, introduced in subsection 2.A.1, and a singlescale closure derived from it in the scale-separated case, described in Subsections 2.A.2 and 2.2.7. If we (i) generate data with the singlescale model, and assimilate using the same model, we are able to test algorithms in their basic (perfect model) form; on the other hand if we (ii) generate data with the multiscale model, and assimilate using the singlescale model, this allows us to study the effect of model misspecification on data assimilation. Subsection 2.A.3 contains an example of the type (ii), whilst Examples 2.2.3, 2.2.5 and 2.2.16 are of type (i).

2.A.1 Lorenz '96 Multiscale Model

Let $v \in C(\mathbb{R}^+, \mathbb{R}^L)$ and $w \in C(\mathbb{R}^+, \mathbb{R}^{L\times J})$. We write down an ODE for (v, w) in which each variable $v_\ell \in \mathbb{R}$ is coupled to a subgroup of fast variables $w_\ell = \{w_{\ell,j}\}_{j=1}^J \in \mathbb{R}^J$; further (discrete) boundary condition couplings impose periodicity in L and link $\{w_{\ell,j}\}_{j=1}^J$ to $\{w_{\ell+1,j}\}_{j=1}^J$. The particular form of the system of ODEs is as given in Fatkullin and Vanden-Eijnden (2004). For $\ell = 1 \dots L$ and $j = 1 \dots J$, the ODEs are

$$\dot{v}_\ell = f_\ell(v) + h_v \overline{w}_\ell, \quad \overline{w}_\ell = \frac{1}{J}\sum_{j=1}^J w_{\ell,j}, \tag{2.84a}$$

$$\dot{w}_{\ell,j} = \frac{1}{\varepsilon} r_j(v_\ell, w_\ell), \tag{2.84b}$$

where

$$f_\ell(v) := -v_{\ell-1}(v_{\ell-2} - v_{\ell+1}) - v_\ell + F, \tag{2.85a}$$

$$r_j(v_\ell, w_\ell) := -w_{\ell,j+1}(w_{\ell,j+2} - w_{\ell,j-1}) - w_{\ell,j} + h_w v_\ell, \tag{2.85b}$$

and we impose the boundary conditions

$$v_{\ell+L} = v_\ell, \quad w_{\ell+L,j} = w_{\ell,j}, \quad w_{\ell,j+J} = w_{\ell+1,j}. \tag{2.86}$$

Here $\varepsilon > 0$ is a scale-separation parameter, $h_v, h_w \in \mathbb{R}$ govern the couplings between the fast and slow systems, and $F > 0$ provides a constant forcing.

2.A.2 Lorenz '96 Singlescale Model

Let $v = \{v_\ell\}_{\ell=1}^L, w = \{w_\ell\}_{\ell=1}^L$ and $\overline{w} = \{\overline{w}_\ell\}_{\ell=1}^L$. Then we may write (2.84) in the form

$$\dot{v} = F(v) + h_v \overline{w}, \tag{2.87a}$$

$$\dot{w} = \frac{1}{\varepsilon} R(v, w), \tag{2.87b}$$

for suitable definitions of F, R. If $\varepsilon \ll 1$ then the dynamics for the w governed (2.87b) evolve on a much faster timescale than the dynamics for the v governed by (2.87a). Thus it is a reasonable approximation to think of v as frozen in (2.84b). If we assume that the dynamics of w with v frozen are ergodic with invariant measure $\mu^v(dw)$ (a measure in w, parameterized by v) then the averaging principle (Vanden-Eijnden et al., 2003; Abdulle et al., 2012; Pavliotis and A. Stuart, 2008) suggests that we may make the approximation

$$\overline{w} \approx M(v) := \int \overline{w}\, \mu^v(\mathrm{d}w).$$

If we also invoke the approximation $M_\ell(v) \approx m(v_\ell)$, which is shown to be valid for large J in Fatkullin and Vanden-Eijnden (2004), then we arrive at the singlescale Lorenz '96 model (2.9). This program of analysis for the Lorenz '96 model, and studies of the validity of the resulting singlescale approximation, was established in the paper Fatkullin and Vanden-Eijnden (2004). The function $m(\cdot)$ is not given explicitly, but may be fit to data in various different ways, as explained in Fatkullin and Vanden-Eijnden (2004). Figure 2.1 shows such an m, fit using Gaussian process regression methodology as detailed in Subsection 4.3 of Burov et al. (2021).

2.A.3 Example

The following example relates to the discussion in Remark 2.2.4.

Example 2.A.1. *Throughout this example we set* $L = 9$, $J = 8$, $h_v = -0.8$, $h_w = 1$, $F = 10$ *and* $\varepsilon = 2^{-7}$. *Note that the equation (2.84a) has the form of the singlescale Lorenz model* (2.9) *for* v *with a coupling to the fast variables* w *governed by (2.84b) replacing the function* $m(\cdot)$. *In Figure 2.1 we display the dynamics of a slow variable and one associated fast variable of the Lorenz '96 multiscale model* (2.84)*; at the parameter values chosen the system is chaotic.*

We consider observations $\{y_n^\dagger\}_{n\in\mathbb{Z}^+}$ *arising from the model*

$$(v_{n+1}^\dagger, w_{n+1}^\dagger) = \Psi_{mult}(v_n^\dagger, w_n^\dagger),$$
$$y_{n+1}^\dagger = h(v_{n+1}^\dagger),$$

where Ψ_{mult} is the solution operator to the multiscale model (2.84)–(2.86) *over the observation time interval τ. We then take the data from the multiscale model and assimilate it into the singlescale model* (2.9)*, recalling the specific function m shown in Figure 2.1, and discussion in the preceding subsection concerning elimination of the fast variables in favour of a simple closure, using the averaging principle.*

We now discuss data assimilation using this multiscale data. As in Example 2.2.3, we assume that the observation function is linear: $h(v) = Hv$ for matrix $H : \mathbb{R}^9 \to \mathbb{R}^6$ defined by (2.10)*. As before we choose the gain $K : \mathbb{R}^6 \to \mathbb{R}^9$ to be defined by* (2.12) *and employ the 3DVAR algorithm* (2.11) *with Ψ the solution operator over time-interval τ for the singlescale model. Thus model misspecification is present, because data $Y^\dagger$ comes from the multiscale model.*

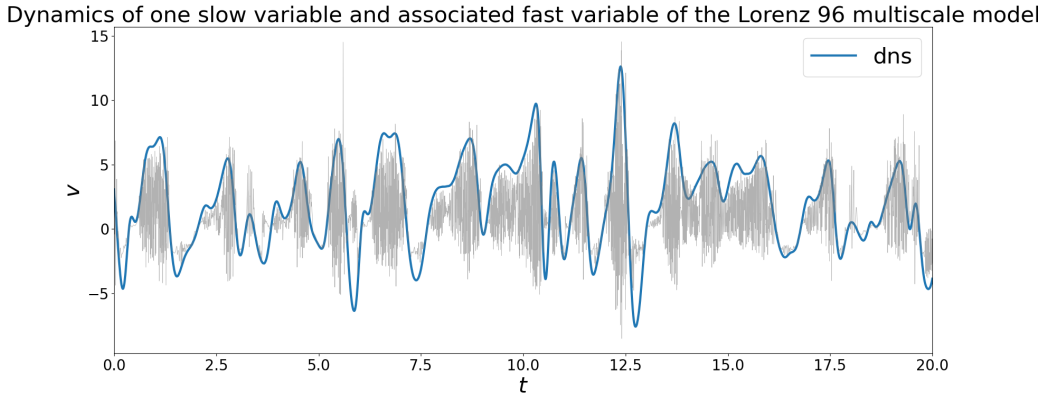
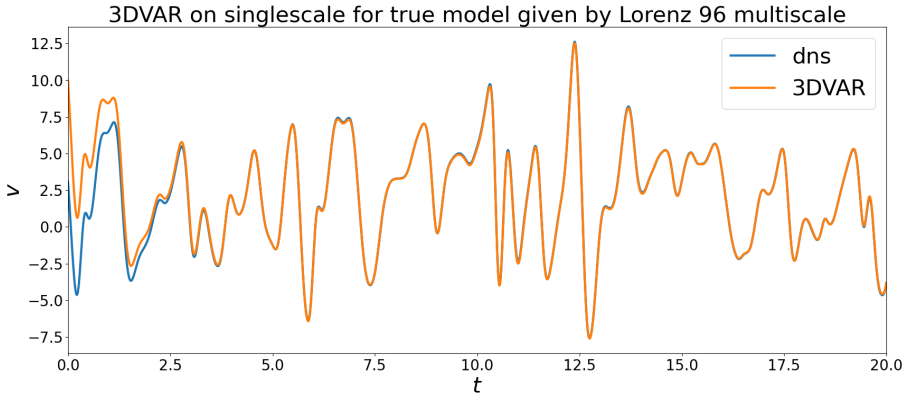


Figure 2.1: Dynamics of a slow variable and an associated fast variable. Here "dns", in blue, labels the slow variable computed by direct numerical simulation; in grey is the related fast variable.

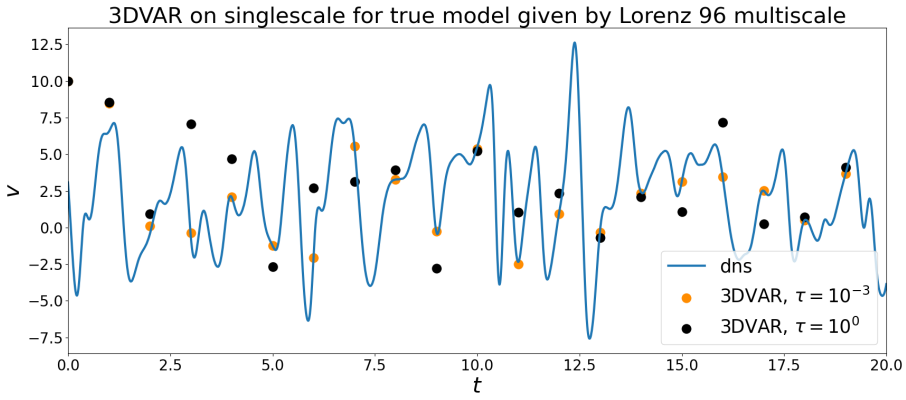
As in Example 2.2.3 we display the results of 3DVAR on the unobserved component v_3. Figure 2.2a shows the behavior of the algorithm for $\tau = 10^{-3}$. Despite the fact that the data is generated from the multiscale model, whilst assimilation is conducted using the singlescale model, 3DVAR produces an accurate estimate of the true state with, after synchronization, the only discernible errors being slight under and overshoots. Figure 2.2b shows the effect of varying τ, the time between observations. As in Example 2.2.3 the assimilation is significantly worse when τ is larger. ■

2.B Mean Field Maps

In Subsection 2.B.1 we discuss the existence, form and properties of mean field maps which carry out the program of approximate transport described in Subsection



(a)



(b)

Figure 2.2: In (a) the estimates of v_3 in time produced by 3DVAR using $\tau = 10^{-3}$ are displayed against the true dynamics. In (b) the estimates at each unit time obtained using 3DVAR with assimilation at $\tau = 10^0$ and $\tau = 10^{-3}$ are shown. Again the acronym "dns" refers to direct numerical simulation. 3DVAR successfully synchronizes with the direct numerical simulation at the smaller value of τ but fails to do so as well when τ is larger. It is noteworthy that the synchronization takes place here in the context of model misspecification: the data is generated by the multiscale model, but 3DVAR is applied using the singlescale model.

2.2.5.4; these maps require access to simulated data and are hence referred as stochastic second order transport maps. In Subsection 2.B.2 we discuss the existence, form and properties of mean field maps which carry out the program of approximate transport described in Subsection 2.2.5.5; these maps do not require access to simulated noisy data and are hence referred to as deterministic second order transport maps. The two approaches provide fundamental underpinnings of ensemble Kalman methods as we develop them in this chapter, but were not adopted in its historical development. Subsection 2.B.3 is devoted to the minimum variance approximation,

a way of deriving the Kalman transport map (2.54), which is part of the historical development of the subject, but which does not play a central role in our presentation and analysis of the subject.

2.B.1 Mean Field Maps – Simulated Data

We note that the maps of interest in this subsection take $Law\,(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ into $Law\,(v_{n+1})$. Since the analysis is independent of any specific value of discrete time n, we drop the subscript $n+1$ throughout the analysis; this streamlines the notation. Similar considerations apply in Subsection 2.B.2 and in Subsection 2.B.3.

Let $(\widehat{v}, \widehat{y}) \sim \pi$, where

$$\mathsf{G}\pi = \mathsf{N}\left(\begin{bmatrix}\widehat{m}\\ \widehat{o}\end{bmatrix}, \begin{bmatrix}\widehat{C} & \widehat{C}^{vy}\\ (\widehat{C}^{vy})^\top & \widehat{C}^{yy},\end{bmatrix}\right).$$

Define

$$m = \widehat{m} + \widehat{C}^{vy}(\widehat{C}^{yy})^{-1}\big(y^\dagger - \widehat{o}\big), \tag{2.88a}$$

$$C = \widehat{C} - \widehat{C}^{vy}(\widehat{C}^{yy})^{-1}(\widehat{C}^{vy})^\top. \tag{2.88b}$$

Note that these are the mean and covariance of the Gaussian random variable $\widehat{v}$ conditioned on $\widehat{y} = y^\dagger$ on the assumption that $(\widehat{v}, \widehat{y})$ is distributed according to the Gaussian measure $\mathsf{G}\pi$; see (2.40). The quantities in the above identities are as defined in Subsection 2.2.3.4, with the caveat that the subscripts $n+1$ have been dropped for clarity of exposition.

Our goal is to identify maps of the form

$$v = A\widehat{v} + B\widehat{y} + a \tag{2.89}$$

so that, if $(\widehat{v}, \widehat{y}) \sim \pi$, then v has mean m and covariance C given by (2.88). We make the following assumptions on the covariance under $\mathsf{G}\pi$ and on the matrices A, B and vector a.

Assumptions 2.B.1. *The covariance under $\mathsf{G}\pi$ is invertible. The matrices A, B and vector a may depend on $y^\dagger$ and measure π but not on the random variables $(\widehat{v}, \widehat{y})$; thus* (2.89) *takes the explicit form*

$$v = A(\pi, y^\dagger)\widehat{v} + B(\pi, y^\dagger)\widehat{y} + a(\pi, y^\dagger). \qquad \blacksquare$$

Recall the discussion of pushforward of measures in the introduction to Subsection 2.2.5. Under Assumptions 2.B.1 pushforward under the map (2.89), when chosen

to match the desired first and second order statistics, defines a nonlinear map on the space of measures, and in particular on π itself. In what follows all expectations are computed under π. Since the covariance under $\mathsf{G}\pi$ is strictly positive-definite so are the marginal covariances $\widehat{C}$ and $\widehat{C}^{yy}$ (see Andrew M Stuart (2010a)[Lemma 6.21]). Thus we may define the conditional mean and covariance by (2.88), as well as Kalman gain K, and conditional covariance $\widetilde{C}$, given by

$$K = \widehat{C}^{vy}(\widehat{C}^{yy})^{-1}, \tag{2.90}$$

$$\widetilde{C} = \widehat{C}^{yy} - (\widehat{C}^{vy})^{\top}(\widehat{C})^{-1}\widehat{C}^{vy}; \tag{2.91}$$

we note that the conditional covariances C and $\widetilde{C}$ are also strictly positive-definite (see Andrew M Stuart (2010a)[Lemma 6.21]). From equations (2.88a) and (2.89) the following is immediate.

Lemma 2.B.2. *Let Assumptions 2.B.1 hold and let $(\widehat{v}, \widehat{y}) \sim \pi$. If v given by* (2.89) *has mean given by (2.88a) then*

$$a = (I - A)\mathbb{E}\widehat{v} + Ky^{\dagger} - (B + K)\mathbb{E}\widehat{y}.$$

$\diamond$

As a consequence it follows that

$$v = \mathbb{E}\widehat{v} + A(\widehat{v} - \mathbb{E}\widehat{v}) + B(\widehat{y} - \mathbb{E}\widehat{y}) + K(y^{\dagger} - \mathbb{E}\widehat{y}), \tag{2.92}$$

and that

$$\mathbb{E}\Big((v - \mathbb{E}v) \otimes (v - \mathbb{E}v)\Big) = A\widehat{C}A^{\top} + B\widehat{C}^{yy}B^{\top} + A\widehat{C}^{vy}B^{\top} + B(\widehat{C}^{vy})^{\top}A^{\top}.$$

Thus, to match the covariance of the conditioned random variable, we obtain

$$C = A\widehat{C}A^{\top} + B\widehat{C}^{yy}B^{\top} + A\widehat{C}^{vy}B^{\top} + B(\widehat{C}^{vy})^{\top}A^{\top}. \tag{2.93}$$

Theorem 2.B.3. *Let Assumptions 2.B.1 hold and let $(\widehat{v}, \widehat{y}) \sim \pi$. If a is given by Lemma 2.B.2 then v defined by* (2.89) *has covariance (2.88b) if and only if real-valued matrices A and B are related by the identity*

$$F\widehat{C}^{-1}F^{\top} = C - B\widetilde{C}B^{\top}, \tag{2.94}$$

where

$$F = A\widehat{C} + B(\widehat{C}^{vy})^{\top}. \tag{2.95}$$

$\diamond$

Proof. We complete the square on the right hand side of (2.93) to obtain

$$(A\widehat{C} + B(\widehat{C}^{vy})^\top)\widehat{C}^{-1}(A\widehat{C} + B(\widehat{C}^{vy})^\top)^\top = C',$$

where

$$C' = C - B\widetilde{C}B^\top.$$

Rearranging gives the desired result. □

Define

$$\mathcal{B} = \{B \in \mathbb{R}^{d_v \times d_y} : C' \succ 0\},$$

noting that this set is non-empty, and contains an open (and hence uncountable) set of B, since $C \succ 0$. For $B \in \mathcal{B}$ consider the eigenvalue problem

$$C'\varphi^{(i)} = (s^{(i)})^2\varphi^{(i)}, \quad \langle\varphi^{(i)}, \varphi^{(j)}\rangle = \delta_{ij}.$$

Note that this has d_v real solutions, up to sign changes in the eigenvectors and assuming the $s^{(i)}$ to be non-negative. We now seek to express F in terms of $B \in \mathcal{B}$. Writing the SVD $F\widehat{C}^{-\frac{1}{2}} = U\Xi V^\top$, where $U, V \in \mathbb{R}^{d_v \times d_v}$ are orthogonal matrices and $\Xi \in \mathbb{R}^{d_v \times d_v}$ is a diagonal matrix, we see from (2.94) that

$$U\Xi^2 U^\top = C'$$

so that U has columns given by the $\{\varphi^{(i)}\}_{i=1}^{d_v}$ and corresponding diagonal entries of Ξ, $\pm s^{(i)}$. We define

$$U = \left(\varphi^{(1)}, \cdots, \varphi^{(d_v)}\right), \quad \Xi = \operatorname{diag}\left(\pm s^{(1)}, \cdots, \pm s^{(d_v)}\right). \tag{2.96}$$

Corollary 2.B.4. *For every $B \in \mathcal{B}$ the choices of A such that the pair (A, B) satisfies the criterion of Theorem 2.B.3 are defined as follows. For U, Ξ as given in* (2.96)*, set*

$$F = U\Xi V^\top \widehat{C}^{\frac{1}{2}},$$

where V is an arbitrary orthogonal matrix in $\mathbb{R}^{d_v \times d_v}$. Then

$$A = \left(F - B(\widehat{C}^{vy})^\top\right)\widehat{C}^{-1}.$$

◇

Example 2.B.5. *Among the uncountably many possible solutions for pairs (A, B) we highlight two. The choice $B = 0$ is interesting because it does not require the*

data variable $\widehat{y}$ in the definition of v. The choice $A = I$ is interesting because it does not require action of an operator acting on $\widehat{v}$.

The first, with $B = 0$, allows the choice $F = C^{\frac{1}{2}}\widehat{C}^{\frac{1}{2}}$ and then $A = C^{\frac{1}{2}}\widehat{C}^{-\frac{1}{2}}$. Thus the map (2.92) *becomes*

$$v = \mathbb{E}\widehat{v} + C^{\frac{1}{2}}\widehat{C}^{-\frac{1}{2}}(\widehat{v} - \mathbb{E}\widehat{v}) + \widehat{C}^{vy}(\widehat{C}^{yy})^{-1}(y^\dagger - \mathbb{E}\widehat{y}) \tag{2.97}$$

$$= m + C^{\frac{1}{2}}\widehat{C}^{-\frac{1}{2}}(\widehat{v} - \mathbb{E}\widehat{v}). \tag{2.98}$$

The second comes from setting $B = -K$ which leads to the possible choice $F = C$ and $A = I$ under which the map (2.92) *becomes*

$$v = \widehat{v} + \widehat{C}^{vy}(\widehat{C}^{yy})^{-1}(y^\dagger - \widehat{y}).$$

We refer to this as the Kalman transport *solution.* ■

Given the plethora of solutions for matrices (A, B), all of which effect the desired measure transport from π into the conditional, it is natural to ask how to choose a specific pair (A, B). One possibility is to use optimal transport. To this end we define, for positive definite $W \in \mathbb{R}^{d_v \times d_v}$,

$$I_W(A, B) = \frac{1}{2}\mathbb{E}\langle (v - \widehat{v}), W(v - \widehat{v})\rangle.$$

Theorem 2.B.6. *Let v be given by* (2.92)*. Consider the problem of finding minimizers of $I_W(A, B)$ over pairs (A, B) satisfying* (2.94)*,* (2.95)*; we refer to the resulting map evaluated at such an (A, B) as an* optimal transport in the $W-$weighted Euclidean distance. *For any positive definite W, such minimizers satisfy $B = 0$. In particular the Kalman transport solution is not an optimal transport solution.* ◇

Proof. We formulate the optimization problem over the pair (F, B) since, because $\widehat{C}$ is invertible, there is a bijection between (A, B) and (F, B). Let : denote the Frobenius inner-product. Then, under constraint (2.95),

$$I_W(A, B) = J_W(F) + \text{const}, \tag{2.99a}$$

$$J_W(F) = -F : W, \tag{2.99b}$$

where const denotes a matrix independent of A, B, F (with exact value changing from instance to instance). To see this note that

$$v - \widehat{v} = -(\widehat{v} - \mathbb{E}\widehat{v}) + A(\widehat{v} - \mathbb{E}\widehat{v}) + B(\widehat{y} - \mathbb{E}\widehat{y}) + K(y^\dagger - \mathbb{E}\widehat{y}).$$

Now, using (2.93),

$$\mathbb{E}\Big((v-\widehat{v})\otimes(v-\widehat{v})\Big) = \widehat{C} - \widehat{C}A^\top - A\widehat{C} - \widehat{C}^{vy}B^\top - B(\widehat{C}^{vy})^\top + C + \text{const.}$$

Noting that

$$I_W(A,B) = \frac{1}{2}\mathbb{E}\Big((v-\widehat{v})\otimes(v-\widehat{v})\Big) : W,$$

and that $D : W = D^\top : W$ for all D since W is symmetric, identity (2.99) follows from (2.95). It then also follows that minimization of $I_W(A,B)$ subject to the constraints given by (2.94), (2.95) is equivalent to minimization of $J_W(F)$ subject to the constraint (2.94).

To effect this latter minimization we introduce the Lagrange multiplier $L \in \mathbb{R}^{d_v \times d_v}$, symmetric because the constraint is symmetric, and define

$$\widetilde{J}_W(F,B,L) = -F : W + L : \big(F\widehat{C}^{-1}F^\top + B\widetilde{C}B^\top - C\big).$$

Differentiating with respect to F, B and L respectively gives

$$-W + 2LF\widehat{C}^{-1} = 0, \tag{2.100a}$$

$$2LB\widetilde{C} = 0, \tag{2.100b}$$

$$F\widehat{C}^{-1}F^\top + B\widetilde{C}B^\top = C. \tag{2.100c}$$

Since F, W, and $\widehat{C}$ in (2.100a) are all invertible, the Lagrange multiplier L is necessarily invertible. Furthermore, since $\widetilde{C}$ is invertible because Σ is, it follows from (2.100b) that $B = 0$ as required. □

Example 2.B.7. *Define symmetric matrix P, and from it symmetric A, by*

$$P = \Big(C^{\frac{1}{2}}\widehat{C}C^{\frac{1}{2}}\Big)^{-\frac{1}{2}}, \quad A = C^{\frac{1}{2}}PC^{\frac{1}{2}}.$$

We notice that if $B = 0$, as required for an optimal transport solution, then with this choice of the pair (A,B), equation (2.100c) has as a solution $F = A\widehat{C}$ and (2.100b) is satisfied. Furthermore (2.100a) delivers the Lagrange multiplier L. Note that the solution is independent of the specific choice of positive-definite W. ■

2.B.2 Mean Field Maps – No Simulated Data

Let $\widehat{v} \sim \widehat{\mu}$ and assume that $\widehat{y} = h(\widehat{v}) + \eta$ where $\eta \sim \mathsf{N}(0,\Gamma)$ is independent of $\widehat{v}$. Implicitly we have defined the joint distribution π of $(\widehat{v},\widehat{y})$. We note that then, expressed in terms of $\widehat{h} := h(\widehat{v})$ and quantities defined in (2.B.1),

$$\widehat{C}^{vh} := \mathbb{E}(\widehat{v}-\mathbb{E}\widehat{v})\otimes(\widehat{h}-\mathbb{E}\widehat{h}) = \widehat{C}^{vy},$$
$$\widehat{C}^{hh} := \mathbb{E}(\widehat{h}-\mathbb{E}\widehat{h})\otimes(\widehat{h}-\mathbb{E}\widehat{h}) = \widehat{C}^{yy} - \Gamma.$$

Thus we may rewrite (2.88) as

$$m = \widehat{m} + \widehat{C}^{vh}(\widehat{C}^{hh} + \Gamma)^{-1}\big(y^\dagger - \mathbb{E}\widehat{h}\big), \tag{2.101a}$$

$$C = \widehat{C} - \widehat{C}^{vh}(\widehat{C}^{hh} + \Gamma)^{-1}(\widehat{C}^{vh})^\top, \tag{2.101b}$$

In so doing we have eliminated reference to $\widehat{y}$ and our goal becomes the idenitification of maps of the form

$$v = R\widehat{v} + S\widehat{h} + r \tag{2.102}$$

so that, if $\widehat{v} \sim \widehat{\mu}$, then v has mean m and covariance C given by (2.101). We make the following assumptions on the covariance under $\mathsf{G}\pi$ and on the matrices R, S and vector r.

Assumptions 2.B.8. *The covariance under $\mathsf{G}\pi$ is invertible. The matrices R, S and vector r may depend on $y^\dagger$ and measure $\widehat{\mu}$ but not on the random variable $(\widehat{v}, \widehat{h})$; thus* (2.102) *takes the explicit form*

$$v = R(\pi, y^\dagger)\widehat{v} + S(\pi, y^\dagger)\widehat{h} + r(\pi, y^\dagger). \qquad \blacksquare$$

With these assumptions the pushforward under map (2.89), when constrained to match the desired first and second order statistics, defines a nonlinear map on the space of measures, and in particular on measure $\widehat{\mu}$. In what follows all expectations are computed under $\widehat{\mu}$. We note that matrix $\widehat{C}$ is invertible and that K in (2.90) may be rewritten as

$$K = \widehat{C}^{vh}(\widehat{C}^{hh} + \Gamma)^{-1}.$$

From equations (2.101a) and (2.102) the following is immediate.

Lemma 2.B.9. *Let Assumptions 2.B.8 hold and let $\widehat{v} \sim \widehat{\mu}$. If v given by* (2.102) *has mean given by (2.101a) then*

$$r = (I - R)\mathbb{E}\widehat{v} + Ky^\dagger - (S + K)\mathbb{E}\widehat{h}.$$

$\diamond$

As a consequence it follows that

$$v = \mathbb{E}\widehat{v} + R(\widehat{v} - \mathbb{E}\widehat{v}) + S(\widehat{h} - \mathbb{E}\widehat{h}) + K(y^\dagger - \mathbb{E}\widehat{h}). \tag{2.103}$$

and that

$$\mathbb{E}\Big((v - \mathbb{E}v) \otimes (v - \mathbb{E}v)\Big) = R\widehat{C}R^\top + S\widehat{C}^{hh}S^\top + R\widehat{C}^{vh}S^\top + S(\widehat{C}^{vh})^\top R. \tag{2.104}$$

Thus, to match the covariance of the conditioned random variable, we obtain

$$C = R\widehat{C}R^\top + S\widehat{C}^{hh}S^\top + R\widehat{C}^{vh}S^\top + S(\widehat{C}^{vh})^\top R. \tag{2.105}$$

Define

$$\check{C} = \widehat{C}^{hh} - (\widehat{C}^{vh})^\top(\widehat{C})^{-1}\widehat{C}^{vh}.$$

Theorem 2.B.10. *Let Assumptions 2.B.8 hold and let $\widehat{\nu} \sim \widehat{\mu}$. If s is given by Lemma 2.B.9 then ν defined by* (2.102) *has covariance (2.101b) if and only if real-valued matrices R and S are related by the identity*

$$E\widehat{C}^{-1}E^\top = C - S\check{C}S^\top,$$

where

$$E = R\widehat{C} + S(\widehat{C}^{vh})^\top.$$

◇

Proof. We complete the square on the right hand side of (2.105) to obtain

$$(R\widehat{C} + S(\widehat{C}^{vh})^\top)\widehat{C}^{-1}(R\widehat{C} + S(\widehat{C}^{vh})^\top)^\top + S\check{C}S^\top = C.$$

Rearranging gives the desired result. □

Define

$$\mathcal{S} = \{S \in \mathbb{R}^{d_v \times d_y} : C - S\check{C}S^\top \succ 0\}$$

and, for $S \in \mathcal{S}$ consider the eigenvalue problem

$$(C - S\check{C}S^\top)\psi^{(i)} = (o^{(i)})^2\psi^{(i)}, \quad \langle \psi^{(i)}, \psi^{(j)} \rangle = \delta_{ij},$$

noting that this has d_v real solutions, upto sign changes in the eigenvectors and assuming the $o^{(i)}$ to be non-negative. We now seek to express E in terms of $S \in \mathcal{S}$. Writing the SVD $E\widehat{C}^{-\frac{1}{2}} = W\Omega Z^\top$, where $W, Z \in \mathbb{R}^{d_v \times d_v}$ are orthogonal matrices and $\Omega \in \mathbb{R}^{d_v \times d_v}$ is a diagonal matrix, we see from (2.94) that

$$W\Omega^2 W^\top = C - S\check{C}S^\top$$

so that W has columns given by the $\{\psi^{(i)}\}_{i=1}^{d_v}$ and corresponding diagonal entries of Ω, $\pm o^{(i)}$. We define

$$W = \left(\psi^{(1)}, \cdots, \psi^{(d_v)}\right), \quad \Omega = \operatorname{diag}\left(\pm o^{(1)}, \ldots, \pm o^{(d_v)}\right). \tag{2.106}$$

Corollary 2.B.11. *For every $S \in \mathcal{S}$ the choices of R such that the pair (R, S) satisfies the criterion of Theorem 2.B.10 are defined as follows. For W, Ω as given in* (2.106)*, set*

$$E = W\Omega Z^\top \widehat{C}^{\frac{1}{2}},$$

where Z is an arbitrary orthogonal matrix in $\mathbb{R}^{d_v \times d_v}$. Then

$$R = \Big(E - S(\widehat{C}^{vh})^\top\Big)\widehat{C}^{-1}.$$

$\diamond$

Example 2.B.12. *As in Example 2.B.5 we highlight two examples, here corresponding to $S = 0$ and to $R = I$. The first, with $S = 0$, allows the choice $E = C^{\frac{1}{2}}\widehat{C}^{\frac{1}{2}}$ and hence $R = C^{\frac{1}{2}}\widehat{C}^{-\frac{1}{2}}$. We thus obtain* (2.97) *again.*

The second comes from setting $R = I$. This leads from (2.104) *to the following equation for S :*

$$S\widehat{C}^{hh}S^\top + \widehat{C}^{vh}S^\top + S(\widehat{C}^{vh})^\top = -\widehat{C}^{vh}(\widehat{C}^{hh} + \Gamma)^{-1}(\widehat{C}^{vh})^\top.$$

We seek a solution for S in the form

$$S = -\widehat{C}^{vh}Y^{-1}$$

and determine Y. We obtain the equation

$$Y^{-1}\widehat{C}^{hh}Y^{-T} - Y^{-T} - Y^{-1} = -(\widehat{C}^{hh} + \Gamma)^{-1}.$$

Thus, premultiplying by Y and post multiplying by $Y^\top$, we obtain

$$Y(\widehat{C}^{hh} + \Gamma)^{-1}Y^\top - Y - Y^\top + \widehat{C}^{hh} = 0$$

which factorizes to give

$$\Big(Y(\widehat{C}^{hh} + \Gamma)^{-1} - I\Big)(\widehat{C}^{hh} + \Gamma)\Big(Y(\widehat{C}^{hh} + \Gamma)^{-1} - I\Big)^\top = \Gamma.$$

Thus, taking the symmetric square root, we have

$$\Big(Y(\widehat{C}^{hh} + \Gamma)^{-1} - I\Big)(\widehat{C}^{hh} + \Gamma)^{\frac{1}{2}} = \Gamma^{\frac{1}{2}}.$$

Hence

$$\Big(Y(\widehat{C}^{hh} + \Gamma)^{-1} - I\Big) = \Gamma^{\frac{1}{2}}(\widehat{C}^{hh} + \Gamma)^{-\frac{1}{2}}.$$

Rearranging gives

$$Y(\widehat{C}^{hh}+\Gamma)^{-1} = \Gamma^{\frac{1}{2}}(\widehat{C}^{hh}+\Gamma)^{-\frac{1}{2}} + (\widehat{C}^{hh}+\Gamma)^{\frac{1}{2}}(\widehat{C}^{hh}+\Gamma)^{-\frac{1}{2}}.$$

Thus

$$Y = \left(\Gamma^{\frac{1}{2}} + (\widehat{C}^{hh}+\Gamma)^{\frac{1}{2}}\right)(\widehat{C}^{hh}+\Gamma)^{\frac{1}{2}}.$$

We obtain $S = -\widetilde{K}$ *where*

$$\widetilde{K} = \widehat{C}^{vh}\left((\widehat{C}^{hh}+\Gamma) + \Gamma^{1/2}(\widehat{C}^{hh}+\Gamma)^{1/2}\right)^{-1}.$$

The map (2.103) *becomes*

$$\begin{aligned} v &= \widehat{v} - \widetilde{K}(\widehat{h} - \mathbb{E}\widehat{h}) + \widehat{C}^{vh}(\widehat{C}^{hh}+\Gamma)^{-1}(y^\dagger - \mathbb{E}\widehat{h}) \\ &= \widehat{v} - \widetilde{K}(\widehat{h} - \mathbb{E}\widehat{h}) + K(y^\dagger - \mathbb{E}\widehat{h}) \\ &= m + (\widehat{v} - \widehat{m}) - \widetilde{K}(\widehat{h} - \mathbb{E}\widehat{h}). \end{aligned}$$

■

2.B.3 Minimum Variance Approximation

In the two preceding subsections we have identified an uncountable set of transport maps that match the second order statistics of true transport. Among all these, the *Kalman transport* (2.54) has a particular appeal because it is constructed around the *Kalman gain* familiar from filtering in the linear Gaussian setting. In this subsection we show how the principle of minimizing the variance within a class of *linear* estimators of the state, given observation, leads to this choice of transport map. We believe that the second order transport approach, which we highlight in the main text, provides a more fundamental viewpoint on the mean field models which underpin ensemble Kalman methods; however the minimum variance perspective is widely adopted in the statistics and geophysics communities (see Subsection 2.2.7) and hence has an important place in the subject of ensemble Kalman methods.

Recall that the Kalman transport approach leads back to the map (2.14c), motivated by control theoretic considerations, and identifies a specific choice of Kalman gain K_n, a choice which depends on the law of the predicted state and data. In the Gaussian case the Kalman transport map (2.54) exactly generates the desired transport, a fact which we discuss in Example 2.2.15. The general form of approximate stochastic second order transport maps which we study using the second order transport approach in the main text is (2.49). Our goal here is to motivate a specific choice of $\widetilde{T}^S$ in (2.49c), in particular to derive (2.54c). In this subsection we achieve this by

first defining, and identifying, the *best linear unbiased estimator*, BLUE for short. From this we will derive Kalman transport.

Lemma 2.B.13. *Assume that* $\Gamma \succ 0$. *Let all expectations be computed under Law* $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ *and consider* m_{n+1}^{BL} *in the form*

$$m_{n+1}^{\mathrm{BL}} = a' + B\widehat{y}_{n+1}.$$

Define $\widehat{C}_{n+1}, \widehat{C}_{n+1}^{vy}$ *and* $\widehat{C}_{n+1}^{yy}$ *by* (2.31)*,* (2.32) *and define*

$$\mathsf{I}(a', B) := \mathbb{E}|\widehat{v}_{n+1} - m_{n+1}^{\mathrm{BL}}|^2.$$

Then m_{n+1}^{BL} *is said to be the BLUE of* $\widehat{v}_{n+1}$ *given* $\widehat{y}_{n+1}$*, if vector* a' *and matrix* B *are independent of* $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$*, but may depend on Law* $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$*, and are chosen to minimize* $\mathsf{I}(a', B)$. *Then*

$$m_{n+1}^{\mathrm{BL}} = \mathbb{E}\widehat{v}_{n+1} + \widehat{C}_{n+1}^{vy}(\widehat{C}_{n+1}^{yy})^{-1}(\widehat{y}_{n+1} - \mathbb{E}\widehat{y}_{n+1}). \tag{2.107}$$

Furthermore, the estimator (2.107) is unbiased, that is,

$$\mathbb{E}(\widehat{v}_{n+1} - m_{n+1}^{\mathrm{BL}}) = 0,$$

and its covariance with respect to Law $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$*,*

$$C_{n+1}^{\mathrm{BL}} := \mathbb{E}\big((\widehat{v}_{n+1} - m_{n+1}^{\mathrm{BL}})(\widehat{v}_{n+1} - m_{n+1}^{\mathrm{BL}})^{\top}\big),$$

is given by

$$C_{n+1}^{\mathrm{BL}} = \widehat{C}_{n+1} - \widehat{C}_{n+1}^{vy}(\widehat{C}_{n+1}^{yy})^{-1}(\widehat{C}_{n+1}^{vy})^{\top}. \tag{2.108}$$

◇

Proof. Noting that we may, without loss of generality, reparametrize the proposed form of m_{n+1}^{BL} as

$$m_{n+1}^{\mathrm{BL}} = \mathbb{E}\widehat{v}_{n+1} + a + B(\widehat{y}_{n+1} - \mathbb{E}\widehat{y}_{n+1}),$$

we see that the desired objective to be minimized over vector-matrix pair (a, B) is

$$\begin{aligned}
\mathsf{J}(a, B) &:= \frac{1}{2}\mathbb{E}|\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1} - a - B(\widehat{y}_{n+1} - \mathbb{E}\widehat{y}_{n+1})|^2 \\
&= \frac{1}{2}\mathbb{E}|\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}|^2 + \frac{1}{2}|a|^2 \\
&\qquad + \frac{1}{2}\mathbb{E}|B(\widehat{y}_{n+1} - \mathbb{E}\widehat{y}_{n+1})|^2 - \mathbb{E}\langle\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}, B(\widehat{y}_{n+1} - \mathbb{E}\widehat{y}_{n+1})\rangle \\
&= \frac{1}{2}\mathbb{E}|\widehat{v}_{n+1} - \mathbb{E}\widehat{v}_{n+1}|^2 + \frac{1}{2}|a|^2 + \frac{1}{2}(BB^{\top}) : \widehat{C}_{n+1}^{yy} - B : \widehat{C}_{n+1}^{vy}.
\end{aligned}$$

Clearly the minimizer with respect to a is achieved by setting $a = 0$. Differentiating with respect to B, noting that $\widehat{C}^{yy}_{n+1}$ is symmetric, shows that $B = K_n$ given by (2.33). That the resulting pair is indeed a minimizer, and not another critical point, follows from the fact that $\widehat{C}^{yy}_{n+1}$ is positive-definite; this is implied by the assumption $\Gamma \succ 0$. Thus we obtain the estimator (2.107). A straightforward calculation shows that the estimator is unbiased and that its covariance is given by (2.108). □

We now connect the BLUE to an approximate (second order) transport map. To this end, note that (2.107) may be viewed as mapping $\widehat{y}_{n+1}$ into $m^{\mathrm{BL}}_{n+1} = m^{\mathrm{BL}}(\widehat{y}_{n+1})$. With this notation we define $m^{\dagger}_{n+1}$ by

$$m^{\dagger}_{n+1} = m^{\mathrm{BL}}(y^{\dagger}_{n+1}) \tag{2.109a}$$

$$= \mathbb{E}\widehat{v}_{n+1} + \widehat{C}^{vy}_{n+1}(\widehat{C}^{yy}_{n+1})^{-1}(y^{\dagger}_{n+1} - \mathbb{E}\widehat{y}_{n+1}). \tag{2.109b}$$

We may now make the following connection between BLUE and Kalman transport.

Theorem 2.B.14. *Let $(\widehat{v}_{n+1}, \widehat{y}_{n+1})$ be distributed according to measure π_{n+1}. Then the transport map (2.54c) has the properties:*

$$\mathbb{E}v_{n+1} = m^{\dagger}_{n+1}$$
$$\mathbb{E}\big((v_{n+1} - m^{\dagger}_{n+1})(v_{n+1} - m^{\dagger}_{n+1})^{\top}\big) = C^{\mathrm{BL}}_{n+1},$$

where $m^{\dagger}_{n+1}$ and C^{BL}_{n+1} are given by (2.109) *and* (2.108) *respectively.* ◇

Proof. Note that the transport map (2.54c) can now be reformulated as

$$v_{n+1} = \widehat{v}_{n+1} + (m^{\dagger}_{n+1} - m^{\mathrm{BL}}_{n+1}).$$

Thus

$$v_{n+1} = m^{\dagger}_{n+1} + \widehat{v}_{n+1} - m^{\mathrm{BL}}_{n+1}.$$

The desired properties of the mean and covariance follow from Lemma 2.B.13. □

Remark 2.B.15. *We now have two derivations of the Kalman gain, the approximate transport derivation from Subsection 2.2.5.4, and the minimum variance derivation given here. We include a third, dimensional, argument that motivates its form. Let* `state` *denote the physical units associated with the state variable and* `data` *those associated with the observation. Then the physical units of the gain matrix K should equal* `state/data`*. We note that $\widehat{C}^{yy}$ has units of* `data` *squared whilst the units of $\widehat{C}^{vy}$ are the product of* `state` *and* `data`*. If we then constrain the gain to be*

determined by covariance matrices involving the state and the data it is natural to choose it to be formed as $\widehat{C}^{vy}(\widehat{C}^{yy})^{-1}$ *where* $\widehat{C}^{yy}$ *is an estimate of covariance in the data, and* $\widehat{C}^{vy}$ *an estimate of covariance between state and data. Making a choice of this type leads to the right units for the gain, since the innovation has units* `data`*, and relies only on use of first and second order statistics. This dimensional argument motivates the form* (2.33) *as derived in both Subsections 2.B.3 and 2.2.4.* ■

Chapter 3

ACCURACY OF THE ENSEMBLE KALMAN FILTER IN THE NONLINEAR SETTING

3.1 Introduction

This chapter describes a setting in which the ensemble Kalman filter (EnKF) accurately approximates the filtering distribution of discrete time, partially observed, stochastic dynamical systems. In Subsection 3.1.1 we set the work in context by providing a literature review. Subsection 3.1.2 details our contributions and overviews the chapter structure. Subsection 3.1.3 contains a precise statement of the filtering distribution which we approximate using the EnKF.

3.1.1 Literature Review

Algorithms for filtering employ noisy observations, arising from a (possibly stochastic) dynamical system, to estimate the distribution of the system state conditional on the observations. The Kalman filter (Kalman, 1960) determines the filtering distribution exactly for linear Gaussian dynamics and observations. The extended Kalman filter generalizes the Kalman filtering methodology to nonlinear problems and is based on a linearization approximation; see Andrew H Jazwinski (2007) and B. D. Anderson and Moore (2012). The linearization approximation leads to an inexact distribution for nonlinear problems, and furthermore requires evaluation of covariance matrices, making the methodology impractical for large-scale geophysical applications (M. Ghil et al., 1981). Particle filters or sequential Monte Carlo methods (Doucet, Freitas, and Gordon, 2001; Chopin and Papaspiliopoulos, 2020) offer an alternative methodology for nonlinear filtering problems, allowing recovery of the exact filtering distribution in the large particle limit. However, this class of methods scales poorly with dimension, and in particular suffers weight collapse, making its application to geophysical problems prohibitive; see for example Snyder et al. (2008), Bickel, B. Li, and Bengtsson (2008), Rebeschini and Van Handel (2015), and Agapiou et al. (2017).

For this reason the EnKF has emerged as popular robust methodology for filtering, especially in high dimensional problems. We refer to paper J. A. Carrillo et al. (2024) for detailed discussion of the literature in areas aligned with, or adjacent to, the results we establish about the EnKF. In this literature review we confine ourselves

primarily to overviewing a narrow subset of the work extensively reviewed in J. A. Carrillo et al. (2024), with focus on several of the open questions detailed in our closing remarks. The EnKF was introduced in the seminal paper Evensen (1994). Its success stems from the low-rank approximation of large covariances, cheaply computed using an ensemble of particles, which allows the methodology to be deployed in geophysical applications. Quantifying the accuracy of the ensemble Kalman filter has been of great practical and theoretical interest. In paper D. T. Kelly, K. J. Law, and Andrew M Stuart (2014) accuracy of the EnKF, viewed as a state estimator and not in terms of the filtering distribution, is established. This work is undertaken in both discrete and continuous time. This continuous time analysis of state estimation accuracy was improved upon in J. d. Wiljes, Reich, and Stannat (2018), where a variance inflation condition from D. T. Kelly, K. J. Law, and Andrew M Stuart (2014) was dispensed with. The continuous time model used in D. T. Kelly, K. J. Law, and Andrew M Stuart (2014) and J. d. Wiljes, Reich, and Stannat (2018), however, was not rigorously justified, as a continuous time limit of the discrete time EnKF, until Lange and Stannat (2021).

For nonlinear non-Gaussian settings, which are of particular practical interest, analysis of the accuracy of the EnKF in relation to the true filtering distribution, rather than just state estimation, remains in its infancy. The papers Le Gland, Monbet, and Tran (2011b) and Jan Mandel, Loren Cobb, and J. D. Beezley (2011) study this issue in the linear Gaussian setting where the mean field limit of the Kalman filter is exact; they demonstrate that the ensemble Kalman filter may be viewed as an interacting particle system approximation of the mean field limit, and establish Monte Carlo type error bounds. The recent article Calvello, Reich, and Andrew M. Stuart (2025) overviews the formulation of ensemble Kalman methods using mean field dynamical systems and provides a platform from which the analyses of Le Gland, Monbet, and Tran (2011b) and Jan Mandel, Loren Cobb, and J. D. Beezley (2011) may be generalized beyond the linear Gaussian setting. We refer the reader to Calvello, Reich, and Andrew M. Stuart (2025) for a comprehensive overview of mean-field limits of ensemble Kalman filters. In the recent paper J. A. Carrillo et al. (2024) the authors establish stability properties of the discrete time mean field ensemble Kalman filter and use them to prove accuracy of the filter in a near-Gaussian setting[1]. However,

[1]We note that in J. A. Carrillo et al. (2024) the authors use "near-Gaussian" to characterize the joint distribution of state and data, which leads to a near-Gaussian filtering distribution. However, in their context, the additional assumption of bounded dynamics and observation model vector fields limits their results to the setting of near-constant vector fields. In this work we are more general and use "near-linear" to characterize the dynamics and observation model vector fields: this also leads

the paper does not consider finite particle approximations of the mean field, and the conditions on the dynamics-observation model require boundedness of the vector fields arising. In this chapter we address both these issues, establishing error bounds between the finite particle discrete time ensemble Kalman filter and the true filtering distribution in settings where the dynamics and observation vector fields grow linearly.

We note that extending our work in this chapter, concerning filter accuracy, to the continuous time setting established in Lange and Stannat (2021) remains open. Likewise an important open question stemming from our work is to establish estimates valid on the infinite time horizon, in either the discrete or the continuous time setting. In the continuous time setting this is studied for ensemble Kalman approximation of the Kalman-Bucy filter, when the true filter is Gaussian, in P. Del Moral and Tugaut (2018); the paper A. N. Bishop and Pierre Del Moral (2023) comprehensively reviews this line of work. The paper Pierre Del Moral and Horton (2023b) contains long time stability estimates for the sample means and sample variances of the discrete ensemble Kalman filter in nonlinear, non-Gaussian settings, but is confined to one dimension. Stability estimates of this kind will be needed in any generalization of our work to the infinite time horizon.

3.1.2 Contributions and Outline

We make the following contributions.

1. Theorem 3.2.2 is a stability result for the discrete time mean field ensemble Kalman filter in the setting of dynamical models and Lipschitz obervation operators that grow at most linearly at infinity.

2. Theorem 3.2.4 quantifies the error between the discrete time mean field ensemble Kalman filter and the true filter in the setting of dynamical models and Lipschitz observation operators that are near-linear.

3. Theorem 3.2.5 quantifies the error between the finite particle discrete time ensemble Kalman filter (found as an interacting particle system approximation of the mean field) and the true filter in the setting of near-linear, Lipschitz dynamical models and linear observation operators.

to near-Gaussian filtering distributions.

In going beyond the work in J. A. Carrillo et al. (2024), the current chapter simultaneously addresses a more applicable problem class, by allowing linear growth in the dynamics and observation operators, and confronts the substantial technical challenges which arise from doing so. Indeed, allowing for unbounded vector fields enables the application of this analysis to the setting of the Kalman filter; in addition, it enables the integration of this mean field analysis with finite-particle error estimates developed in the literature, such as in Le Gland, Monbet, and Tran (2011b). Bounds on moments of the filtering distribution and mean field ensemble Kalman filter must be established; these bounds grow in the number of iterations which is in contrast to J. A. Carrillo et al. (2024) where the L^∞ bounds on model and observation operator ensure a uniform bound on moments. A further challenge is presented by the need to establish stability bounds for the conditioning map, giving rise to the true filter, and the transport map giving rise to the ensemble Kalman filter. These results exhibit dependence on moments in the stability constants and require control of the growth given by the dynamics; this is again in contrast to J. A. Carrillo et al. (2024) where the L^∞ bounds allow for a uniform control.

We introduce the filtering problem in Subsection 3.1.3. In Section 3.2 we outline the main results of the chapter concerning the ensemble Kalman filter. In Subsection 3.2.1 we formulate the ensemble Kalman filter that we consider in this chapter along with the relevant assumptions we will use in the analysis. We state a stability theorem for the mean field formulation of the ensemble Kalman filter in Subsection 3.2.2, hence Contribution 1. We leverage this result in Subsection 3.2.3 to derive a theorem quantifying the error between the mean field ensemble Kalman filter and the true filter, Contribution 2. Finally, in Subsection 3.2.4, we make use of the results of the previous subsections to state a theorem quantifying the error between the (finite particle) ensemble Kalman filter itself and the true filter, yielding Contribution 3. Various technical results, used in the proof of our three main theorems, may be found in the appendices. We conclude with closing remarks in Section 3.3.

3.1.3 Filtering Distribution

Here we introduce the hidden Markov model that gives rise to the filtering distribution. We employ notation similar to that established in Calvello, Reich, and Andrew M. Stuart (2025) and J. A. Carrillo et al. (2024). We consider $\{u_j\}_{j\in[\![0,J]\!]} \subset \mathbb{R}^{d_u}$ to be unknown states to be determined from associated observations $\{y_j\}_{j\in[\![1,J]\!]} \subset \mathbb{R}^{d_y}$. We assume the states and observations to be governed by the following stochastic

dynamics and data model:

$$u_{j+1} = \Psi(u_j) + \xi_j, \qquad \xi_j \sim \mathsf{N}(0, \Sigma), \tag{3.1a}$$

$$y_{j+1} = h(u_{j+1}) + \eta_{j+1}, \qquad \eta_{j+1} \sim \mathsf{N}(0, \Gamma). \tag{3.1b}$$

We assume that the initial state is a Gaussian random variable $u_0 \sim \mathsf{N}(m_0, C_0)$ and that the following independence condition is satisfied:

$$u_0 \perp\!\!\!\perp \xi_0 \perp\!\!\!\perp \cdots \perp\!\!\!\perp \xi_{J-1} \perp\!\!\!\perp \eta_1 \perp\!\!\!\perp \cdots \perp\!\!\!\perp \eta_J. \tag{3.2}$$

We define the conditional distribution of the state u_j given a realization $Y_j^\dagger := \{y_1^\dagger, \ldots, y_j^\dagger\}$ as the *filtering distribution*, which we denote by μ_j. We also refer to μ_j as the *true filter*. In the context of this chapter, data $Y_j^\dagger$ is assumed to be a collection of realizations from (3.1). As formulated in Kody Law, A. Stuart, and Konstantinos Zygalakis (2015) and Reich and Colin Cotter (2015), the evolution of the filtering distribution may be written as

$$\mu_{j+1} = \mathsf{L}_j \mathsf{P} \mu_j, \tag{3.3}$$

where P and L_j are maps from $\mathcal{P}(\mathbb{R}^{d_u})$ into itself and which effect what are referred to in the data assimilation community as the *prediction* and *analysis* steps, respectively (Asch, Marc Bocquet, and Nodet, 2016). The prediction operator P is linear and is determined by the Markov kernel arising from (3.1a). On the other hand, the operator L_j is nonlinear and encodes the incorporation of the new data point $y_{j+1}^\dagger$ using Bayes' theorem. These operators are defined via action on probability measures with Lebesgue density μ as

$$\mathsf{P}\mu(u) = \frac{1}{\sqrt{(2\pi)^{d_u} \det \Sigma}} \int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}|u - \Psi(v)|_\Sigma^2\right) \mu(v)\, \mathrm{d}v, \tag{3.4a}$$

$$\mathsf{L}_j\mu(u) = \frac{\exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(u)\right|_\Gamma^2\right) \mu(u)}{\displaystyle\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(U)\right|_\Gamma^2\right) \mu(U)\, \mathrm{d}U}. \tag{3.4b}$$

We may rewrite the analysis map L_j as the composition $\mathsf{B}_j\mathsf{Q}$, where the operators $\mathsf{Q}\colon \mathcal{P}(\mathbb{R}^{d_u}) \to \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ and $\mathsf{B}_j\colon \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}) \to \mathcal{P}(\mathbb{R}^{d_u})$ are defined by

$$\mathsf{Q}\mu(u, y) = \frac{1}{\sqrt{(2\pi)^{d_y} \det \Gamma}} \exp\left(-\frac{1}{2}\left|y - h(u)\right|_\Gamma^2\right) \mu(u), \tag{3.5a}$$

$$\mathsf{B}_j\mu(u) = \frac{\mu(u, y_{j+1}^\dagger)}{\displaystyle\int_{\mathbb{R}^{d_u}} \mu(U, y_{j+1}^\dagger)\, \mathrm{d}U}. \tag{3.5b}$$

Linear operator Q maps a probability measure with density μ into the law of random variable $\big(U, h(U) + \eta\big)$, with $U \sim \mu$ independent of $\eta \sim \mathsf{N}(0, \Gamma)$. Nonlinear operator B_j effects conditioning on the datum $y_{j+1}^{\dagger}$. Hence, we may reformulate (3.3) as

$$\mu_{j+1} = \mathsf{B}_j \mathsf{Q} \mathsf{P} \mu_j. \tag{3.6}$$

3.2 The Ensemble Kalman Filter

We begin in Subsection 3.2.1 by introducing the specific form of the mean field ensemble Kalman filter that we consider in this chapter; other versions leading to implementable algorithms and for which similar analysis may be developed can be found in Calvello, Reich, and Andrew M. Stuart (2025). In this subsection we also outline the various assumptions that will be employed for the results of the chapter. We note that the subsections that follow proceed with increasingly restrictive assumptions on dynamics and observation operator. However, all of the theorems allow for linear growth of the vector fields defining the dynamics and the observation processes. In Subsection 3.2.2 we state and prove a stability theorem, which shows that the error between the true filter and the mean field ensemble Kalman filter may be controlled by the error between the true filter and its Gaussian projection on the joint space of state and observations. In Subsection 3.2.3 we establish an error estimate which shows that the mean field ensemble Kalman filter provides an accurate approximation of the true filtering distribution for dynamics and observation operators that are close-to-linear. In Subsection 3.2.4 we deploy the results of the two preceding subsections to state a theorem quantifying the error between the finite particle ensemble Kalman filter itself and the true filtering distribution.

3.2.1 The Algorithm

The ensemble Kalman filter as implemented in practice may be derived as a particle approximation of various mean field dynamics (Calvello, Reich, and Andrew M. Stuart, 2025). The particular mean field ensemble Kalman filter that we consider in this chapter may be written as

$$\widehat{u}_{j+1} = \Psi(u_j) + \xi_j, \qquad \xi_j \sim \mathsf{N}(0, \Sigma), \tag{3.7a}$$

$$\widehat{y}_{j+1} = h(\widehat{u}_{j+1}) + \eta_{j+1}, \qquad \eta_{j+1} \sim \mathsf{N}(0, \Gamma), \tag{3.7b}$$

$$u_{j+1} = \widehat{u}_{j+1} + C^{uy}\left(\widehat{\pi}_{j+1}^{\mathrm{EK}}\right) C^{yy}\left(\widehat{\pi}_{j+1}^{\mathrm{EK}}\right)^{-1}\left(y_{j+1}^{\dagger} - \widehat{y}_{j+1}\right), \tag{3.7c}$$

where $\widehat{\pi}^{\mathrm{EK}}_{j+1} = \mathrm{Law}(\widehat{u}_{j+1}, \widehat{y}_{j+1})$ and where the independence condition (3.2) holds. We refer the reader back to Section 1.2 for the definition of the covariance matrices appearing in (3.7c). We denote by μ^{EK}_j the law of u_j. We aim to formulate the evolution of μ^{EK}_j in terms of maps on probability measures, hence we introduce, for a given $y^\dagger_{j+1}$, the map $\mathsf{T}_j \colon \mathcal{P}^2_{\succ 0}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}) \to \mathcal{P}(\mathbb{R}^{d_u})$ defined by

$$\mathsf{T}_j \pi = \mathscr{T}(\bullet, \bullet; \pi, y^\dagger_{j+1}) \sharp \pi. \tag{3.8}$$

Here for given $\pi \in \mathcal{P}^2_{\succ 0}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, $z \in \mathbb{R}^{d_y}$, the transport map $\mathscr{T}$ is defined as

$$\mathscr{T}(\bullet, \bullet; \pi, z) \colon \mathbb{R}^{d_u} \times \mathbb{R}^{d_y} \to \mathbb{R}^{d_u}; \tag{3.9a}$$

$$(u, y) \mapsto u + C^{uy}(\pi) C^{yy}(\pi)^{-1} \big(z - y\big). \tag{3.9b}$$

Evolution of the probability measure μ^{EK}_j may then be written (see Calvello, Reich, and Andrew M. Stuart (2025)) as

$$\mu^{\mathrm{EK}}_{j+1} = \mathsf{T}_j \mathsf{Q}\mathsf{P} \mu^{\mathrm{EK}}_j. \tag{3.10}$$

We note that a specific instance of map $\mathscr{T}$ defined in (3.9) appears in (3.7c) and is determined by probability measure $\pi := \widehat{\pi}^{\mathrm{EK}}_{j+1}$ (here equal to $\mathsf{Q}\mathsf{P}\mu^{\mathrm{EK}}_j$) and data $z := y^\dagger_{j+1}$. We refer to J. A. Carrillo et al. (2024) for a step-by-step argument detailing why the law of u_j in (3.7) evolves according to (3.10). The operator T_j is equivalent, over the Gaussians $\mathcal{G}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}) \subset \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, to the conditioning operator B_j; see Lemma 3.B.7. We recall that in the particular case where μ_0 is Gaussian, which we assume throughout the chapter, and the operators Ψ and h are linear, the mean field ensemble Kalman filter (3.10) coincides with the filtering distribution (3.6). However, this chapter focuses on the aim of analyzing the accuracy of ensemble Kalman methods when Ψ and h are not assumed to be linear.

The ensemble Kalman filter as implemented in practice may be derived as a particle approximation of the mean field dynamics as defined by sample paths in (3.7) or as an evolution on measures in (3.10). To this end, we observe that for any π in the range of Q, $C^{yy}(\pi) = C^{hh}(\pi) + \Gamma$ and $C^{uy}(\pi) = C^{uh}(\pi)$, using the notation introduced in, and following, (1.4). This is since the noise in the observation component defining π is then independent of the state component defining π. Using this observation, the

particle approximation of (3.7) takes the form

$$\widehat{u}_{j+1}^{(i)} = \Psi\big(u_j^{(i)}\big) + \xi_j^{(i)}, \qquad \xi_j^{(i)} \sim \mathsf{N}(0,\Sigma), \tag{3.11a}$$

$$\widehat{y}_{j+1}^{(i)} = h\big(\widehat{u}_{j+1}^{(i)}\big) + \eta_{j+1}^{(i)}, \qquad \eta_{j+1}^{(i)} \sim \mathsf{N}(0,\Gamma), \tag{3.11b}$$

$$u_{j+1}^{(i)} = \widehat{u}_{j+1}^{(i)} + C^{uh}\left(\widehat{\pi}_{j+1}^{\mathrm{EK},N}\right)\left(C^{hh}\left(\widehat{\pi}_{j+1}^{\mathrm{EK},N}\right) + \Gamma\right)^{-1}\big(y_{j+1}^{\dagger} - \widehat{y}_{j+1}^{(i)}\big), \tag{3.11c}$$

where $\widehat{\pi}_{j+1}^{\mathrm{EK},N}$ is the empirical measure

$$\widehat{\pi}_{j+1}^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\widehat{u}_{j+1}^{(i)}, \widehat{y}_{j+1}^{(i)}\right)},$$

and $\xi_j^{(i)} \sim \mathsf{N}(0,\Sigma)$ i.i.d. in both i and j and $\eta_{j+1}^{(i)} \sim \mathsf{N}(0,\Gamma)$ i.i.d. in both i and j; furthermore, the set of $\{\xi_j^{(i)}\}$ are independent from the set of $\{\eta_{j+1}^{(i)}\}$. Choosing to express the particle approximation of the covariance in observation space through C^{hh} and Γ ensures invertibility, provided Γ is invertible. From the particles evolving according to the dynamics in (3.11) we define the empirical measure

$$\mu_{j+1}^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^{N} \delta_{u_{j+1}^{(i)}}, \tag{3.12}$$

whose evolution describes the ensemble Kalman filter.

Our theorems in Subsections 3.2.2 and 3.2.3 regard the relationship between the true filter (3.6) and the mean field ensemble Kalman filter (3.10). In Subsection 3.2.2 we consider the setting in which Ψ and h exhibit linear growth but are not assumed to be linear, hence the true filter is not Gaussian; in Subsection 3.2.3 we then consider small perturbations of the Gaussian setting that arise when the vector fields Ψ and h are close to affine. The theorem in Subsection 3.2.4 concerns the relationship between the true filter (3.6) and the ensemble Kalman filter (3.12). In this subsection, we combine existing analysis on the convergence of the ensemble Kalman filter to the mean field ensemble Kalman filter with the results from the previous subsections to state and prove an error estimate between the ensemble Kalman filter and the true filter in the nonlinear setting. Specifically, we study the case of a vector field Ψ that is a bounded perturbation away from affine and an affine vector field h. To state our theorems we will use the following set of assumptions.

Assumption H1. *There exist positive constants $\kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \sigma$, and γ such that the data $\{y_j^{\dagger}\}$, the vector fields (Ψ, h), and the covariances (Σ, Γ) satisfy:*

(H1) *data* $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ *lies in set* $B_y \subset \mathbb{R}^{KJ}$ *defined by*

$$B_y := \left\{ Y^\dagger \in \mathbb{R}^{KJ} : \max_{j \in [\![1,J]\!]} |y_j^\dagger| \leqslant \kappa_y \right\};$$

(H2) *covariance matrices* Σ *and* Γ *are positive definite:* $\Sigma \succcurlyeq \sigma I_{d_u}$ *and* $\Gamma \succcurlyeq \gamma I_{d_y}$ *for positive* σ *and* γ*;*

(H3) *function* $\Psi \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ *satisfies* $\left|\Psi(u)\right| \leqslant \kappa_\Psi\big(1 + |u|\big)$ *for all* $u \in \mathbb{R}^{d_u}$*;*

(H4) *function* $h \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ *satisfies* $\left|h(u)\right| \leqslant \kappa_h\big(1 + |u|\big)$ *for all* $u \in \mathbb{R}^{d_u}$*;*

(H5) *function* $h \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ *satisfies* $|h|_{C^{0,1}} \leqslant \ell_h < \infty$*;*

Assumption V. *The vector fields* (Ψ, h) *are affine, i.e. they satisfy the following:*

(V1) *The function* $\Psi \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ *satisfies* $\Psi(u) := Mu + b$*, for some* $M \in \mathbb{R}^{d_u \times d_u}$ *and* $b \in \mathbb{R}^{d_u}$*.*

(V2) *The function* $h \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ *satisfies* $h(u) := Hu + w$*, for some* $H \in \mathbb{R}^{d_y \times d_u}$ *and* $w \in \mathbb{R}^{d_y}$*.*

3.2.2 Stability Theorem: Mean Field Ensemble Kalman Filter

In this section we consider the setting of dynamics and observation operator satisfying the linear growth assumptions contained in Assumption H1. We recall the definition of the weighted total variation metric in Definition 1.2.1 and make the following remark.

Remark 3.2.1. *The following remarks concern the distance* d_g*:*

- *If* μ_1, μ_2 *have Lebesgue densities* ρ_1*,* ρ_2*, then*

$$d_g(\mu_1, \mu_2) = \int g(v)|\rho_1(v) - \rho_2(v)| \, \mathrm{d}v.$$

- *Unlike the usual total variation distance, the weighted metric in Definition 1.2.1 enables control of the differences* $|\mathcal{M}(\mu_1) - \mathcal{M}(\mu_2)|$ *and* $\|C(\mu_1) - C(\mu_2)\|$*. This is the content of Lemma 3.B.6, stated in Subsection 3.B.1, which is used in the proof of a key auxiliary result (Lemma 3.B.11).*

Informally the result of Theorem 3.2.2 shows that, under these assumptions, and if the true filtering distributions $(\mu_j)_{j\in[\![0,J-1]\!]}$ are close to Gaussian after lifting to the joint state and data space, then the distribution μ_j^{EK} given by the mean field ensemble Kalman filter (3.10) is close to the filtering distribution μ_j given by (3.6) for all $j \in [\![0, J]\!]$.

Theorem 3.2.2 (Stability: Mean Field Ensemble Kalman Filter). *Assume that the probability measures $(\mu_j)_{j\in[\![0,J]\!]}$ and $(\mu_j^{\mathrm{EK}})_{j\in[\![0,J]\!]}$ are given respectively by the dynamical systems* (3.6) *and* (3.10)*, initialized at the Gaussian probability measure $\mu_0 = \mu_0^{\mathrm{EK}} \in \mathcal{G}(\mathbb{R}^{d_u})$. That is,*

$$\mu_{j+1} = \mathsf{B}_j\mathsf{QP}\mu_j, \qquad \mu_{j+1}^{\mathrm{EK}} = \mathsf{T}_j\mathsf{QP}\mu_j^{\mathrm{EK}}.$$

If Assumption H1 holds, then there exists $C = C\big(\mathcal{M}_{\max\{3+d_u,4+d_y\}}(\mu_0), \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma, J\big)$ such that

$$d_g(\mu_J^{\mathrm{EK}}, \mu_J) \leqslant C \max_{j\in[\![0,J-1]\!]} d_g(\mathsf{QP}\mu_j, \mathsf{GQP}\mu_j).$$

Proof of Theorem 3.2.2 below relies on the following auxiliary results, all proved in Section 3.B.

1. For any probability measure μ with finite first and second order polynomial moments $\mathcal{M}_1(\mu)$ and $\mathcal{M}_2(\mu)$, the means of $\mathsf{P}\mu$ and $\mathsf{QP}\mu$ are bounded from above, and their covariances are bounded both from above and from below. The constants in these bounds depend only on the parameters κ_Ψ, κ_h, Σ, Γ and on $\mathcal{M}_1(\mu)$ and $\mathcal{M}_2(\mu)$. See Lemmas 3.B.1 and 3.B.2.

2. Let $(\mu_j)_{j\in[\![1,J]\!]}$ and $(\mu_j^{\mathrm{EK}})_{j\in[\![1,J]\!]}$ denote the probability measures obtained respectively from the dynamical systems (3.6) and (3.10), initialized at the same Gaussian measure $\mu_0 = \mu_0^{\mathrm{EK}} \in \mathcal{G}(\mathbb{R}^{d_u})$. Then for any integer $q \geqslant 2$, there exist constants $M_q, M_q^{\mathrm{EK}} < \infty$ depending on $\mathcal{M}_q(\mu_0), \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, J$ so that

$$\max_{j\in[\![0,J]\!]} \mathcal{M}_q(\mu_j) \leqslant M_q \quad \text{and} \quad \max_{j\in[\![0,J]\!]} \mathcal{M}_q\big(\mu_j^{\mathrm{EK}}\big) \leqslant M_q^{\mathrm{EK}}.$$

   See Lemmas 3.B.3 and 3.B.5. This will facilitate use of the stability results from Items 6 and 7.

3. For any Gaussian measure $\mu \in \mathcal{G}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, we have that $\mathsf{B}_j\mu = \mathsf{T}_j\mu$. See Lemma 3.B.7 and Calvello, Reich, and Andrew M. Stuart (2025) and J. A. Carrillo et al. (2024).

4. The map P is Lipschitz on $\mathcal{P}(\mathbb{R}^{d_u})$ for the metric d_g. The Lipschitz constant L_{P} depends only on the parameters κ_Ψ and Σ. See Lemma 3.B.8.

5. The map Q is Lipschitz on $\mathcal{P}(\mathbb{R}^{d_u})$ for the metric d_g. The Lipschitz constant L_{Q} depends only on the parameters κ_h and Γ. See Lemma 3.B.9.

6. The map B_j satisfies that, for all $\pi \in \{\mathsf{QP}\mu : \mu \in \mathcal{P}(\mathbb{R}^{d_u}) \text{ and } \mathcal{M}_2(\mu) < \infty\} \subset \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, it holds

$$\forall j \in [\![0, J]\!], \qquad d_g(\mathsf{B}_j \mathsf{G}\pi, \mathsf{B}_j \pi) \leqslant C_{\mathsf{B}} d_g(\mathsf{G}\pi, \pi),$$

   where $C_{\mathsf{B}} = C_{\mathsf{B}}\big(\mathcal{M}_2(\mu), \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\big)$. Namely, this item regards the stability of the operator B_j between a probability measure and its Gaussian projection. See Lemma 3.B.10.

7. The map T_j satisfies the following bound: for all $R \geqslant 1$, it holds for all $\pi \in \mathcal{P}_R(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ and $p \in \{\mathsf{QP}\mu : \mu \in \mathcal{P}(\mathbb{R}^{d_u}) \text{ and } \mathcal{M}_{2\cdot\max\{3+d_u, 4+d_y\}}(\mu) < \infty\} \subset \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ that

$$\forall j \in [\![0, J]\!], \qquad d_g(\mathsf{T}_j \pi, \mathsf{T}_j p) \leqslant L_{\mathsf{T}}\, d_g(\pi, p),$$

   for a constant $L_{\mathsf{T}} = L_{\mathsf{T}}\big(R, \mathcal{M}_{2\cdot\max\{3+d_u, 4+d_y\}}(\mu), \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma\big)$. Namely, this item may be regarded as a local Lipschitz continuity result. Refer to Lemma 3.B.11.

Proof of Theorem 3.2.2. Within the following argument we make reference to the list of items outlined above. We begin by defining for ease of exposition the following measure of difference between the true filter and its Gaussian projection:

$$\varepsilon := \max_{j \in [\![0, J-1]\!]} d_g(\mathsf{QP}\mu_j, \mathsf{GQP}\mu_j). \tag{3.13}$$

We assume throughout the argument that $j \in [\![0, J-1]\!]$. The proof rests upon the following application of the triangle inequality:

$$\begin{aligned} d_g(\mu_{j+1}^{\mathrm{EK}}, \mu_{j+1}) &= d_g\left(\mathsf{T}_j \mathsf{QP}\mu_j^{\mathrm{EK}}, \mathsf{B}_j \mathsf{QP}\mu_j\right) \\ &\leqslant d_g\left(\mathsf{T}_j \mathsf{QP}\mu_j^{\mathrm{EK}}, \mathsf{T}_j \mathsf{QP}\mu_j\right) + d_g\left(\mathsf{T}_j \mathsf{QP}\mu_j, \mathsf{T}_j \mathsf{GQP}\mu_j\right) + \quad (3.14a) \\ &\quad + d_g\left(\mathsf{B}_j \mathsf{GQP}\mu_j, \mathsf{B}_j \mathsf{QP}\mu_j\right). \quad (3.14b) \end{aligned}$$

Here we have used the fact that $\mathsf{T}_j \mathsf{GQP}\mu_j = \mathsf{B}_j \mathsf{GQP}\mu_j$ by Item 3 (Lemma 3.B.7). Item 2 (Lemmas 3.B.3 and 3.B.5) shows that, for any integer $q \geqslant 2$, there exist

constants $M_q, M_q^{\mathrm{EK}} < \infty$ depending on $\mathcal{M}_q(\mu_0), \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, J$ so that

$$\max_{j\in[\![0,J]\!]} \mathcal{M}_q(\mu_j) \leqslant M_q \quad \text{and} \quad \max_{j\in[\![0,J]\!]} \mathcal{M}_q\big(\mu_j^{\mathrm{EK}}\big) \leqslant M_q^{\mathrm{EK}}.$$

Therefore by Item 1 (Lemma 3.B.2), there is a constant $R \geqslant 1$, depending on the covariances Σ, Γ, the bounds κ_Ψ and κ_h from Assumption H1, and the moment bounds M_2 and M_2^{EK}, such that for any $j \in [\![0, J-1]\!]$ it holds that $\mathsf{QP}\mu_j, \mathsf{QP}\mu_j^{\mathrm{EK}} \in \mathcal{P}_R(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$. In view of Items 4, 5 and 7 (Lemmas 3.B.8, 3.B.9 and 3.B.11), the first term in (3.14b) satisfies

$$d_g\left(\mathsf{T}_j\mathsf{QP}\mu_j^{\mathrm{EK}}, \mathsf{T}_j\mathsf{QP}\mu_j\right) \leqslant L_{\mathsf{T}}\big(R, M_{2\cdot\max\{3+d_u,4+d_y\}}\big) L_{\mathsf{Q}} L_{\mathsf{P}} d_g\left(\mu_j^{\mathrm{EK}}, \mu_j\right), \quad (3.15)$$

where, for conciseness, we have omitted the dependence of the constants in the bound on $\kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$. By definition of R it holds that $\mathsf{GQP}\mu_j \in \mathcal{G}_R(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, hence the second term in (3.14b) may be bounded using Item 7 (Lemma 3.B.11) and the definition in of ε in (3.13). Indeed, it holds that

$$\begin{aligned} d_g\left(\mathsf{T}_j\mathsf{QP}\mu_j, \mathsf{T}_j\mathsf{GQP}\mu_j\right) &\leqslant L_{\mathsf{T}}\big(R, M_{2\cdot\max\{3+d_u,4+d_y\}}\big) d_g\left(\mathsf{QP}\mu_j, \mathsf{GQP}\mu_j\right) \\ &\leqslant L_{\mathsf{T}}\big(R, M_{2\cdot\max\{3+d_u,4+d_y\}}\big)\varepsilon. \end{aligned}$$

Using Item 6 (Lemma 3.B.10) and the definition of ε in (3.13) we establish the following bound on the third term in (3.14b):

$$d_g\left(\mathsf{B}_j\mathsf{GQP}\mu_j, \mathsf{B}_j\mathsf{QP}\mu_j\right) \leqslant C_{\mathsf{B}} d_g\left(\mathsf{GQP}\mu_j, \mathsf{QP}\mu_j\right) \leqslant C_{\mathsf{B}}\varepsilon.$$

Therefore, setting $\ell = L_{\mathsf{T}}\big(R, M_{2\cdot\max\{3+d_u,4+d_y\}}\big) L_{\mathsf{Q}} L_{\mathsf{P}}$, we have that

$$d_g(\mu_{j+1}^{\mathrm{EK}}, \mu_{j+1}) \leqslant \ell d_g(\mu_j^{\mathrm{EK}}, \mu_j) + \big(L_{\mathsf{T}}\big(R, M_{\max\{3+d_u,4+d_y\}}\big) + C_{\mathsf{B}}\big)\varepsilon.$$

The conclusion then follows from the discrete Grönwall lemma, using the fact that $\mu_0^{\mathrm{EK}} = \mu_0$. □

3.2.3 Error Estimate: Mean Field Ensemble Kalman Filter

It is possible to deduce from the result of Theorem 3.2.2 that the error between the true filter and the mean field ensemble Kalman filter can be arbitrarily small if the true filter is arbitrarily close to its Gaussian projection, in state-observation space. This condition on the true filter, which we refer to as "closeness to Gaussian", can be satisfied in the setting of unbounded vector fields by considering small perturbations of affine vector fields, as we prove in Proposition 3.2.3. Combining Proposition 3.2.3 with Theorem 3.2.2 gives an error estimate for the mean field ensemble Kalman filter, yielding Theorem 3.2.4.

Proposition 3.2.3 (Approximation Result). *Suppose that the data* $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ *and the matrices* (Σ, Γ) *satisfy Assumptions (H1) and (H2). Fix* $\kappa_\Psi, \kappa_h > 0$ *and assume that* $\Psi_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ *and* $h_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ *are affine functions and hence satisfy Assumptions (V1) and (V2), respectively, while also satisfying Assumptions (H3) and (H4) with* κ_Ψ, κ_h. *Let* $(\mu_j)_{j\in[\![0,J]\!]}$ *denote the true filtering distribution associated with functions* (Ψ, h), *initialized at the Gaussian probability measure* $\mu_0 = \mathsf{N}(m_0, C_0) \in \mathcal{G}(\mathbb{R}^{d_u})$. *Then, for any* $J \in \mathbb{Z}^+$, *there is* $C = C(m_0, C_0, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, J) > 0$ *such that for all* $\varepsilon \in [0, 1]$ *and all* $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$, *it holds that*

$$\max_{j\in[\![0,J-1]\!]} d_g(\mathsf{GQP}\mu_j, \mathsf{QP}\mu_j) \leqslant C\varepsilon. \tag{3.16}$$

Proof. In what follows $(\mu_j^0)_{j\in[\![0,J]\!]}$ and $(\mu_j)_{j\in[\![0,J]\!]}$ denote the true filtering distributions associated with functions (Ψ_0, h_0) and (Ψ, h), respectively, initialized at the same Gaussian measure $\mathsf{N}(m_0, C_0)$. Furthermore, we let P_0 and Q_0 denote the kernel integral operators (3.4a) and (3.5a) defined by the specific vector fields (Ψ_0, h_0). By Lemma 3.B.3, the filtering distributions have bounded second moments. Let

$$\mathcal{R} = \max_{j\in[\![0,J-1]\!]} \left(\left| y_{j+1}^\dagger \right|^2,\ 1 + \mathcal{M}_2\big(\mu_j^0\big),\ 1 + \mathcal{M}_2\big(\mu_j\big) \right).$$

Throughout this proof, C denotes a constant whose value is irrelevant in the context, depends only on the constants $m_0, C_0, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, j$ (in particular, it does not depend on ε, Ψ, h), and may change from line to line. Fix $j \in [\![0, J-1]\!]$. Note that the filtering distribution defined by (Ψ_0, h_0) is Gaussian. Then, using the triangle inequality and Gaussianity of $\mathsf{Q}_0\mathsf{P}_0\mu_j^0$ we obtain

$$\begin{aligned} d_g(\mathsf{GQP}\mu_j, \mathsf{QP}\mu_j) &\leqslant d_g(\mathsf{GQP}\mu_j, \mathsf{GQ}_0\mathsf{P}_0\mu_j^0) + d_g(\mathsf{GQ}_0\mathsf{P}_0\mu_j^0, \mathsf{QP}\mu_j) \\ &= d_g(\mathsf{GQP}\mu_j, \mathsf{GQ}_0\mathsf{P}_0\mu_j^0) + d_g(\mathsf{Q}_0\mathsf{P}_0\mu_j^0, \mathsf{QP}\mu_j). \end{aligned}$$

We note that since the filters have bounded first and second order polynomial moments, by Lemmas 3.B.1 and 3.B.2 we may deduce that there exists $R \geqslant 1$ such that

$$\forall j \in [\![0, J-1]\!], \qquad \mathsf{P}_0\mu_j^0, \in \mathcal{P}_R(\mathbb{R}^{d_u}), \qquad \mathsf{QP}\mu_j, \mathsf{Q}_0\mathsf{P}_0\mu_j^0 \in \mathcal{P}_R(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}). \tag{3.17}$$

The second part of this display allows direct application of Lemma 3.C.1, which concerns the local Lipschitz continuity result for the Gaussian projection operator

G, to obtain

$$\begin{aligned} d_g(\mathsf{GQP}\mu_j, \mathsf{QP}\mu_j) &\leqslant C d_g(\mathsf{Q}_0\mathsf{P}_0\mu_j^0, \mathsf{QP}\mu_j) \\ &\leqslant C d_g(\mathsf{Q}_0\mathsf{P}_0\mu_j^0, \mathsf{QP}_0\mu_j^0) + C d_g(\mathsf{QP}_0\mu_j^0, \mathsf{QP}\mu_j). \end{aligned}$$

Using (3.66) from Lemma 3.C.2, noting that since $\mathsf{P}_0\mu^0 \in \mathcal{P}_R(\mathbb{R}^{d_u})$ by (3.17) it holds that $\mathcal{M}_2(\mathsf{P}_0\mu_j^0) \leqslant d_u R$, and using the Lipschitz continuity of Q (Lemma 3.B.9) we deduce that

$$\begin{aligned} d_g(\mathsf{GQP}\mu_j, \mathsf{QP}\mu_j) &\leqslant C\varepsilon(1 + d_u R) + C d_g(\mathsf{P}_0\mu_j^0, \mathsf{P}\mu_j) \\ &\leqslant C\varepsilon(1 + d_u R) + C d_g\big(\mathsf{P}_0\mu_j^0, \mathsf{P}\mu_j^0\big) + C d_g\big(\mathsf{P}\mu_j^0, \mathsf{P}\mu_j\big) \\ &\leqslant C\varepsilon(1 + d_u R) + C\varepsilon\mathcal{R} + C d_g(\mu_j^0, \mu_j), \end{aligned}$$

where the second inequality follows by the triangle inequality and the third inequality follows from bounding $d_g(\mathsf{P}_0\mu_j^0, \mathsf{P}\mu_j)$ using (3.65) from Lemma 3.C.2 and from the Lipschitz continuity of P (Lemma 3.B.9). The statement then follows from Lemma 3.C.3. □

Theorem 3.2.4 (Error Estimate: Mean Field Ensemble Kalman Filter). *Assume that the probability measures $(\mu_j)_{j\in[\![0,J]\!]}$ and $(\mu_j^{\mathrm{EK}})_{j\in[\![0,J]\!]}$ are given respectively by the dynamical systems* (3.6) *and* (3.10) *initialized at the Gaussian measure $\mu_0 = \mu_0^{\mathrm{EK}} \in \mathcal{G}(\mathbb{R}^{d_u})$. That is,*

$$\mu_{j+1} = \mathsf{B}_j\mathsf{QP}\mu_j, \qquad \mu_{j+1}^{\mathrm{EK}} = \mathsf{T}_j\mathsf{QP}\mu_j^{\mathrm{EK}}.$$

Suppose that the data $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ and the matrices (Σ, Γ) satisfy Assumptions (H1) and (H2). Let the vector fields Ψ_0, h_0 be affine functions satisfying Assumptions (V1) and (V2), respectively, while also satisfying Assumptions (H3) and (H4) with some constants $\kappa_\Psi, \kappa_h > 0$. Additionally, let $\ell_\Psi > 0$ be a constant. Then there exists a constant $C = C\big(\mathcal{M}_q(\mu_0), \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma, J\big)$, where $q := 2 \cdot \max\{3 + d_u, 4 + d_y\}$, such that for all $\varepsilon \in [0, 1]$ and all $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$ with h satisfying Assumption (H5), it holds that

$$d_g(\mu_J^{\mathrm{EK}}, \mu_J) \leqslant C\varepsilon.$$

Proof. Since Assumption H1 is satisfied, it is possible to apply the result of Theorem 3.2.2 to deduce that there exists $C = C\big(\mathcal{M}_{2\cdot\max\{3+d_u,4+d_y\}}(\mu_0), J, \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma\big)$ such that

$$d_g(\mu_J^{\mathrm{EK}}, \mu_J) \leqslant C \max_{j\in[\![0,J-1]\!]} d_g(\mathsf{QP}\mu_j, \mathsf{GQP}\mu_j).$$

Additionally, since $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$ for Ψ_0, h_0 satisfying Assumptions (V1) and (V2), and moreover Assumptions (H3) and (H4), we may apply Proposition 3.2.3 to deduce the result. □

3.2.4 Error Estimate: Finite Particle Ensemble Kalman Filter

In this subsection, we combine the results from the work in Le Gland, Monbet, and Tran (2011b) with stability Theorem 3.2.2, together with approximation Theorem 3.2.4, to derive a quantitative error estimate between the finite particle ensemble Kalman filter and the true filter in the nonlinear setting. In order to define an appropriate metric we introduce the following class of vector fields.

Assumption P1. *The vector field $\phi : \mathbb{R}^{d_u} \to \mathbb{R}$ satisfies for any $u, v \in \mathbb{R}^{d_u}$ the condition*

$$|\phi(u) - \phi(v)| \leqslant L_\phi |u - v| \big(1 + |u|^\varsigma + |v|^\varsigma\big),$$

for some $\varsigma \geqslant 0$ and for some $L_\phi > 0$. We note that for any such ϕ, there exists $R_\phi > 0$ which depends on L_ϕ so that $|\phi(u)| \leqslant R_\phi\big(1 + |u|^{\varsigma+1}\big)$ for any $u \in \mathbb{R}^{d_u}$.

We will prove various technical lemmas under the Assumption P1; these may be useful beyond the confines of this chapter. However for our theorems we use the more specific Assumption P2 which enables the control of first and second moments.

Assumption P2. *The vector field $\phi : \mathbb{R}^{d_u} \to \mathbb{R}$ satisfies for any $u, v \in \mathbb{R}^{d_u}$ the condition*

$$|\phi(u) - \phi(v)| \leqslant L_\phi |u - v| \big(1 + |u| + |v|\big),$$

for some $L_\phi > 0$. Note that for any such ϕ, there exists $R_\phi > 0$ depending on L_ϕ so that $|\phi(u)| \leqslant R_\phi\big(1 + |u|^2\big)$ for any $u \in \mathbb{R}^{d_u}$.

Theorem 3.2.5 (Error Estimate: Finite Particle Ensemble Kalman Filter). *Assume that the probability measures $(\mu_j)_{j\in[\![0,J]\!]}$ and $(\mu_j^{\mathrm{EK},N})_{j\in[\![0,J]\!]}$ for finite $J \in \mathbb{N}$ are obtained respectively from the dynamical systems* (3.6) *and* (3.12)*, initialized at the Gaussian probability measure $\mu_0 \in \mathcal{G}(\mathbb{R}^{d_u})$ and at the empirical measure $\mu_0^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^N \delta_{u_0^{(i)}}$ for $u_0^{(i)} \sim \mu_0$ i.i.d. samples, respectively. That is,*

$$\mu_{j+1} = \mathsf{B}_j \mathsf{Q} \mathsf{P} \mu_j, \qquad \mu_{j+1}^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^N \delta_{u_{j+1}^{(i)}},$$

where $u_{j+1}^{(i)}$ evolve according to the iteration in (3.11)*. Suppose that the data $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ and the matrices (Σ, Γ) satisfy Assumptions (H1) and (H2). Assume that*

the vector field h is linear, and let κ_h, ℓ_h be positive constants such that Assumptions (H4) and (H5) are satisfied. Furthermore, let the vector field Ψ_0 be an affine function satisfying Assumption (V1) as well as Assumption (H3) with $\kappa_\Psi > 0$. Additionally, let $\ell_\Psi > 0$ be a constant. Then, there exists a constant $C = C\big(\mathcal{M}_q(\mu_0), R_\phi, L_\phi, \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \ell_\Psi, \Sigma, \Gamma, J\big)$, where $q \coloneqq 2\cdot\max\{3+d_u, 4+d_y, 2\cdot 4^J\}$, such that for all $\varepsilon \in [0,1]$ and all $\Psi \in B_{L^\infty}\big(\Psi_0, \varepsilon\big)$ satisfying $|\Psi|_{C^{0,1}} \leqslant \ell_\Psi < \infty$, the following bound holds for any ϕ satisfying Assumption P2:

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J[\phi]\right|^2\right)^{1/2} \leqslant C\Big(\frac{1}{\sqrt{N}} + \varepsilon\Big).$$

We note that the expectation appearing in the bound from Theorem 3.2.5 is with respect to the law of the particles $u_j^{(i)}$, which evolve according to (3.11) and which define the empirical measure $\mu_{j+1}^{\mathrm{EK},N}$. The proof presented hereafter relies on the following elements.

1. We apply the triangle inequality in order to employ two distinct results concerning the mean field ensemble Kalman filter. The proof thus involves quantifying the error between finite particle ensemble Kalman filter and the mean field ensemble Kalman filter, as discussed in Item 2, and the error between the mean field ensemble Kalman filter and the true filter, as discussed in Item 3.

2. The work in Le Gland, Monbet, and Tran (2011b) establishes a Monte Carlo error estimate, with rate of $1/\sqrt{N}$, between the empirical measure $\mu^{\mathrm{EK},N}$, representing the particle ensemble Kalman filter, and the measure μ^{EK} describing the evolution of the mean field ensemble Kalman filter. In particular, this holds under Assumptions (H1) and (H2) on the data and covariances of the noise processes, the linearity of the vector field h and Assumption (H3) on Ψ with the additional assumption that $|\Psi|_{C^{0,1}} \leqslant \ell_\Psi < \infty$. In Lemma 3.D.1 we provide a self-contained proof of Le Gland, Monbet, and Tran, 2011b, Theorem 5.2 to gain insight into the dependence of the constant prefactor multiplying $1/\sqrt{N}$ on the parameters of the Gaussian initial condition and the number of steps J.

3. We assume that the data $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ and the matrices (Σ, Γ) satisfy Assumptions (H1) and (H2), and assume that h satisfies Assumption (V2). Furthermore, we assume Ψ satisfies Assumption (H3) and $\Psi \in B_{L^\infty}(\Psi_0, \varepsilon)$

with Ψ_0 satisfying Assumption (V1). These assumptions allow us to apply the result from Theorem 3.2.4.

Proof. Recall that μ_j^{EK} is the mean field ensemble Kalman filter, here initialized at the same Gaussian μ_0 as the true filter. We fix a function ϕ satisfying Assumption P2 and apply the triangle inequality to deduce that

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J[\phi]\right|^2\right)^{\frac{1}{2}} \leqslant \left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}} + \left(\mathbb{E}\left|\mu_J^{\mathrm{EK}}[\phi] - \mu_J[\phi]\right|^2\right)^{\frac{1}{2}}. \tag{3.18}$$

Since μ_J^{EK} and μ_J are deterministic probability measures, it holds that

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK}}[\phi] - \mu_J[\phi]\right|^2\right)^{1/2} = \left|\mu_J^{\mathrm{EK}}[\phi] - \mu_J[\phi]\right|.$$

Since ϕ satisfies Assumption P2, it follows that

$$\left|\mu_J^{\mathrm{EK}}[\phi] - \mu_J[\phi]\right| \leqslant \sup_{|\phi| \leqslant R_\phi(1+|u|^2)} \left|\mu_J^{\mathrm{EK}}[\phi] - \mu_J[\phi]\right| = R_\phi \cdot d_g\big(\mu_J^{\mathrm{EK}}, \mu_J\big).$$

We note that Assumption H1 holds as h is assumed to be affine; since we additionally assume that $\Psi \in B_{L^\infty}\big(\Psi_0, \varepsilon\big)$, where Ψ_0 is an affine function, we may apply the result of Theorem 3.2.4 to deduce that

$$d_g\big(\mu_J^{\mathrm{EK}}, \mu_J\big) \leqslant C\varepsilon, \tag{3.19}$$

where C is a constant depending on $\mathcal{M}_{2\cdot\max\{3+d_u,4+d_y\}}(\mu_0), J, \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma$.

The fact that h is assumed to be linear and the additional assumption that $|\Psi|_{C^{0,1}} \leqslant \ell_\Psi < \infty$ allows us to apply the result of Lemma 3.D.1, so that

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{1/2} \leqslant \frac{C}{\sqrt{N}}, \tag{3.20}$$

where C is a constant depending on $\mathcal{M}_{4^{J+1}}(\mu_0), J, R_\phi, L_\phi, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$. In view of (3.18), combining (3.19) and (3.20) yields the desired result. □

3.3 Discussion and Future Directions

In this chapter we have presented the first analysis, in the setting of a filtering problem defined by nonlinear state space dynamics, quantifying the error between the empirical measure obtained by the finite particle discrete time ensemble Kalman filter and the true filtering distribution. The analysis for the EnKF outlined is based

on the proof methodology developed for particle filters in Rebeschini and Van Handel (2015) and developed for the mean field ensemble Kalman filter in J. A. Carrillo et al. (2024). Our work goes beyond J. A. Carrillo et al. (2024) by considering finite particle filters and by allowing for dynamical models and observation operators that are unbounded. The work builds substantially on the new avenue for analysis of the ensemble Kalman methodology that was introduced in J. A. Carrillo et al. (2024) but leaves open numerous avenues of investigation for further work; we detail these.

1. As surveyed in Calvello, Reich, and Andrew M. Stuart (2025), the ensemble Kalman filtering methodology may be used for solving inverse problems and for sampling; it is of interest to extend the analysis in our work to the ensemble Kalman based inversion algorithms outlined in that paper.

2. There is a substantial body of literature studying the continuous time limits of ensemble Kalman methods; see Calvello, Reich, and Andrew M. Stuart (2025) for a review. Performing analysis analogous to that presented here, but in the continuous time setting, would be of interest.

3. For large scale applications, there has been recent wide interest in replacing the dynamical model Ψ, representing the solution operator obtained via a high fidelity numerical solver, with a cheap to evaluate surrogate. Multifidelity ensemble methods (Bach and Michael Ghil, 2023) allow the use of a small number of particles evolved according to the high fidelity solver and a large number evolved according to the surrogate. Extending our error analysis to incorporate the effect of model error would be of interest in this context.

4. We highlight that there may be other "near-Gaussian problems", arising in the small noise or large data volume limits, that would be interesting to study, using Bernstein-von-Mises theorems; however this will require new analysis that makes observability assumptions and controls behavior of the constants in the analysis with respect to noise and number of observations.

5. We have derived results in a conditional setting, arising from assuming a uniform bound on the set of considered observations; removing this assumption, and understanding the probabilistic nature of error bounds with respect to observational noise, would be of interest.

6. Our error estimate comparing the EnKF with the true filter blows up exponentially in time. In the presence of stability assumptions about the true filter,

it is natural to ask whether our results can be extended to the infinite time horizon.

3.A Auxiliary Results

We first state and prove a standard result that will be used throughout the chapter.

Lemma 3.A.1. *Let X be a random vector in $\mathbb{R}^{d_u}$ with finite second moment. Then it holds that*

$$\mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(X - \mathbb{E}[X]\big)^{\mathsf{T}}\Big] = \mathbb{E}\big[XX^{\mathsf{T}}\big] - \mathbb{E}[X]\mathbb{E}[X]^{\mathsf{T}} \tag{3.21}$$

and

$$\forall a \in \mathbb{R}^{d_u}, \qquad \mathbb{E}\Big[\big(X - a\big)\big(X - a\big)^{\mathsf{T}}\Big] \succcurlyeq \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(X - \mathbb{E}[X]\big)^{\mathsf{T}}\Big]. \tag{3.22}$$

Proof. The statements follow from the following equalities:

$$\begin{aligned}\mathbb{E}\Big[(X-a)(X-a)^{\mathsf{T}}\Big] &= \mathbb{E}\Big[\Big(\big(X - \mathbb{E}[X]\big) + \big(\mathbb{E}[X] - a\big)\Big)\Big(\big(X - \mathbb{E}[X]\big) + \big(\mathbb{E}[X] - a\big)\Big)^{\mathsf{T}}\Big] \\ &= \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(X - \mathbb{E}[X]\big)^{\mathsf{T}}\Big] + \big(\mathbb{E}[X] - a\big)\big(\mathbb{E}[X] - a\big)^{\mathsf{T}}. \quad \square\end{aligned}$$

Lemma 3.A.2. *Let $g(\bullet; m, S)$ be the Lebesgue density of $\mathsf{N}(m, S)$ and $\mathcal{S}^n_\alpha$ the set of symmetric $n \times n$ matrices M satisfying*

$$\frac{1}{\tau} I_n \preccurlyeq M \preccurlyeq \tau I_n.$$

Then for any $n \in \mathbb{N}^+$ and $\tau \geqslant 1$, there is $L_{n,\tau} > 0$ such that for all parameters $(c_1, m_1, S_1) \in \mathbb{R} \times \mathbb{R}^n \times \mathcal{S}^n_\tau$ and $(c_2, m_2, S_2) \in \mathbb{R} \times \mathbb{R}^n \times \mathcal{S}^n_\tau$ it holds that

$$\|h\|_\infty \leqslant L_{n,\tau} \|h\|_1, \qquad h(y) = c_1 g(y; m_1, S_1) - c_2 g(y; m_2, S_2).$$

Proof. The lemma as stated may be found in J. A. Carrillo et al., 2024, Lemma A.4, where a complete proof is given. $\square$

Lemma 3.A.3. *Let P and Q denote the operators on probability measures given respectively in* (3.4a) *and* (3.5a)*. Let Assumptions (H2) to (H5) be satisfied and suppose that $\mu \in \mathcal{P}(\mathbb{R}^{d_u})$ satisfies, for some $q > 0$, the moment bound*

$$\mathcal{M}_q(\mu) = \int_{\mathbb{R}^{d_u}} |x|^q \, \mu(\mathrm{d}x) < \infty. \tag{3.23}$$

Then there is $L = L(\mathcal{M}_q(\mu), \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma) > 0$ such that for all $(u_1, u_2, y) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_u} \times \mathbb{R}^{d_y}$, the probability density of $p = \mathsf{QP}\mu$ satisfies

$$\big|p(u_1, y) - p(u_2, y)\big| \leqslant L|u_1 - u_2| \min\left\{\max\left\{\frac{1}{1 + |u_1|^q}, \frac{1}{1 + |u_2|^q}\right\}, \frac{1}{1 + |y|^q}\right\}. \tag{3.24}$$

Proof. Throughout this proof, C denotes a constant whose value is irrelevant in the context, depends only on $\mathcal{M}_q(\mu), \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma$, and may change from line to line. Sometimes we write the dependence explicitly to indicate which parameters are involved.

Step 1. Bounding the density of $\mathsf{P}\mu$ We first rewrite

$$\big(1+|u|^q\big)\mathsf{P}\mu(u) = C(\Sigma)\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right)\frac{1+|u|^q}{1+|v|^q}\,\big(1+|v|^q\big)\mu(\mathrm{d}v).$$

We note that

$$S(\Sigma,\kappa_\Psi) := \sup_{(u,v)\in\mathbb{R}^{d_u}\times\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right)\frac{1+|u|^q}{1+|v|^q} < \infty;$$

this may be seen observing that

$$\exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right)\frac{1+|u|^q}{1+|v|^q} \leqslant \exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right)\frac{(1+2^{q-1}|u-\Psi(v)|^q)+2^{q-1}|\Psi(v)|^q}{1+|v|^q}. \tag{3.25}$$

The first term on the righthand side of (3.25) may be bounded by noting that $1+2^{q-1}e^{-x^2}x^q$ is uniformly bounded in $x\in\mathbb{R}$; on the other hand the second term may be bounded by applying Assumption (H3). Now, using (3.23), we obtain that

$$\mathsf{P}\mu(u) \leqslant \frac{C\big(\Sigma,\kappa_\Psi,\mathcal{M}_q(\mu)\big)}{1+|u|^q}. \tag{3.26}$$

Step 2. Establishing Lipschitz continuity of $u\mapsto\mathsf{P}\mu(u)$ Since $g(x) := \mathrm{e}^{-x^2}$ has derivative $2x\,\mathrm{e}^{-x^2}$ and $|x\,\mathrm{e}^{-x^2}| \leqslant \mathrm{e}^{-\frac{x^2}{2}}$ for all $x\in\mathbb{R}$, it holds for some ξ between $|a|$ and $|b|$ that

$$\forall (a,b)\in\mathbb{R}\times\mathbb{R},\qquad \left|\mathrm{e}^{-a^2}-\mathrm{e}^{-b^2}\right| = |b-a|\,|g'(\xi)| \leqslant 2|b-a|\left(\mathrm{e}^{-\frac{a^2}{2}}+\mathrm{e}^{-\frac{b^2}{2}}\right). \tag{3.27}$$

Using this inequality with $a^2 = \frac{1}{2}|u_1-\Psi(v)|_\Sigma^2$ and $b^2 = \frac{1}{2}|u_2-\Psi(v)|_\Sigma^2$, the triangle inequality, and equivalence of norms, we deduce that, for all $(u_1,u_2,v)\in\mathbb{R}^{d_u}\times\mathbb{R}^{d_u}\times\mathbb{R}^{d_u}$,

$$\begin{aligned}&\left|\exp\left(-\frac{1}{2}|u_1-\Psi(v)|_\Sigma^2\right)-\exp\left(-\frac{1}{2}|u_2-\Psi(v)|_\Sigma^2\right)\right| \\ &\qquad\leqslant C|u_2-u_1|\left(\exp\left(-\frac{1}{4}|u_1-\Psi(v)|_\Sigma^2\right)+\exp\left(-\frac{1}{4}|u_2-\Psi(v)|_\Sigma^2\right)\right)\end{aligned}$$

for constant $C = C(\Sigma)$. Integrating out the v variable with respect to μ we obtain that

$$|\mathsf{P}\mu(u_1) - \mathsf{P}\mu(u_2)| \leqslant C|u_1 - u_2| \int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{4}|u_1 - \Psi(v)|_\Sigma^2\right) \mu(\mathrm{d}v) + C|u_1 - u_2| \int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{4}|u_2 - \Psi(v)|_\Sigma^2\right) \mu(\mathrm{d}v).$$

The integrals on the right-hand side can be bounded as in the first step, which leads to the inequality

$$\forall (u_1, u_2) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_u}, \qquad |\mathsf{P}\mu(u_1) - \mathsf{P}\mu(u_2)| \leqslant C|u_1 - u_2| \max\left\{\frac{1}{1 + |u_1|^q}, \frac{1}{1 + |u_2|^q}\right\}. \tag{3.28}$$

Step 3. Obtaining a coarse estimate In view of the elementary inequality (3.27) and the assumed Lipschitz continuity of h, it holds that, for constant $C = C(\Gamma, \ell_h)$

$$\begin{aligned} &\left|\mathsf{N}\big(h(u_1), \Gamma\big)(y) - \mathsf{N}\big(h(u_2), \Gamma\big)(y)\right| \\ &\leqslant C|u_1 - u_2| \exp\left(-\frac{1}{4}|y - h(u_1)|_\Gamma^2\right) + C|u_1 - u_2| \exp\left(-\frac{1}{4}|y - h(u_2)|_\Gamma^2\right). \end{aligned} \tag{3.29}$$

Using the decomposition

$$\begin{aligned} p(u_1, y) - p(u_2, y) &= \mathsf{P}\mu(u_1)\, \mathsf{N}\big(h(u_1), \Gamma\big)(y) - \mathsf{P}\mu(u_2)\, \mathsf{N}\big(h(u_2), \Gamma\big)(y) \\ &= \big(\mathsf{P}\mu(u_1) - \mathsf{P}\mu(u_2)\big)\, \mathsf{N}\big(h(u_1), \Gamma\big)(y) \\ &\quad + \mathsf{P}\mu(u_2)\Big(\mathsf{N}\big(h(u_1), \Gamma\big)(y) - \mathsf{N}\big(h(u_2), \Gamma\big)(y)\Big), \end{aligned}$$

and employing (3.26), (3.28), and (3.29), we deduce that

$$\begin{aligned} |p(u_1, y) - p(u_2, y)| \leqslant C|u_1 - u_2| &\max\left\{\frac{1}{1 + |u_1|^q}, \frac{1}{1 + |u_2|^q}\right\} \\ &\times \max\left\{\exp\left(-\frac{1}{4}|y - h(u_1)|_\Gamma^2\right), \exp\left(-\frac{1}{4}|y - h(u_2)|_\Gamma^2\right)\right\}. \end{aligned} \tag{3.30}$$

Note that, for the function $h(u) = u$, the quantity multiplying $|u_1 - u_2|$ on the right-hand does not tend to 0 along the sequence $\left(u_1^{(n)}, u_2^{(n)}, y^{(n)}\right) = (0, n, n)$. For our purposes in this work, we need the finer estimate (3.24); establishing this bound is the aim of the next two steps.

Step 4. Bounding the density $p(u,y)$ Recall that $p(u,y) = \mathsf{P}\mu(u)\mathcal{N}\big(h(u),\Gamma\big)(y)$. We prove in this step the inequality

$$p(u,y) \leqslant C \min\left\{\frac{1}{1+|u|^q}, \frac{1}{1+|y|^q}\right\}, \tag{3.31}$$

or equivalently

$$\sup_{(u,y)\in\mathbb{R}^{d_u}\times\mathbb{R}^{d_y}} p(u,y)\ \max\{1+|u|^q, 1+|y|^q\} < \infty.$$

To this end, note that

$$\begin{aligned} p(u,y)\max\{|u|^q,|y|^q\} &= \mathsf{P}\mu(u)\,\mathcal{N}\big(h(u),\Gamma\big)(y)\max\{|u|^q,|y|^q\} \\ &\leqslant \mathsf{P}\mu(u)\,\mathcal{N}\big(h(u),\Gamma\big)(y)|u|^q + \mathsf{P}\mu(u)\,\mathcal{N}\big(h(u),\Gamma\big)(y)|y|^q. \end{aligned}$$

The first term is bounded uniformly by **Step 1**, estimate (3.26). For the second term, we use that

$$|y|^q \leqslant 2^{q-1}\big|h(u)\big|^q + 2^{q-1}\big|y-h(u)\big|^q, \tag{3.32}$$

to obtain

$$\mathsf{P}\mu(u)\,\mathcal{N}\big(h(u),\Gamma\big)(y)|y|^q \leqslant C\mathsf{P}\mu(u)\,\mathcal{N}\big(h(u),\Gamma\big)(y)\Big(|h(u)|^q + |y-h(u)|^q\Big).$$

The first term on the right-hand side is bounded uniformly, again by **Step 1**, estimate (3.26), and using Assumption (H4). The second term is also bounded uniformly because the function $x \mapsto \mathcal{N}(0,\Gamma)(x)|x|^q$ is uniformly bounded in x, by a value depending only on Γ and q.

Step 5. Obtaining the estimate (3.24) The claimed inequality is equivalent to

$$\sup_{u_1,u_2,y} \frac{\big|p(u_1,y)-p(u_2,y)\big|}{|u_1-u_2|}\max\Big\{\min\left\{1+|u_1|^q, 1+|u_2|^q\right\}, 1+|y|^q\Big\} < \infty,$$

where the supremum is over $\mathbb{R}^{d_u}\times\mathbb{R}^{d_u}\times\mathbb{R}^{d_y}$. By (3.30), it holds that

$$\sup_{u_1,u_2,y} \frac{\big|p(u_1,y)-p(u_2,y)\big|}{|u_1-u_2|}\min\{1+|u_1|^q, 1+|u_2|^q\} < \infty,$$

so it remains to show that

$$\sup_{u_1,u_2,y} \frac{\big|p(u_1,y)-p(u_2,y)\big|}{|u_1-u_2|}|y|^q < \infty.$$

By (3.31), it is clear that the supremum is uniformly bounded if restricted to the set $|u_1 - u_2| \geqslant 1$, so it suffices to show that

$$\sup_{(u,\delta,y)\in\mathbb{R}^{d_u}\times B(0,1)\times\mathbb{R}^{d_y}} \frac{\big|p(u,y) - p(u+\delta,y)\big|}{|\delta|}|y|^q < \infty,$$

where $B(0,1)$ is the open ball of radius 1 centered radius at the origin in $\mathbb{R}^{d_u}$. We use again (3.32) in order to bound

$$\begin{aligned}\frac{\big|p(u,y) - p(u+\delta,y)\big|}{|\delta|}|y|^q \leqslant C\frac{\big|p(u,y) - p(u+\delta,y)\big|}{|\delta|}\big|h(u)\big|^q \\ + C\frac{\big|p(u,y) - p(u+\delta,y)\big|}{|\delta|}|y - h(u)|^q.\end{aligned}$$

The first term is bounded uniformly by (3.30) and the assumption that $\big|h(u)\big| \leqslant \kappa_h\big(1 + |u|\big)$. To conclude the proof, it remains to show that

$$\sup_{(u,\delta,y)\in\mathbb{R}^{d_u}\times B(0,1)\times\mathbb{R}^{d_y}} C\frac{\big|p(u,y) - p(u+\delta,y)\big|}{|\delta|}|y - h(u)|^q < \infty. \tag{3.33}$$

By (3.30) it holds that

$$\begin{aligned}&\big|p(u,y) - p(u+\delta,y)\big| \\ &\quad\leqslant \frac{C|\delta|}{1+|u|^q}\max\left\{\exp\left(-\frac{1}{4}|y - h(u)|_\Gamma^2\right), \exp\left(-\frac{1}{4}|y - h(u+\delta)|_\Gamma^2\right)\right\}. \\ &\quad\leqslant \frac{C|\delta|}{1+|u|^q}\exp\left(-\frac{1}{8}|y - h(u)|_\Gamma^2 + \frac{1}{4}\Big|h(u) - h(u+\delta)\Big|_\Gamma^2\right).\end{aligned}$$

In the last line we used the inequality $|a+b|^2 \geqslant \frac{1}{2}|a|^2 - |b|^2$, which follows from Young's inequality. Since the function $x \mapsto \mathrm{e}^{-\frac{x^2}{8}+C}x^q$ is bounded uniformly in $x \in \mathbb{R}$ and by Assumption (H5), the bound (3.33) easily follows, concluding the proof. □

3.B Technical Results for Theorem 3.2.2

We establish moment bounds in Subsection 3.B.1, we recall that on Gaussian measures the action of the conditioning map and the Kalman transport map are equivalent in Subsection 3.B.2, and we prove stability results in Subsection 3.B.3.

3.B.1 Moment Bounds

Lemma 3.B.1 (Moment Bounds). *Let μ be a probability measure on $\mathbb{R}^{d_u}$ with bounded first and second order polynomial moments $\mathcal{M}_1(\mu), \mathcal{M}_2(\mu) < \infty$. Under Assumptions (H2) and (H3), it holds that*

$$\big|\mathcal{M}(\mathsf{P}\mu)\big| \leqslant \kappa_\Psi\big(1 + \mathcal{M}_1(\mu)\big), \qquad \Sigma \preccurlyeq C(\mathsf{P}\mu) \preccurlyeq \Sigma + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)I_{d_u}. \tag{3.34}$$

Proof. The proof of this lemma follows the steps of the proof of J. A. Carrillo et al., 2024, Lemma B.1; however, here different bounds reflecting the assumptions on Ψ and h in this chapter are required. Using the definition of P in (3.4a), it holds that

$$\begin{aligned}\mathcal{M}(\mathsf{P}\mu) &= \int_{\mathbb{R}^{d_u}} u\, \mathsf{P}\mu(u)\, \mathrm{d}u \\ &= \frac{1}{\sqrt{(2\pi)^{d_u} \det \Sigma}} \int_{\mathbb{R}^{d_u}} \int_{\mathbb{R}^{d_u}} u \exp\left(-\frac{1}{2}\big|u - \Psi(v)\big|_\Sigma^2\right) \mu(\mathrm{d}v)\, \mathrm{d}u \\ &= \int_{\mathbb{R}^{d_u}} \Psi(v)\, \mu(\mathrm{d}v),\end{aligned}$$

where application of Fubini's theorem yields the last equality. The first inequality in (3.34) then follows from Assumption (H3). Obtaining the lower bound of the second inequality in (3.34) may be done identically to the proof of J. A. Carrillo et al., 2024, Lemma B.1. We turn our attention to obtaining the upper bound. Using Lemma 3.A.1 and noting that $ww^\mathsf{T} \preccurlyeq (w^\mathsf{T} w) I_{d_u}$ for any vector $w \in \mathbb{R}^{d_u}$ by the Cauchy-Schwarz inequality, we deduce that

$$\begin{aligned}C(\mathsf{P}\mu) &\preccurlyeq \int_{\mathbb{R}^{d_u}} u \otimes u\, \mathsf{P}\mu(u)\, \mathrm{d}u \\ &= \frac{1}{\sqrt{(2\pi)^{d_u} \det \Sigma}} \int_{\mathbb{R}^{d_u}} \int_{\mathbb{R}^{d_u}} u \otimes u \exp\left(-\frac{1}{2}|u - \Psi(v)|_\Sigma^2\right) \mu(\mathrm{d}v)\, \mathrm{d}u \\ &= \int_{\mathbb{R}^{d_u}} \big(\Psi(v) \otimes \Psi(v) + \Sigma\big)\, \mu(\mathrm{d}v) \\ &\preccurlyeq \Sigma + \left(\int_{\mathbb{R}^{d_u}} |\Psi(v)|^2 \mu(\mathrm{d}v)\right) I_{d_u} \preccurlyeq \Sigma + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big) I_{d_u}, \qquad (3.35)\end{aligned}$$

which yields the desired result. □

Lemma 3.B.2. *Let μ be a probability measure on $\mathbb{R}^{d_u}$ with bounded first and second order polynomial moments $\mathcal{M}_1(\mu), \mathcal{M}_2(\mu) < \infty$. Under Assumptions (H2) to (H4), it holds that*

$$\big|\mathcal{M}^u(\mathsf{QP}\mu)\big| \leqslant \kappa_\Psi\big(1 + \mathcal{M}_1(\mu)\big), \qquad (3.36a)$$

$$\big|\mathcal{M}^y(\mathsf{QP}\mu)\big| \leqslant \kappa_h \sqrt{2\Big(1 + \mathrm{tr}(\Sigma) + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)\Big)}. \qquad (3.36b)$$

Furthermore, it holds that

$$C(\mathsf{QP}\mu) \preccurlyeq \begin{pmatrix} 4\kappa_\Psi^2\big(1+\mathcal{M}_2(\mu)\big)I_{d_u} + 2\Sigma & 0_{d_u\times d_y} \\ 0_{d_y\times d_u} & 4\kappa_h^2\big(1+\operatorname{tr}(\Sigma)+2\kappa_\Psi^2\big(1+\mathcal{M}_2(\mu)\big)\big)I_{d_y} + \Gamma \end{pmatrix}, \tag{3.37a}$$

$$C(\mathsf{QP}\mu) \succcurlyeq \frac{\gamma\cdot\min\Big\{2\sigma, \gamma + 4\kappa_h^2\Big(1+\operatorname{tr}(\Sigma)+2\kappa_\Psi^2\big(1+\mathcal{M}_2(\mu)\big)\Big)\Big\}}{2\gamma + 8\kappa_h^2\Big(1+\operatorname{tr}(\Sigma)+2\kappa_\Psi^2\big(1+\mathcal{M}_2(\mu)\big)\Big)} I_{d_u+d_y}. \tag{3.37b}$$

Proof. The inequalities (3.36a) and (3.36b) follow from the assumptions and the fact that

$$\mathcal{M}(\mathsf{QP}\mu) = \begin{pmatrix} \mathcal{M}(\mathsf{P}\mu) \\ \mathsf{P}\mu[h] \end{pmatrix}. \tag{3.38}$$

Indeed, from Lemma 3.B.1 we know that $\big|\mathcal{M}(\mathsf{P}\mu)\big| \leqslant \kappa_\Psi\big(1+\mathcal{M}_1(\mu)\big)$, which leads by Jensen's inequality to (3.36a). To deduce (3.36b), we note that

$$\begin{aligned} \big|P\mu[h]\big|^2 &\leqslant \int_{\mathbb{R}^{d_u}} |h(u)|^2\, \mathsf{P}\mu(\mathrm{d}u) \\ &\leqslant 2\kappa_h^2 + 2\kappa_h^2 \int_{\mathbb{R}^{d_u}} |u|^2\, \mathsf{P}\mu(\mathrm{d}u) && \text{(3.39a)} \\ &= 2\kappa_h^2 + 2\kappa_h^2 \int_{\mathbb{R}^{d_u}} \Big(|\Psi(v)|^2 + \operatorname{tr}(\Sigma)\Big)\, \mu(\mathrm{d}v), && \text{(3.39b)} \end{aligned}$$

where (3.39a) follows by applying Assumption (H4) and Young's inequality, and (3.39b) follows by applying the properties of the Gaussian transition density. The result then follows by applying Assumption (H3).

To obtain the covariance bounds, we proceed using analogous steps to the proof of J. A. Carrillo et al., 2024, Lemma B.2; however, here different bounds reflecting the assumptions on Ψ and h in this chapter are required. We begin by establishing inequality (3.37a). To this end, letting $\phi\colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u+d_y}$ define the map $\phi(u) = \big(u, h(u)\big)$, it holds that

$$C(\mathsf{QP}\mu) = C(\phi_\sharp \mathsf{P}\mu) + \begin{pmatrix} 0_{d_u\times d_u} & 0_{d_u\times d_y} \\ 0_{d_y\times d_u} & \Gamma \end{pmatrix} = \begin{pmatrix} C^{uu}(\phi_\sharp\mathsf{P}\mu) & C^{uy}(\phi_\sharp\mathsf{P}\mu) \\ C^{yu}(\phi_\sharp\mathsf{P}\mu) & C^{yy}(\phi_\sharp\mathsf{P}\mu) + \Gamma \end{pmatrix}. \tag{3.40}$$

As shown in the proof of J. A. Carrillo et al., 2024, Lemma B.2 we have, for all $(a,b) \in \mathbb{R}^{d_u}\times\mathbb{R}^{d_y}$,

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} a \\ b \end{pmatrix} \preccurlyeq 2 \begin{pmatrix} aa^\mathsf{T} & 0_{d_u\times d_y} \\ 0_{d_y\times d_u} & bb^\mathsf{T} \end{pmatrix},$$

from which we establish, using Lemmas 3.A.1 and 3.B.1 and Assumptions (H2) to (H4), that

$$\begin{aligned} C(\phi_\sharp \mathsf{P}\mu) &\preccurlyeq \int_{\mathbb{R}^{d_u}} \begin{pmatrix} u \\ h(u) \end{pmatrix} \otimes \begin{pmatrix} u \\ h(u) \end{pmatrix} \mathsf{P}\mu(u)\,\mathrm{d}u \\ &\preccurlyeq 2 \int_{\mathbb{R}^{d_u}} \begin{pmatrix} uu^\mathsf{T} & 0_{d_u \times d_y} \\ 0_{d_y \times d_u} & h(u)h(u)^\mathsf{T} \end{pmatrix} \mathsf{P}\mu(u)\,\mathrm{d}u \\ &\preccurlyeq 2 \begin{pmatrix} \Sigma + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big) I_{d_u} & 0_{d_u \times d_y} \\ 0_{d_y \times d_u} & 2\kappa_h^2\big(1 + \mathrm{tr}(\Sigma) + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)\big) I_{d_y} \end{pmatrix}, \end{aligned} \tag{3.41}$$

where we applied (3.35) and the calculation resulting in (3.39a) in the last inequality. Noting this inequality in combination with (3.40) yields the upper bound (3.37a). We now show that

$$\begin{pmatrix} a \\ b \end{pmatrix}^\mathsf{T} C(\mathsf{QP}\mu) \begin{pmatrix} a \\ b \end{pmatrix} \geqslant (1-\varepsilon)\sigma|a|^2 + \left(\gamma - \left(\frac{1}{\varepsilon} - 1\right) \mathsf{P}\mu\big[|h|^2\big]\right) |b|^2 \tag{3.42}$$

in order to establish the lower bound (3.37b). The argument is similar to that in J. A. Carrillo et al., 2024, Lemma B.2 but is explicit in certain constants whose dependencies on moments need to be controlled in this chapter. We first note that for any $\pi \in \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ and all $(a, b) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_y}$ it holds that

$$\begin{aligned} \big|a^\mathsf{T} C^{uy}(\pi) b\big| &= \int_{\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}} \Big(a^\mathsf{T}\big(u - \mathcal{M}^u(\pi)\big)\Big)\Big(b^\mathsf{T}\big(y - \mathcal{M}^y(\pi)\big)\Big)\pi(\mathrm{d}u\mathrm{d}y) \\ &\leqslant \sqrt{a^\mathsf{T} C^{uu}(\pi) a}\, \sqrt{b^\mathsf{T} C^{yy}(\pi) b}, \end{aligned}$$

by the Cauchy–Schwarz inequality. Hence, we deduce that for all $\varepsilon \in (0, 1)$ and for all $(a, b) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_y}$

$$\begin{aligned} \begin{pmatrix} a \\ b \end{pmatrix}^\mathsf{T} C(\phi_\sharp \mathsf{P}\mu) \begin{pmatrix} a \\ b \end{pmatrix} &\geqslant (1-\varepsilon) a^\mathsf{T} C^{uu}(\phi_\sharp \mathsf{P}\mu) a - \left(\frac{1}{\varepsilon} - 1\right) b^\mathsf{T} C^{yy}(\phi_\sharp \mathsf{P}\mu) b \\ &\geqslant (1-\varepsilon) a^\mathsf{T} \Sigma a - \left(\frac{1}{\varepsilon} - 1\right) \mathsf{P}\mu\big[|h|^2\big], \end{aligned}$$

where we applied Young's inequality for the bound in the first line, and (3.34) and the bound $C^{yy}(\phi_\sharp \mathsf{P}\mu) \preccurlyeq \mathsf{P}\mu\big[|h|^2\big] I_{d_y}$ for the bound in the second line. We then apply (3.40) to find

$$\begin{aligned} \begin{pmatrix} a \\ b \end{pmatrix}^\mathsf{T} C(\mathsf{QP}\mu) \begin{pmatrix} a \\ b \end{pmatrix} &\geqslant (1-\varepsilon) a^\mathsf{T} \Sigma a - \left(\frac{1}{\varepsilon} - 1\right) \mathsf{P}\mu\big[|h|^2\big] + b^\mathsf{T} \Gamma b \\ &\geqslant (1-\varepsilon)\sigma|a|^2 + \left(\gamma - \left(\frac{1}{\varepsilon} - 1\right) \mathsf{P}\mu\big[|h|^2\big]\right) |b|^2. \end{aligned}$$

Choosing $\varepsilon = \frac{2\mathsf{P}\mu\left[|h|^2\right]}{\gamma+2\mathsf{P}\mu\left[|h|^2\right]}$, so that the coefficient of the $|b|^2$ term is $\frac{\gamma}{2}$, we obtain

$$\begin{pmatrix} a \\ b \end{pmatrix}^{\mathsf{T}} C(\mathsf{QP}\mu) \begin{pmatrix} a \\ b \end{pmatrix} \geqslant \frac{\gamma\sigma}{\gamma + 2\mathsf{P}\mu\left[|h|^2\right]}|a|^2 + \frac{\gamma}{2}|b|^2 \geqslant \min\left\{\frac{\gamma\sigma}{\gamma + 2\mathsf{P}\mu\left[|h|^2\right]}, \frac{\gamma}{2}\right\}\left(|a|^2 + |b|^2\right). \tag{3.43}$$

Since this is true for any $(a, b) \in \mathbb{R}^{d_u} \times \mathbb{R}^{d_y}$, it follows that

$$C(\mathsf{QP}\mu) \succcurlyeq \gamma \frac{\min\left\{2\sigma, \gamma + 2\mathsf{P}\mu\left[|h|^2\right]\right\}}{2\gamma + 4\mathsf{P}\mu\left[|h|^2\right]} I_{d_u+d_y}.$$

In order to deduce (3.37b), we use (3.39b) to find that

$$\mathsf{P}\mu\left[|h|^2\right] \leqslant 2\kappa_h^2\Big(1 + \mathrm{tr}(\Sigma) + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)\Big),$$

from which we conclude the desired result. □

Lemma 3.B.3 (Moment Bound for the True Filtering Distribution). *Let $q \geqslant 2$ be an integer. Assume that the probability measures $(\mu_j)_{j\in[\![0,J]\!]}$ are obtained from the dynamical system* (3.6) *initialized at the probability measure $\mu_0 \in \mathcal{P}(\mathbb{R}^{d_u})$ with bounded qth order polynomial moment $\mathcal{M}_q(\mu_0) < \infty$. If Assumptions (H1) to (H4) hold then there exists $C = C\big(\mathcal{M}_q(\mu_0), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\big)$ such that*

$$\max_{j\in[\![0,J]\!]} \mathcal{M}_q(\mu_j) \leqslant C.$$

Proof. We have $\mu_{j+1} = \mathsf{B}_j\mathsf{QP}\mu_j$, for $j \in [\![0, J-1]\!]$. Equivalently, using the notation from (3.3), it holds that $\mu_{j+1} = \mathsf{L}_j\mathsf{P}\mu_j$ for each $j \in [\![0, J-1]\!]$. Hence, we note that

$$\mu_{j+1} = \frac{\exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(u)\right|_\Gamma^2\right)\mathsf{P}\mu_j(u)}{\displaystyle\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(U)\right|_\Gamma^2\right)\mathsf{P}\mu_j(U)\,\mathrm{d}U}. \tag{3.44}$$

It is readily observed that

$$\begin{aligned}\mathcal{M}_q(\mu_{j+1}) &= \frac{\displaystyle\int_{\mathbb{R}^{d_u}} |u|^q \exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(u)\right|_\Gamma^2\right)\mathsf{P}\mu_j(u)\mathrm{d}u}{\displaystyle\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(U)\right|_\Gamma^2\right)\mathsf{P}\mu_j(U)\,\mathrm{d}U} \\ &\leqslant \frac{\displaystyle\int_{\mathbb{R}^{d_u}} |u|^q\,\mathsf{P}\mu_j(u)\mathrm{d}u}{\displaystyle\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y_{j+1}^\dagger - h(U)\right|_\Gamma^2\right)\,\mathsf{P}\mu_j(U)\,\mathrm{d}U}.\end{aligned} \tag{3.45}$$

We first bound from above the numerator of (3.45); indeed, note that

$$\begin{aligned}\int_{\mathbb{R}^{d_u}} |u|^q \mathsf{P}\mu_j(u)\mathrm{d}u &= \int_{\mathbb{R}^{d_u}} |u|^q \left(\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right) \mu_j(\mathrm{d}v)\right)\mathrm{d}u \\ &= \int_{\mathbb{R}^{d_u}} \left(\int_{\mathbb{R}^{d_u}} |u|^q \exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right) \mathrm{d}u\right)\mu_j(\mathrm{d}v) \quad (3.46a) \\ &\leqslant C\int_{\mathbb{R}^{d_u}} \big(1+|\Psi(v)|^q\big)\mu_j(\mathrm{d}v) \quad (3.46b) \\ &\leqslant C\Big(1+\mathcal{M}_q(\mu_j)\Big), \quad (3.46c)\end{aligned}$$

where (3.46a) follows from Fubini's theorem, the inequality (3.46b) from properties of Gaussians and Assumption (H2), and (3.46c) from application of Assumption (H3). We note that in (3.46c) the constant C depends on κ_Ψ, Σ. Now, to obtain a lower bound on the denominator of (3.45), we observe that

$$\begin{aligned}\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\big|y_{j+1}^\dagger - h(u)\big|_\Gamma^2\right)\mathsf{P}\mu_j(u)\,\mathrm{d}u &\geqslant \exp\left(-\frac{1}{2}\int_{\mathbb{R}^{d_u}} \big|y_{j+1}^\dagger - h(u)\big|_\Gamma^2 \mathsf{P}\mu_j(u)\,\mathrm{d}u\right) \\ &\geqslant C\exp\left(-\big|\Gamma^{-1}\big|\int_{\mathbb{R}^{d_u}} |h(u)|^2\mathsf{P}\mu_j(u)\,\mathrm{d}u\right) \\ &\geqslant C\exp\Big(-4\kappa_h^2\kappa_\Psi^2\big|\Gamma^{-1}\big|\mathcal{M}_2(\mu_j)\Big), \quad (3.47)\end{aligned}$$

where the first inequality follows by application of Jensen's inequality and (3.47) follows from the calculation leading to (3.39b). We note that the C in (3.47) is a constant depending on $\kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$. Therefore, by combining (3.47) with (3.46c), it is possible to deduce from (3.45) that

$$\mathcal{M}_q(\mu_{j+1}) \leqslant C\exp\Big(4\kappa_h^2\kappa_\Psi^2\big|\Gamma^{-1}\big|\mathcal{M}_2(\mu_j)\Big)\Big(1+\mathcal{M}_q(\mu_j)\Big), \quad (3.48)$$

where C is a constant depending on $\kappa_h, \kappa_\Psi, \kappa_y, \Sigma, \Gamma$. □

Remark 3.B.4.

- *In some situations, the bound* (3.48) *is overly pessimistic. For example, if h satisfies Assumption (H4) as well as the inequality $\big|h(u)-y_{j+1}^\dagger\big| \geqslant c_\ell\,(|u|-1)$ for all $u \in \mathbb{R}^{d_u}$ for some positive c_ℓ, then by Gerber, Hoffmann, and Urbain Vaes, 2023, Proposition A.3 for all $q > 0$ there is $C = C(c_\ell, q)$ such that*

$$\forall \mu \in \mathcal{P}(\mathbb{R}^{d_u}), \qquad \mathsf{L}_j\mu\Big[|u|^q\Big] \leqslant C\mu\Big[|u|^q\Big]. \quad (3.49)$$

  In this setting, better control of the moments can be achieved than in (3.48).

- *In the absence of any additional assumption on h beyond Assumption (H4), there may not exist a finite constant $C > 0$ such that the bound* (3.49) *holds*

for an arbitrary probability measure μ. To illustrate this, consider the case where $d_u = 2$, $d_y = 1$, $\Gamma = \frac{1}{4}$, $y_{j+1}^{\dagger} = 0$, and

$$h(u) = u_1, \qquad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \qquad \mu_R = \frac{x_R^2 \delta_{(R,0)} + \delta_{(0,x_R)}}{x_R^2 + 1}, \qquad x_R = \mathrm{e}^{\frac{R^2}{2}} .$$

Then, as $R \to \infty$, $\mu_R\left[|u|^2\right] \sim R^2$, while

$$\mathsf{L}_j \mu_R\left[|u|^2\right] = \frac{\int_{\mathbb{R}^2} (u_1^2 + u_2^2) \exp\left(-2u_1^2\right)\, \mu_R(\mathrm{d}u_1 \mathrm{d}u_2)}{\int_{\mathbb{R}^2} \exp\left(-2u_1^2\right)\, \mu_R(\mathrm{d}u_1 \mathrm{d}u_2)} = \frac{x_R^2\, \mathrm{e}^{-2R^2}\, R^2 + x_R^2}{x_R^2\, \mathrm{e}^{-2R^2} + 1} \sim x_R^2 = \mathrm{e}^{R^2} .$$

Thus, it holds that

$$\lim_{R \to \infty} \frac{\mathsf{L}_j \mu_R\left[|u|^2\right]}{\mu_R\left[|u|^2\right]} = +\infty,$$

which shows that (3.49) *fails to hold in this case.*

Lemma 3.B.5 (Moment Bound for the Approximate Filtering Distribution). *Let $q \geqslant 2$ be an integer. Assume that the probability measures $(\mu_j^{\mathrm{EK}})_{j \in [\![0,J]\!]}$ are obtained from the dynamical system* (3.10) *initialized at the probability measure $\mu_0^{\mathrm{EK}} \in \mathcal{P}(\mathbb{R}^{d_u})$ with bounded qth polynomial order moment $\mathcal{M}_q\left(\mu_0^{\mathrm{EK}}\right) < \infty$. If Assumptions (H1) to (H4) hold then there exists $C = C\left(\mathcal{M}_q\left(\mu_0^{\mathrm{EK}}\right), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\right)$ such that*

$$\max_{j \in [\![0,J]\!]} \mathcal{M}_q\left(\mu_j^{\mathrm{EK}}\right) \leqslant C.$$

Proof. We begin by noting that $\mu_{j+1}^{\mathrm{EK}} = \mathsf{T}_j \mathsf{Q}\mathsf{P} \mu_j^{\mathrm{EK}}$, for $j \in [\![0, J-1]\!]$. Thus

$$\begin{aligned}
\mathcal{M}_q\left(\mu_{j+1}^{\mathrm{EK}}\right) &= \int_{\mathbb{R}^{d_u}} |u|^q \mathsf{T}_j \mathsf{Q}\mathsf{P} \mu_j^{\mathrm{EK}}(\mathrm{d}u) \\
&= \int_{\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}} \left| \mathscr{T}(u, y; \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}, y_{j+1}^{\dagger}) \right|^q \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}(u, y)\, \mathrm{d}y\, \mathrm{d}u \\
&= \int_{\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}} \left| u + C^{uy}(\mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}) C^{yy}(\mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}})^{-1}\left(y_{j+1}^{\dagger} - y\right) \right|^q \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}(u, y)\, \mathrm{d}y\, \mathrm{d}u \\
&\leqslant C \int_{\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}} |u|^q\, \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}(u, y)\, \mathrm{d}y\, \mathrm{d}u \\
&\qquad + C\left(1 + \mathcal{M}_2\left(\mu_j^{\mathrm{EK}}\right)\right)^{2q} \cdot \left(1 + \int_{\mathbb{R}^{d_u} \times \mathbb{R}^{d_y}} |y|^q\, \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}(u, y)\, \mathrm{d}y\, \mathrm{d}u\right),
\end{aligned} \tag{3.50}$$

where in (3.50) the constant C depends on $\kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$ and where the dependence on the second moment of μ_j^{EK} in the second term is derived from (3.37a) and (3.37b).

By using the definitions of Q and P and by applying reasoning analogous to (3.46c), we deduce that

$$\begin{aligned}\mathcal{M}_q(\mu_{j+1}^{\mathrm{EK}}) &\leqslant C\Big(1+\mathcal{M}_2(\mu_j^{\mathrm{EK}})\Big)^{2q}\cdot\Big(1+\mathcal{M}_q(\mu_j^{\mathrm{EK}})\Big),\\ &\leqslant C\Big(1+\mathcal{M}_q(\mu_j^{\mathrm{EK}})\Big)^{2q+1},\end{aligned}$$

where C is a constant depending on $\kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$. Iteration gives the desired result. □

Lemma 3.B.6. *Let $\mu_1, \mu_2 \in \mathcal{P}(\mathbb{R}^n)$ have finite second moments, then the following bounds hold:*

$$\begin{aligned}\big|\mathcal{M}(\mu_1)-\mathcal{M}(\mu_2)\big| &\leqslant \frac{1}{2}d_g(\mu_1,\mu_2),\\ \big\|C(\mu_1)-C(\mu_2)\big\| &\leqslant \left(1+\frac{1}{2}|\mathcal{M}(\mu_1)+\mathcal{M}(\mu_2)|\right) d_g(\mu_1,\mu_2).\end{aligned}$$

Proof. The lemma as stated may be found in J. A. Carrillo et al., 2024, Lemma B.4, where a complete proof is given. □

3.B.2 Action of T_j on Gaussians

Lemma 3.B.7 ($\mathsf{B}_j\mathsf{G} = \mathsf{T}_j\mathsf{G}$). *Fix $y_{j+1}^\dagger \in \mathbb{R}^{d_y}$ and let π be a Gaussian measure over $\mathbb{R}^{d_u}\times\mathbb{R}^{d_y}$. Then the probability measure $\mathsf{B}_j\pi$, with B_j defined in* (3.5b) *is equivalent to the probability measure $\mathsf{T}_j\pi = \mathscr{T}(\bullet,\bullet;\pi,y_{j+1}^\dagger)_\sharp\pi$, where $\mathscr{T}$ is defined in* (3.9b).

Proof. The lemma as stated may be found in J. A. Carrillo et al., 2024, Lemma B.5, where a complete proof is given. □

3.B.3 Stability Results

Lemma 3.B.8 (Map P is Lischitz). *Suppose that Σ and Ψ satisfy Assumptions (H2) and (H3), respectively. Then it holds that*

$$\forall(\mu,\pi)\in\mathcal{P}(\mathbb{R}^{d_u})\times\mathcal{P}(\mathbb{R}^{d_u}),\qquad d_g(\mathsf{P}\mu,\mathsf{P}\pi)\leqslant\Big(1+2\kappa_\Psi^2+\mathrm{tr}(\Sigma)\Big)\,d_g(\mu,\pi).$$

Proof. By definition of P, it holds that

$$\mathsf{P}\mu(\mathrm{d}u)=\int_{\mathbb{R}^{d_u}}p(v,\mathrm{d}u)\,\mu(\mathrm{d}v),\qquad\text{where } p(v,\mathrm{d}u):=\frac{\exp\left(-\frac{1}{2}|u-\Psi(v)|_\Sigma^2\right)}{\sqrt{(2\pi)^{d_u}\det\Sigma}}\,\mathrm{d}u.$$

Let $g(u) = 1 + |u|^2$, as in Definition 1.2.1, and take any function $f\colon \mathbb{R}^{d_u} \to \mathbb{R}$ such that $|f| \leqslant g$. Assumption (H3) implies that

$$\begin{aligned}\left|\int_{\mathbb{R}^{d_u}} f(u)\, p(v, \mathrm{d}u)\right| &\leqslant \int_{\mathbb{R}^{d_u}} g(u)\, p(v, \mathrm{d}u) \\ &= 1 + |\Psi(v)|^2 + \mathrm{tr}(\Sigma) \\ &\leqslant 1 + 2\kappa_\Psi^2 + 2\kappa_\Psi^2 |v|^2 + \mathrm{tr}(\Sigma) \leqslant \Big(1 + 2\kappa_\Psi^2 + \mathrm{tr}(\Sigma)\Big) g(v).\end{aligned}$$

By Fubini's theorem, it follows that

$$\begin{aligned}\Big|\mathsf{P}\mu[f] - \mathsf{P}\pi[f]\Big| &= \left|\int_{\mathbb{R}^{d_u}} \left(\int_{\mathbb{R}^{d_u}} f(u)\, p(v, \mathrm{d}u)\right) \big(\mu(\mathrm{d}v) - \pi(\mathrm{d}v)\big)\right| \\ &= \Big(1 + 2\kappa_\Psi^2 + \mathrm{tr}(\Sigma)\Big)\Big|\mu[g] - \pi[g]\Big| \leqslant \Big(1 + 2\kappa_\Psi^2 + \mathrm{tr}(\Sigma)\Big)\, d_g(\mu, \pi),\end{aligned}$$

thus concluding the proof. □

Lemma 3.B.9 (Map Q is Lipschitz). *Suppose that Γ and h satisfy Assumptions (H2) and (H4), respectively. Then it holds that*

$$\forall (\mu, \pi) \in \mathcal{P}(\mathbb{R}^{d_u}) \times \mathcal{P}(\mathbb{R}^{d_u}), \qquad d_g(\mathsf{Q}\mu, \mathsf{Q}\pi) \leqslant \Big(1 + 2\kappa_h^2 + \mathrm{tr}(\Gamma)\Big) d_g(\mu, \pi). \tag{3.51}$$

Proof. Take $f\colon \mathbb{R}^{d_u} \times \mathbb{R}^{d_y} \to \mathbb{R}$ satisfying $|f| \leqslant g$, where $g(u, v) = 1 + |u|^2 + |y|^2$. By Fubini's theorem, it holds that

$$\mathsf{Q}\mu[f] - \mathsf{Q}\pi[f] = \int_{\mathbb{R}^{d_u}} \Pi f(u) \Big(\mu(\mathrm{d}u) - \pi(\mathrm{d}u)\Big),$$

where

$$\Pi f(u) := \int_{\mathbb{R}^{d_y}} f(u, y)\, \mathsf{N}\big(h(u), \Gamma\big)(\mathrm{d}y).$$

The function $\Pi f\colon \mathbb{R}^{d_u} \to \mathbb{R}$ satisfies

$$\begin{aligned}\forall u \in \mathbb{R}^{d_u}, \qquad |\Pi f(u)| &\leqslant \int_{\mathbb{R}^{d_y}} \big|f(u, y)\big|\, \mathsf{N}\big(h(u), \Gamma\big)(\mathrm{d}y) \\ &\leqslant \int_{\mathbb{R}^{d_y}} \Big(1 + |u|^2 + |y|^2\Big)\, \mathsf{N}\big(h(u), \Gamma\big)(\mathrm{d}y) \\ &= 1 + |u|^2 + \big|h(u)\big|^2 + \mathrm{tr}(\Gamma) \leqslant \Big(1 + 2\kappa_h^2 + \mathrm{tr}(\Gamma)\Big)\Big(1 + |u|^2\Big).\end{aligned}$$

Therefore, we deduce that

$$\Big|\mathsf{Q}\mu[f] - \mathsf{Q}\pi[f]\Big| \leqslant \Big(1 + 2\kappa_h^2 + \mathrm{tr}(\Gamma)\Big)\, d_g(\mu, \pi),$$

which concludes the proof. □

Lemma 3.B.10 (Stability of Map B_j). *Suppose that Assumptions (H1) to (H4) are satisfied. Then for any probability measure $\mu \in \mathcal{P}(\mathbb{R}^{d_u})$ with $\mathcal{M}_2(\mu) < \infty$ there exists a constant $C_{\mathsf{B}} = C_{\mathsf{B}}\big(\mathcal{M}_2(\mu), \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\big) > 0$ such that*

$$\forall j \in [\![0, J]\!], \qquad d_g(\mathsf{B}_j \mathsf{GQP}\mu, \mathsf{B}_j \mathsf{QP}\mu) \leqslant C_{\mathsf{B}} d_g(\mathsf{GQP}\mu, \mathsf{QP}\mu).$$

Proof. The proof of this lemma follows the steps of the proof of J. A. Carrillo et al., 2024, Lemma B.9; however, each step requires different bounds reflecting the assumptions on Ψ and h in this chapter. For ease of exposition we write $y^\dagger = y^\dagger_{j+1}$. We define the y-marginal densities

$$\alpha_\mu(y) := \int_{\mathbb{R}^{d_u}} \mathsf{GQP}\mu(u, y)\,\mathrm{d}u\,, \qquad \beta_\mu(y) := \int_{\mathbb{R}^{d_u}} \mathsf{QP}\mu(u, y)\,\mathrm{d}u\,.$$

By definition of B_j, it holds that

$$\begin{aligned} d_g(\mathsf{B}_j \mathsf{GQP}\mu, \mathsf{B}_j \mathsf{QP}\mu) &= \int_{\mathbb{R}^{d_u}} \Big(1 + |u|^2\Big) \left| \frac{\mathsf{GQP}\mu(u, y^\dagger)}{\alpha_\mu(y^\dagger)} - \frac{\mathsf{QP}\mu(u, y^\dagger)}{\beta_\mu(y^\dagger)} \right| \mathrm{d}u \\ &\leqslant \frac{1}{\alpha_\mu(y^\dagger)} \int_{\mathbb{R}^{d_u}} \Big(1 + |u|^2\Big) \big|\mathsf{GQP}\mu(u, y^\dagger) - \mathsf{QP}\mu(u, y^\dagger)\big| \,\mathrm{d}u \\ &\quad + \left| \frac{\alpha_\mu(y^\dagger) - \beta_\mu(y^\dagger)}{\alpha_\mu(y^\dagger)\beta_\mu(y^\dagger)} \right| \int_{\mathbb{R}^{d_u}} \Big(1 + |u|^2\Big) \mathsf{QP}\mu(u, y^\dagger)\,\mathrm{d}u. \end{aligned} \tag{3.52}$$

Step 1: bounding $\alpha_\mu(y^\dagger)$ and $\beta_\mu(y^\dagger)$ from below The distribution $\alpha_\mu(\bullet)$ is Gaussian with mean $\mathcal{M}^y(\mathsf{QP}\mu)$ and covariance

$$C^{yy}(\mathsf{QP}\mu) = \Gamma + \mathsf{P}\mu\big[h \otimes h\big] - \mathsf{P}\mu\big[h\big] \otimes \mathsf{P}\mu\big[h\big]. \tag{3.53}$$

Clearly $C^{yy}(\mathsf{QP}\mu) \succcurlyeq \Gamma$. Furthermore, it holds that $C^{yy}(\mathsf{QP}\mu) \preccurlyeq \Gamma + 2\kappa_h^2\big(1 + \mathrm{tr}(\Sigma) + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)\big) I_{d_y}$ by (3.41). Therefore, noting that (3.36b) implies that $|\mathcal{M}^y(\mathsf{QP}\mu)| \leqslant C$ for a constant C depending only on the parameters $\mathcal{M}_2(\mu), \kappa_h, \kappa_\Psi, \Sigma$, we obtain

$$\begin{aligned} \alpha_\mu(y) &= \frac{1}{\sqrt{(2\pi)^{d_u} \det\big(C^{yy}(\mathsf{QP}\mu)\big)}} \exp\left(-\frac{1}{2}\big(y - \mathcal{M}^y(\mathsf{QP}\mu)\big)^{\mathsf{T}} C^{yy}(\mathsf{QP}\mu)^{-1}\big(y - \mathcal{M}^y(\mathsf{QP}\mu)\big)\right) \\ &\geqslant \frac{\exp\Big(-\frac{1}{2}\big(|y| + C\big)^2 \big|\Gamma^{-1}\big|\Big)}{\sqrt{(2\pi)^{d_y} \det\big(\Gamma + 2\kappa_h^2\big(1 + \mathrm{tr}(\Sigma) + 2\kappa_\Psi^2\big(1 + \mathcal{M}_2(\mu)\big)\big) I_{d_y}\big)}}. \end{aligned} \tag{3.54}$$

The function β_μ can be bounded from below using similar reasoning. Indeed, by Assumptions (H2) to (H4) we have that for all $y \in \mathbb{R}^{d_y}$,

$$\begin{aligned}\beta_\mu(y) &= \int_{\mathbb{R}^{d_u}} \mathsf{QP}\mu(u,y)\,\mathrm{d}u = \int_{\mathbb{R}^{d_u}} \frac{\exp\left(-\frac{1}{2}\big(y-h(u)\big)^{\mathsf{T}}\Gamma^{-1}\big(y-h(u)\big)\right)}{\sqrt{(2\pi)^{d_y}\det(\Gamma)}}\mathsf{P}\mu(u)\,\mathrm{d}u \\ &\geqslant \frac{1}{\sqrt{(2\pi)^{d_y}\det(\Gamma)}}\exp\Big(-\big|\Gamma^{-1}\big||y|^2 - \big|\Gamma^{-1}\big|\kappa_h^2\int_{\mathbb{R}^{d_u}}\big(2+|u|^2\big)\mathsf{P}\mu(u)\,\mathrm{d}u\Big) \qquad (3.55a) \\ &\geqslant C\exp\Big(-\big|\Gamma^{-1}\big||y|^2\Big), \qquad (3.55b)\end{aligned}$$

where we applied Jensen's inequality in (3.55a) and the constant in (3.55b) depends on $\mathcal{M}_2(\mu), \kappa_\Psi, \kappa_h, \Sigma, \Gamma$.

Step 2: bounding the first term in (3.52) First note that, for fixed $u \in \mathbb{R}^{d_u}$, the functions $\mathsf{QP}\mu(u,\bullet)$ and $\mathsf{GQP}\mu(u,\bullet)$ are Gaussians up to constant factors, with covariance matrices given respectively by Γ and

$$C^{yy}(\mathsf{QP}\mu) - C^{yu}(\mathsf{QP}\mu)C^{uu}(\mathsf{QP}\mu)^{-1}C^{uy}(\mathsf{QP}\mu), \qquad (3.56)$$

where we have used the formula for the covariance of the conditional distribution of a Gaussian. We note that $C^{yu}(\mathsf{QP}\mu)C^{uu}(\mathsf{QP}\mu)^{-1}C^{uy}(\mathsf{QP}\mu)$ is positive semi-definite, hence by (3.53) and its upper bound it follows that the matrix (3.56) is bounded from above by $2\kappa_h^2\big(1+\mathrm{tr}(\Sigma)+2\kappa_\Psi^2\big(1+\mathcal{M}_2(\mu)\big)\big)I_{d_y}+\Gamma$. As shown in the proof of J. A. Carrillo et al., 2024, Lemma B.9, the matrix (3.56) is bounded from below by Γ (see J. A. Carrillo et al., 2024, Equation (B.20)). It follows from using Lemma 3.A.2 in Section 3.A, with parameter $\tau = \tau\big(\mathcal{M}_2(\mu), \kappa_h, \kappa_\Psi, \Sigma, \Gamma\big)$, that

$$\int_{\mathbb{R}^{d_u}}\Big(1+|u|^2\Big)\,|\mathsf{GQP}\mu(u,y) - \mathsf{QP}\mu(u,y)|\,\mathrm{d}u \leqslant C d_g(\mathsf{GQP}\mu, \mathsf{QP}\mu), \qquad (3.57)$$

where C is a constant depending on $\mathcal{M}_2(\mu), \kappa_h, \kappa_\Psi, \Sigma, \Gamma$. We refer to J. A. Carrillo et al., 2024, Equation (B.21) for the detailed steps used to establish this bound.

Step 3: bounding the second term in (3.52) In view of (3.57), it holds that

$$\big|\alpha_\mu(y) - \beta_\mu(y)\big| \leqslant \int_{\mathbb{R}^{d_u}}\Big(1+|u|^2\Big)\,\big|\mathsf{GQP}\mu(u,y) - \mathsf{QP}\mu(u,y)\big|\,\mathrm{d}u \leqslant C d_g(\mathsf{GQP}\mu, \mathsf{QP}\mu).$$

Now, since $\mathsf{QP}\mu(u,\bullet)/\mathsf{P}\mu(u)$ defines a Gaussian density which has covariance Γ and is bounded uniformly from above by $\big((2\pi)^{d_y}\det(\Gamma)\big)^{-1/2}$, we also have that

$$\int_{\mathbb{R}^{d_u}}\Big(1+|u|^2\Big)\,\mathsf{QP}\mu(u,y)\,\mathrm{d}u \leqslant \int_{\mathbb{R}^{d_u}}\frac{\big(1+|u|^2\big)\,\mathsf{P}\mu(u)}{\sqrt{(2\pi)^{d_y}\det(\Gamma)}}\,\mathrm{d}u = \frac{1+\mathrm{tr}\big(C(\mathsf{P}\mu)\big)+|\mathcal{M}(\mathsf{P}\mu)|^2}{\sqrt{(2\pi)^{d_y}\det(\Gamma)}}.$$

By the moment bounds in Lemma 3.B.1, the right-hand side is bounded from above by a constant which depends on $\mathcal{M}_2(\mu), \kappa_\Psi, \Sigma$.

Step 4: concluding the proof Putting together the above bounds, we conclude that

$$d_g(\mathsf{B}_j\mathsf{GQP}\mu, \mathsf{B}_j\mathsf{QP}\mu) \leqslant \frac{C\big(\mathcal{M}_2(\mu), \kappa_\Psi, \kappa_h, \Sigma, \Gamma\big)}{\alpha_\mu(y^\dagger_{j+1})}\left(1 + \frac{1}{\beta_\mu(y^\dagger_{j+1})}\right) d_g(\mathsf{GQP}\mu, \mathsf{QP}\mu).$$

Applying the inequalities (3.54) and (3.55b) yields the desired result. □

Lemma 3.B.11 (Stability of Map T_j). *Suppose that Assumption H1 is satisfied. Then, for all $R \geqslant 1$, it holds that for any $\pi \in \mathcal{P}_R(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ and $p \in \{\mathsf{QP}\mu : \mu \in \mathcal{P}(\mathbb{R}^{d_u})$ and $\mathcal{M}_{2\cdot\max\{3+d_u,4+d_y\}}(\mu) < \infty\} \subset \mathcal{P}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$, there is $L_\mathsf{T} := L_\mathsf{T}\big(R, \mathcal{M}_{2\cdot\max\{3+d_u,4+d_y\}}(\mu), \kappa_y, \kappa_\Psi, \kappa_h, \ell_h, \Sigma, \Gamma\big)$, such that*

$$\forall j \in [\![1, J]\!], \qquad d_g(\mathsf{T}_j\pi, \mathsf{T}_j p) \leqslant L_\mathsf{T}\, d_g(\pi, p).$$

Proof. By the results of Lemma 3.B.2, it holds that $\mathsf{QP}\mu \in \mathcal{P}_{\widetilde{R}}(\mathbb{R}^{d_u} \times \mathbb{R}^{d_y})$ for some $\widetilde{R}(\mathcal{M}_2(\mu), \kappa_\Psi, \kappa_h, \Sigma, \Gamma) \geqslant 1$. Using analogous notation to the one found in the proof of J. A. Carrillo et al., 2024, Lemma B.10 we define

$$r = \max\Big\{R, \widetilde{R}, \kappa_y\Big\}.$$

Letting $K = C(\pi)$, $S = C(p)$ and $y^\dagger = y^\dagger_{j+1}$, we also define the affine maps $\mathscr{T}^\pi$ and $\mathscr{T}^p$ corresponding to use of covariance information at the probability measures π and $p = \mathsf{QP}\mu$,

$$\begin{aligned}
\mathscr{T}^\pi(u, y) &= u + A_\pi(y^\dagger - y), \qquad A_\pi := K_{uy}K_{yy}^{-1},\\
\mathscr{T}^p(u, y) &= u + A_p(y^\dagger - y), \qquad A_p := S_{uy}S_{yy}^{-1}.
\end{aligned}$$

By a straightforward application of the triangle inequality, it holds that

$$d_g(\mathsf{T}_j\pi, \mathsf{T}_j p) \leqslant d_g(\mathscr{T}^\pi_\sharp\pi, \mathscr{T}^\pi_\sharp p) + d_g(\mathscr{T}^\pi_\sharp p, \mathscr{T}^p_\sharp p). \tag{3.58}$$

In the following steps we will separately bound the two terms on the right hand side of (3.58). Before proceeding we outline two auxiliary bounds that will be used in the rest of the proof. These bounds are identical to the ones found in the proof of J. A. Carrillo et al., 2024, Lemma B.10; we include statements here for expository purposes. Noting that the operator 2-norm of any submatrix is bounded from above

by the operator 2-norm of the full matrix, we observe (see J. A. Carrillo et al., 2024, Equation (B.23)) that

$$|A_\pi| \leqslant |K_{uy}||K_{yy}^{-1}| \leqslant |K||K^{-1}| \leqslant r^4. \tag{3.59}$$

Note that the above bound similarly holds for A_p. Using (3.59) and assuming without loss of generality that $r > 1$, we deduce (see J. A. Carrillo et al., 2024, Equation (B.24)) that

$$|A_\pi - A_p| \leqslant |(K_{uy} - S_{uy})K_{yy}^{-1}| + |S_{uy}\left(K_{yy}^{-1} - S_{yy}^{-1}\right)| \leqslant 2r^6|K - S| \leqslant 2r^6(1+2r)\, d_g(\pi, p), \tag{3.60}$$

where the second inequality follows again from fact that the 2-norm of any submatrix is bounded from above by the 2-norm of the full matrix, while the last inequality follows from the result in Lemma 3.B.6.

Bounding the first term in (3.58) With an identical argument to the one used in the proof of J. A. Carrillo et al., 2024, Lemma B.10 we have that

$$d_g(\mathscr{T}^\pi_\sharp \pi, \mathscr{T}^\pi_\sharp p) \leqslant 3\left(1 + r^{10}\right) r^8\, d_g(\pi, p). \tag{3.61}$$

Bounding the second term in (3.58) Let f again satisfy $|f| \leqslant g$. We note that

$$\left|\mathscr{T}^\pi_\sharp p[f] - \mathscr{T}^p_\sharp p[f]\right| = \left|p[f \circ \mathscr{T}^\pi] - p[f \circ \mathscr{T}^p]\right| = \left|p[f \circ \mathscr{T}^\pi - f \circ \mathscr{T}^p]\right|.$$

The last term may be expressed as

$$\begin{aligned}
&\left|p[f \circ \mathscr{T}^\pi - f \circ \mathscr{T}^p]\right| \\
&\qquad = \left|\int_{\mathbb{R}^{d_y}} \int_{\mathbb{R}^{d_u}} \left(f\left(u + A_\pi\left(y^\dagger - y\right)\right) - f\left(u + A_p\left(y^\dagger - y\right)\right)\right) p(u, y)\, \mathrm{d}u\, \mathrm{d}y\right| \\
&\qquad = \int_{\mathbb{R}^{d_y}} \int_{\mathbb{R}^{d_u}} \left(f(u + A_\pi z) - f(u + A_p z)\right) p(u, y^\dagger - z)\, \mathrm{d}u\, \mathrm{d}z \\
&\qquad = \int_{\mathbb{R}^{d_y}} \int_{\mathbb{R}^{d_u}} f(v)\Big(p(v - A_\pi z, y^\dagger - z) - p(v - A_p z, y^\dagger - z)\Big)\, \mathrm{d}v\, \mathrm{d}z,
\end{aligned} \tag{3.62}$$

where we used a change of variables in the second equality. Since $\mathcal{M}_{2\cdot\max\{3+d_u, 4+d_y\}}(\mu) < \infty$ by assumption, it follows from Lemma 3.A.3 and from the inequality $\min\{a, b\} \leqslant$

$\sqrt{a}\sqrt{b}$ that there is a constant C so that

$$\begin{aligned}
&\Big|p(v - A_\pi z, y^\dagger - z) - p(v - A_p z, y^\dagger - z)\Big| \\
&\quad \leqslant C|A_\pi z - A_p z| \cdot \max\left\{\frac{1}{1 + |v - A_\pi z|^{\max\{3+d_u, 4+d_y\}}}, \frac{1}{1 + |v - A_p z|^{\max\{3+d_u, 4+d_y\}}}\right\} \\
&\qquad \times \frac{1}{1 + |y^\dagger - z|^{\max\{3+d_u, 4+d_y\}}}.
\end{aligned}$$

We apply this inequality to bound for fixed $z \in \mathbb{R}^{d_y}$ the inner integral in (3.62). Considering only the terms that depend on v gives

$$\begin{aligned}
&\int_{\mathbb{R}^{d_u}} |f(v)| \max\left\{\frac{1}{1 + |v - A_\pi z|^{\max\{3+d_u, 4+d_y\}}}, \frac{1}{1 + |v - A_p z|^{\max\{3+d_u, 4+d_y\}}}\right\} \mathrm{d}v \qquad (3.63) \\
&\quad \leqslant \int_{\mathbb{R}^{d_u}} \frac{|f(v)|}{1 + |v - A_\pi z|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}v + \int_{\mathbb{R}^{d_u}} \frac{|f(v)|}{1 + |v - A_p z|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}v \\
&\quad \leqslant \int_{\mathbb{R}^{d_u}} \frac{1 + |v|^2}{1 + |v - A_\pi z|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}v + \int_{\mathbb{R}^{d_u}} \frac{1 + |v|^2}{1 + |v - A_p z|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}v \\
&\quad \leqslant \int_{\mathbb{R}^{d_u}} \frac{1 + |w + A_\pi z|^2}{1 + |w|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}w + \int_{\mathbb{R}^{d_u}} \frac{1 + |w + A_p z|^2}{1 + |w|^{\max\{3+d_u, 4+d_y\}}} \mathrm{d}w,
\end{aligned}$$

where we used that $|f(v)| \leqslant 1 + |v|^2$, as well as a change of variable in the last line. It follows that the integral in (3.63) is bounded from above by

$$C\Big(1 + |A_\pi z|^2 + |A_p z|^2\Big) \leqslant Cr^8 \left(1 + |z|^2\right),$$

where the inequality follows from (3.59). Finally, the resulting integral in the z variable can be bounded analogously, which gives

$$\begin{aligned}
d_g(\mathscr{T}^\pi_\sharp p, \mathscr{T}^p_\sharp p) &\leqslant Cr^8 \int_{\mathbb{R}^{d_y}} \frac{1 + |z|^2}{1 + |y^\dagger - z|^{\max\{3+d_u, 4+d_y\}}} |A_\pi z - A_p z| \, \mathrm{d}z \\
&\leqslant Cr^{11} |A_\pi - A_p| \leqslant Cr^{18} d_g(\pi, p), \qquad (3.64)
\end{aligned}$$

where the last inequality follows from (3.60). Combining (3.61) and (3.64) yields the desired result. □

3.C Technical Results for Approximation Result in Proposition 3.2.3

In Lemma 3.C.1 we recall the local Lipschitz continuity result for the operator G established in J. A. Carrillo et al. (2024). Lemma 3.C.3 establishes that the filtering distribution is a locally Lipschitz function of (Ψ, h), viewed as a mapping from

Banach space equipped with the $\|\cdot\|_\infty$ norm into the space of probability measures metrized using the d_g distance. This is preceded by Lemma 3.C.2 which establishes bounds used to prove this Lipschitz property. The two lemmas do not require Ψ_0 and h_0 to be affine, but simply require that they both satisfy Assumptions (H3) and (H4). This is in contrast with the more specific setting of Proposition 3.2.3, which imposes an affine assumption on (Ψ_0, h_0).

Lemma 3.C.1. *For all $R \geqslant 1$, there exists $L_{\mathsf{G}} = L_{\mathsf{G}}(R, n)$ so that for any $\mu_1, \mu_2 \in \mathcal{P}_R(\mathbb{R}^n)$ it holds that*

$$d_g(\mathsf{G}\mu_1, \mathsf{G}\mu_2) \leqslant L_{\mathsf{G}}(R, n) \cdot d_g(\mu_1, \mu_2).$$

Proof. The lemma as stated may be found in J. A. Carrillo et al., 2024, Lemma B.12, where a complete proof is given. □

Lemma 3.C.2. *Suppose that the matrices (Σ, Γ) satisfy Assumption (H2). Fix $\kappa_\Psi, \kappa_h > 0$ and assume that $\Psi_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ and $h_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ are functions satisfying Assumptions (H3) and (H4). Then the following statements hold:*

- *There is a constant $C_p = C_p(\kappa_\Psi, \Sigma)$ such that for all $\varepsilon \in [0, 1]$ and all $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$,*

$$\forall \mu \in \mathcal{P}(\mathbb{R}^{d_u}), \qquad d_g(\mathsf{P}_0\mu, \mathsf{P}\mu) \leqslant C_p\varepsilon \cdot \big(1 + \mathcal{M}_2(\mu)\big); \tag{3.65}$$

- *There is a constant $C_q = C_q(\kappa_h, \Gamma)$ such that for all $\varepsilon \in [0, 1]$ and all $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$,*

$$\forall \mu \in \mathcal{P}(\mathbb{R}^{d_u}), \qquad d_g(\mathsf{Q}_0\mu, \mathsf{Q}\mu) \leqslant C_q\varepsilon \cdot \big(1 + \mathcal{M}_2(\mu)\big); \tag{3.66}$$

- *There is $C_b = C_b(\kappa_h, \Gamma)$ such that for all $y^\dagger \in \mathbb{R}^{d_y}$, all $\varepsilon \in [0, 1]$, all $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$, and all probability measures $(\mu_0, \mu) \in \mathcal{P}(\mathbb{R}^{d_u}) \times \mathcal{P}(\mathbb{R}^{d_u})$,*

$$d_g\Big(\mathsf{B}(\mathsf{Q}_0\mu_0; y^\dagger), \mathsf{B}(\mathsf{Q}\mu, y^\dagger)\Big) \leqslant \exp(C_b\mathcal{R})\Big(\varepsilon + d_g(\mu_0, \mu)\Big), \tag{3.67}$$

where $\mathcal{R} \in [1, \infty]$ is given by

$$\mathcal{R} = \max\left\{\left|y^\dagger\right|^2, 1 + \mathcal{M}_2(\mu_0), 1 + \mathcal{M}_2(\mu)\right\}.$$

Here P_0 and Q_0 denote the maps associated to (Ψ_0, h_0), and P and Q are the maps associated to (Ψ, h).

Proof. Throughout this proof, C_p denotes a constant depending only on (κ_Ψ, Σ), and C_q is a constant that depends only on (κ_h, Γ). Both may change from line to line.

Proof of (3.65) It holds that

$$\mathsf{P}_0\mu(u) - \mathsf{P}\mu(u) = C_p \int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|u - \Psi_0(v)\right|_\Sigma^2\right) - \exp\left(-\frac{1}{2}\left|u - \Psi(v)\right|_\Sigma^2\right) \mu(\mathrm{d}v).$$

By the elementary inequality (3.27), the integrand on the right-hand side is bounded in absolute value by

$$2\left|\Psi_0(v) - \Psi(v)\right| \left(\exp\left(-\frac{1}{4}\left|u - \Psi_0(v)\right|_\Sigma^2\right) + \exp\left(-\frac{1}{4}\left|u - \Psi(v)\right|_\Sigma^2\right)\right).$$

By Young's inequality, it holds that $|a+b|^2 \geqslant \frac{1}{2}|a|^2 - |b|^2$, and so this is bounded by

$$4\left|\Psi_0(v) - \Psi(v)\right| \left(\exp\left(-\frac{1}{8}\left|u - \Psi_0(v)\right|_\Sigma^2 + \frac{1}{4}\left|\Psi_0(v) - \Psi(v)\right|_\Sigma^2\right)\right) \leqslant C_p\varepsilon \exp\left(-\frac{1}{8}\left|u - \Psi_0(v)\right|_\Sigma^2\right).$$

It follows, by Fubini's theorem, that

$$\begin{aligned} d_g(\mathsf{P}_0\mu, \mathsf{P}\mu) &\leqslant C_p\varepsilon \int_{\mathbb{R}^{d_u}} \int_{\mathbb{R}^{d_u}} \left(1 + |u|^2\right) \exp\left(-\frac{1}{8}\left|u - \Psi_0(v)\right|_\Sigma^2\right) \mathrm{d}u\, \mu(\mathrm{d}v) \\ &\leqslant C_p\varepsilon \int_{\mathbb{R}^{d_u}} \left(1 + \left|\Psi_0(v)\right|^2\right) \mu(\mathrm{d}v) \leqslant C_p\varepsilon\, (1 + \kappa_\Psi^2) \int_{\mathbb{R}^{d_u}} \left(1 + |v|^2\right) \mu(\mathrm{d}v). \end{aligned}$$

This concludes the proof of (3.65).

Proof of (3.66) Recall that

$$d_g(\mathsf{Q}_0\mu, \mathsf{Q}\mu) = C_q \int_{\mathbb{R}^{d_u}} \int_{\mathbb{R}^{d_y}} g(u,y) \left(\exp\left(-\frac{1}{2}\left|y - h_0(u)\right|_\Gamma^2\right) - \exp\left(-\frac{1}{2}\left|y - h(u)\right|_\Gamma^2\right)\right) \mathrm{d}y\, \mu(\mathrm{d}u),$$

where $g(u,y) = 1 + |u|^2 + |v|^2$. Using the same reasoning as above we obtain that

$$\left|\exp\left(-\frac{1}{2}\left|y - h_0(u)\right|_\Gamma^2\right) - \exp\left(-\frac{1}{2}\left|y - h(u)\right|_\Gamma^2\right)\right| \leqslant C_q\varepsilon \exp\left(-\frac{1}{8}\left|y - h_0(u)\right|_\Gamma^2\right). \tag{3.68}$$

Therefore, we deduce that

$$d_g(\mathsf{Q}_0\mu, \mathsf{Q}\mu) \leqslant C_q\varepsilon \int_{\mathbb{R}^{d_u}} \left(1 + \left|h_0(u)\right|^2\right) \mu(\mathrm{d}u) \leqslant C_q\varepsilon(1 + \kappa_h^2) \int_{\mathbb{R}^{d_u}} \left(1 + |u|^2\right) \mu(\mathrm{d}u),$$

which proves (3.66).

Proof of (3.67) We assume for simplicity that μ_0 and μ have densities, but note that this is not required. Let $\pi_0 = \mathsf{B}(\mathsf{Q}_0\mu_0; y^\dagger)$ and $\pi = \mathsf{B}(\mathsf{Q}\mu, y^\dagger)$ and recall that

$$\pi_0(u) = \frac{\exp\left(-\frac{1}{2}\left|y^\dagger - h_0(u)\right|_\Gamma^2\right) \mu_0(u)}{\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y^\dagger - h_0(U)\right|_\Gamma^2\right) \mu_0(U)\, \mathrm{d}U} =: \frac{f_0(u)}{\int_{\mathbb{R}^{d_u}} f_0(U)\, \mathrm{d}U} =: \frac{f_0(u)}{Z_0}$$

and similarly

$$\pi(u) = \frac{\exp\left(-\frac{1}{2}\left|y^\dagger - h(u)\right|_\Gamma^2\right)\mu(u)}{\int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y^\dagger - h(U)\right|_\Gamma^2\right)\mu(U)\,\mathrm{d}U} =: \frac{f(u)}{\int_{\mathbb{R}^{d_u}} f(U)\,\mathrm{d}U} =: \frac{f_0(u)}{Z}.$$

Note that

$$\begin{aligned} d_g(\pi_0,\pi) &= \int_{\mathbb{R}^{d_u}} \left(1+|u|^2\right)\left|\frac{f_0(u)}{Z_0} - \frac{f(u)}{Z}\right|\mathrm{d}u \\ &= \frac{1}{Z}\int_{\mathbb{R}^{d_u}} \left(1+|u|^2\right)\left|f_0(u) - f(u)\right|\mathrm{d}u + \left|\frac{1}{Z_0} - \frac{1}{Z}\right|\int_{\mathbb{R}^{d_u}} \left(1+|u|^2\right) f_0(u)\,\mathrm{d}u. \end{aligned}$$

In order to bound the first term, we write

$$\begin{aligned} f_0(u) - f(u) = &\left(\exp\left(-\frac{1}{2}\left|y^\dagger - h_0(u)\right|_\Gamma^2\right) - \exp\left(-\frac{1}{2}\left|y^\dagger - h(u)\right|_\Gamma^2\right)\right)\mu_0(u) \\ &+ \exp\left(-\frac{1}{2}\left|y^\dagger - h(u)\right|_\Gamma^2\right)\left(\mu_0(u) - \mu(u)\right). \end{aligned} \tag{3.69}$$

Using (3.68), we obtain that

$$\left|f_0(u) - f(u)\right| \leqslant C_q\varepsilon \exp\left(-\frac{1}{8}\left|y^\dagger - h_0(u)\right|_\Gamma^2\right)\mu_0(u) + \exp\left(-\frac{1}{2}\left|y^\dagger - h(u)\right|_\Gamma^2\right)\left|\mu_0(u) - \mu(u)\right|, \tag{3.70}$$

and so

$$\int_{\mathbb{R}^{d_u}} \left(1+|u|^2\right)\left|f_0(u) - f(u)\right|\mathrm{d}u \leqslant C_q\varepsilon\mathcal{R} + d_g(\mu_0,\mu).$$

Therefore it holds that

$$d_g(\pi_0,\pi) \leqslant \mathcal{R}\left(\frac{C_q\varepsilon}{Z} + \frac{|Z_0 - Z|}{Z_0 Z}\right) + \frac{1}{Z}d_g(\mu_0,\mu).$$

By (3.70) it holds that

$$|Z_0 - Z| \leqslant \int_{\mathbb{R}^{d_u}} |f_0(U) - f(U)|\,\mathrm{d}U \leqslant C_q\varepsilon + d_g(\mu_0,\mu),$$

and so we obtain finally

$$d_g(\pi_0,\pi) \leqslant \mathcal{R}\left(\frac{1}{Z} + \frac{1}{Z_0 Z}\right)\left(C_q\varepsilon + d_g(\mu_0,\mu)\right).$$

Furthermore Z_0 is bounded from below because, by Jensen's inequality,

$$\begin{aligned} Z_0 &= \int_{\mathbb{R}^{d_u}} \exp\left(-\frac{1}{2}\left|y^\dagger - h_0(U)\right|_\Gamma^2\right)\mu_0(U)\,\mathrm{d}U \geqslant \exp\left(-\frac{1}{2}\int_{\mathbb{R}^{d_u}} \left|y^\dagger - h_0(U)\right|_\Gamma^2 \mu_0(U)\,\mathrm{d}U\right) \\ &\geqslant \exp\left(-\left|y^\dagger\right|_\Gamma^2 - C_q\kappa_h^2\int_{\mathbb{R}^{d_u}} \left(1+|u|^2\right)\mu_0(U)\,\mathrm{d}U\right) = \exp\left(-C_q\mathcal{R}\right). \end{aligned}$$

The same bound holds for Z, and so we obtain the result. □

Lemma 3.C.3. *Suppose that the data $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ and the matrices (Σ, Γ) satisfy Assumptions (H1) and (H2). Fix $\kappa_\Psi, \kappa_h > 0$ and assume that $\Psi_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ and $h_0 \colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ are functions satisfying Assumptions (H3) and (H4), respectively. Let $(\mu_j^0)_{j\in[\![1,J]\!]}$ and $(\mu_j)_{j\in[\![1,J]\!]}$ denote the true filtering distributions associated with functions (Ψ_0, h_0) and (Ψ, h), respectively, initialized at the same Gaussian probability measure $\mu_0 = \mathsf{N}(m_0, C_0) \in \mathcal{G}(\mathbb{R}^{d_u})$. For all $J \in \mathbb{Z}^+$ there is $C = C(m_0, C_0, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, J) > 0$ such that for all $\varepsilon \in [0,1]$ and all $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$, it holds that*

$$\max_{j\in[\![0,J]\!]} d_g(\mu_j^0, \mu_j) \leqslant C\varepsilon. \tag{3.71}$$

Proof. By Lemma 3.B.3, the filtering distributions have bounded second moments. Let

$$\mathcal{R} = \max_{j\in[\![0,J-1]\!]} \left(\left|y_{j+1}^\dagger\right|^2,\ 1 + \mathcal{M}_2(\mu_j^0),\ 1 + \mathcal{M}_2(\mu_j) \right).$$

Throughout this proof, C denotes a constant whose value is irrelevant in the context, depends only on the constants $m_0, C_0, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma, k$ (but neither on ε, nor on Ψ and h) and may change from line to line.

The statement is obviously true for $J = 0$. Reasoning by induction, we assume that the statement is true up to $J = k$ and show that there is $C > 0$ such that

$$\forall \varepsilon \in [0,1], \qquad \forall (\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big), \qquad d_g(\mu_{k+1}^0, \mu_{k+1}) \leqslant C\varepsilon.$$

To this end, let P_0 and Q_0 denote the maps associated to (Ψ_0, h_0), fix $\varepsilon \in [0,1]$, and fix $(\Psi, h) \in B_{L^\infty}\big((\Psi_0, h_0), \varepsilon\big)$. Using (3.67), then the triangle inequality, and finally (3.65) and Lemma 3.B.8, we have that

$$\begin{aligned}
d_g(\mu_{k+1}^0, \mu_{k+1}) &= d_g\big(\mathsf{B}_k\mathsf{Q}_0\mathsf{P}_0\mu_k^0, \mathsf{B}_k\mathsf{Q}\mathsf{P}\mu_k\big) \\
&\leqslant \mathrm{e}^{C\mathcal{R}}\Big(\varepsilon + d_g\big(\mathsf{P}_0\mu_k^0, \mathsf{P}\mu_k\big)\Big) \\
&\leqslant \mathrm{e}^{C\mathcal{R}}\Big(\varepsilon + d_g\big(\mathsf{P}_0\mu_k^0, \mathsf{P}\mu_k^0\big) + d_g\big(\mathsf{P}\mu_k^0, \mathsf{P}\mu_k\big)\Big) \\
&\leqslant \mathrm{e}^{C\mathcal{R}}\Big(\varepsilon + C\varepsilon\mathcal{R} + C d_g(\mu_k^0, \mu_k)\Big).
\end{aligned}$$

Using the induction hypothesis gives us the desired bound. □

3.D Technical Result for Theorem 3.2.5

In Le Gland, Monbet, and Tran (2011b) machinery is established to prove Monte Carlo error estimates between the finite particle ensemble Kalman filter and its

mean field limit. We use such results as a component in proving Theorem 3.2.5 and, in so-doing, explicit dependence on moments must be tracked. In this section we give a self-contained proof of Le Gland, Monbet, and Tran, 2011b, Theorem 5.2, following the analysis closely[2] and, in addition, tracking dependence on moments; this moment dependence may be useful in the context of future work generalizing what we do in this chapter. This leads to the following error estimate stating the desired Monte Carlo error estimate.

Lemma 3.D.1. *Assume that the probability measures* $(\mu_j^{\mathrm{EK}})_{j\in[\![0,J]\!]}$ *and* $(\mu_j^{\mathrm{EK},N})_{j\in[\![0,J]\!]}$ *are obtained, respectively, from the dynamical systems* (3.10) *and* (3.12)*, initialized at the Gaussian probability measure* $\mu_0^{\mathrm{EK}} \in \mathcal{G}(\mathbb{R}^{d_u})$ *and at the empirical measure* $\mu_0^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^N \delta_{u_0^{(i)}}$ *for* $u_0^{(i)} \sim \mu_0^{\mathrm{EK}}$ *i.i.d. samples. That is,*

$$\mu_{j+1}^{\mathrm{EK}} = \mathsf{T}_j \mathsf{Q}\mathsf{P}\mu_j^{\mathrm{EK}}, \qquad \mu_{j+1}^{\mathrm{EK},N} = \frac{1}{N}\sum_{i=1}^N \delta_{u_{j+1}^{(i)}},$$

where $u_{j+1}^{(i)}$ *evolve according to the iteration in* (3.11)*. Suppose that the data* $Y^\dagger = \{y_j^\dagger\}_{j=1}^J$ *and the matrices* (Σ, Γ) *satisfy Assumptions (H1) and (H2). We assume that the vector field* Ψ *satisfies Assumption (H3) and that* h *is a linear transformation, i.e. that Assumption (V2) is satisfied and hence also Assumption (H4). Furthermore, if the vector field* Ψ *additionally satisfies* $|\Psi|_{C^{0,1}} \leqslant \ell_\Psi < \infty$*, then for all* ϕ *satisfying Assumption P1, there exists a constant* $C = C\big(\mathcal{M}_q(\mu_0^{\mathrm{EK}}), J, R_\phi, L_\phi, \varsigma, \kappa_y, \kappa_\Psi, \kappa_h, \ell_\Psi, \Sigma, \Gamma\big)$ *where* $q := \max\{4^{J+1}, 4\varsigma, 2(\varsigma+1)\}$ *such that*

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}} \leqslant \frac{C}{\sqrt{N}}.$$

Proof. To prove the proposition, we apply the coupling argument used in Le Gland, Monbet, and Tran (2011b). Using similar notation to the one in Le Gland, Monbet, and Tran (2011b), to the interacting N-particle system $\{u_j^{(i)}\}_{n=1}^N$ evolving according to the ensemble Kalman dynamics (3.11), we couple N copies of the mean field dynamics $\{\overline{u}_j^{(j)}\}_{n=1}^N$ evolving according to the mean field ensemble Kalman dynamics (3.7). The mean field replicas are synchronously coupled to the interacting particle system, in the sense that they are initialized at the same initial condition and driven by the same noises; namely, the two particle systems are initialized at

[2]The result of Le Gland, Monbet, and Tran (2011b) holds more generally for functions Ψ that are locally Lipschitz and that grow at most polynomially at infinity Le Gland, Monbet, and Tran, 2011b, Assumption B. However, our proof uses that Ψ is globally Lipschitz, for instance in (3.82).

i.i.d. samples $u_0^{(i)} = \overline{u}_0^{(i)} \sim \mu_0^{\mathrm{EK}}$ for $i = 1, \dots, N$. For simplicity of notation, the forecast particles are denoted by the letter v, and we drop the hat notation from the forecast and simulated observations. Furthermore, we add a bar $\overline{\bullet}$ to all the variables related to the synchronously coupled mean field particles, including the probability measures $\overline{\pi}_j^{\mathrm{EK}}$. We also define for $\pi \in \mathcal{P}(\mathbb{R}^{d_u \times d_y})$ the Kalman gain

$$\mathcal{K}(\pi) := C^{uh}(\pi)\left(C^{hh}(\pi) + \Gamma\right)^{-1}.$$

With this notation, the interacting particle system, and synchronously coupled system read as follows:

Interacting particle system	*Synchronous coupling*
Initialization: $u_0^{(i)} = \overline{u}_0^{(i)}$	**Initialization:** $\overline{u}_0^{(i)} = u_0^{(i)}$
$v_{j+1}^{(i)} = \Psi\left(u_j^{(i)}\right) + \xi_j^{(i)},$	$\overline{v}_{j+1}^{(i)} = \Psi\left(\overline{u}_j^{(i)}\right) + \xi_j^{(i)},$
$y_{j+1}^{(i)} = h\left(v_{j+1}^{(i)}\right) + \eta_{j+1}^{(i)},$	$\overline{y}_{j+1}^{(i)} = h\left(\overline{v}_{j+1}^{(i)}\right) + \eta_{j+1}^{(i)},$
$u_{j+1}^{(i)} = v_{j+1}^{(i)} + \mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right)\left(y_{j+1}^{\dagger} - y_{j+1}^{(i)}\right).$	$\overline{u}_{j+1}^{(i)} = \overline{v}_{j+1}^{(i)} + \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\left(y_{j+1}^{\dagger} - \overline{y}_{j+1}^{(i)}\right).$

Note that the synchronously coupled particles are independent and identically distributed. With this set-up we proceed with the proof. Let ϕ satisfy Assumption P1. By applying the triangle inequality, we deduce that

$$\left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}} \leqslant \left(\mathbb{E}\left|\mu_J^{\mathrm{EK},N}[\phi] - \frac{1}{N}\sum_{i=1}^{N}\phi\left(\overline{u}_J^{(i)}\right)\right|^2\right)^{\frac{1}{2}} + \left(\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N}\phi\left(\overline{u}_J^{(i)}\right) - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}}. \tag{3.74}$$

Step 1: bounding the second term in (3.74) Noting that the random variables $\phi\left(\overline{u}_J^{(i)}\right) - \mu_J^{\mathrm{EK}}[\phi]$ are i.i.d. and expanding the square, we obtain

$$\left(\mathbb{E}\left|\sum_{i=1}^{N}\frac{1}{N}\left(\phi\left(\overline{u}_J^{(i)}\right) - \mu_J^{\mathrm{EK}}[\phi]\right)\right|^2\right)^{\frac{1}{2}} = \frac{1}{\sqrt{N}}\left(\mathbb{E}\left|\phi\left(\overline{u}_J^{(1)}\right) - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}}.$$

Since by Assumption P1 it holds that $|\phi(u)| \leqslant R_\phi\left(1 + |u|^{\varsigma+1}\right)$ for any $u \in \mathbb{R}^{d_u}$, it follows that

$$\left(\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N}\phi\left(\overline{u}_J^{(i)}\right) - \mu_J^{\mathrm{EK}}[\phi]\right|^2\right)^{\frac{1}{2}} \leqslant \frac{C\left(\mathcal{M}_{2(\varsigma+1)}\left(\mu_0^{\mathrm{EK}}\right), R_\phi, J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\right)}{\sqrt{N}}, \tag{3.75}$$

where we used Lemma 3.B.5 to bound $\mathcal{M}_{2(\varsigma+1)}\big(\mu_J^{\mathrm{EK}}\big)$ in terms of $\mathcal{M}_{2(\varsigma+1)}\big(\mu_0^{\mathrm{EK}}\big)$.

Step 2: bounding the first term in (3.74) For the first term, by Jensen's inequality, exchangeability, and finally by Assumption P1 on ϕ together with the Cauchy–Schwarz inequality, it holds without loss of generality that

$$\begin{aligned}\left(\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N}\phi\Big(u_J^{(i)}\Big)-\phi\Big(\overline{u}_J^{(i)}\Big)\right|^2\right)^{\frac{1}{2}} &\leqslant \left(\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left|\phi\Big(u_J^{(i)}\Big)-\phi\Big(\overline{u}_J^{(i)}\Big)\right|^2\right)^{\frac{1}{2}}\\ &= \left(\mathbb{E}\left|\phi\Big(u_J^{(1)}\Big)-\phi\Big(\overline{u}_J^{(1)}\Big)\right|^2\right)^{\frac{1}{2}}\\ &\leqslant 3L_\phi\left(\mathbb{E}\left|u_J^{(1)}-\overline{u}_J^{(1)}\right|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\left[1+\left|u_J^{(1)}\right|^{4\varsigma}+\left|\overline{u}_J^{(1)}\right|^{4\varsigma}\right]\right)^{\frac{1}{4}}.\end{aligned} \tag{3.76}$$

By Lemma 3.B.5, there exists $C = C\Big(\mathcal{M}_{4\varsigma}\big(\mu_0^{\mathrm{EK}}\big), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\Big)$ so that

$$\mathbb{E}\left|\overline{u}_J^{(1)}\right|^{4\varsigma} \leqslant C,$$

and

$$\mathbb{E}\left|u_J^{(1)}\right|^{4\varsigma} \leqslant 2^{4\varsigma-1}\left(\mathbb{E}\left|\overline{u}_J^{(1)}\right|^{4\varsigma}+\mathbb{E}\left|u_J^{(1)}-\overline{u}_J^{(1)}\right|^{4\varsigma}\right) \leqslant 2^{4\varsigma-1}\left(C+\mathbb{E}\left|u_J^{(1)}-\overline{u}_J^{(1)}\right|^{4\varsigma}\right).$$

Hence it remains to bound terms of the form $\mathbb{E}\big|u_J^{(1)}-\overline{u}_J^{(1)}\big|^p$ with $p=4$ and $p=4\varsigma$. Such a bound will follow from a *propagation of chaos* result, which will be the object of Step 4. In Step 3 we show auxiliary moment bounds for the mean field dynamics.

Step 3: moment bounds for the mean field dynamics By Lemma 3.B.5, for any $j \in [\![0, J-1]\!]$ there exists a constant $C = C\Big(\mathcal{M}_p\big(\mu_0^{\mathrm{EK}}\big), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma\Big)$ so that

$$\left(\mathbb{E}\left|\overline{u}_j^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant C. \tag{3.77}$$

From this it immediately follows, by Assumptions (H3) and (H4), that there exist constants C_v, C_y depending on $\mathcal{M}_p\big(\mu_0^{\mathrm{EK}}\big), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$ so that

$$\left(\mathbb{E}\left|\overline{v}_{j+1}^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant C_v, \qquad \left(\mathbb{E}\left|\overline{y}_{j+1}^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant C_y. \tag{3.78}$$

Furthermore, it holds that for the Kalman gain that

$$\left|\mathcal{K}\big(\overline{\pi}_{j+1}^{\mathrm{EK}}\big)\right|^p \leqslant \left|C^{uh}\big(\overline{\pi}_{j+1}^{\mathrm{EK}}\big)\right|^p\left|\Gamma^{-1}\right|^p = \left|\Gamma^{-1}\right|^p\left|C^{uu}\big(\overline{\pi}_{j+1}^{\mathrm{EK}}\big)H^{\mathsf{T}}\right|^p \leqslant C, \tag{3.79}$$

where C depends on $\mathcal{M}_2\big(\mu_0^{\mathrm{EK}}\big), J, \kappa_y, \kappa_\Psi, \kappa_h, \Sigma, \Gamma$.

Step 4: propagation of chaos In this step, we prove a propagation of chaos result stating for all p, there exists a constant $C_p = C_p\left(\mathcal{M}_{4^J\cdot p}(\mu_0^{\mathrm{EK}}), J, \kappa_y, \kappa_\Psi, \kappa_h, \ell_\Psi, \ell_h, \Sigma, \Gamma\right)$ such that

$$\sup_{N\in\mathbb{N}} \sqrt{N}\left(\mathbb{E}\left|u_J^{(1)} - \overline{u}_J^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant C_p. \tag{3.80}$$

To prove this result, we follow the strategy in Le Gland, Monbet, and Tran (2011b) and reason by induction. Fix $j \in [\![0, J-1]\!]$ and assume that for all $p \in \mathbb{N}$ there is $C_{j,p} = C_{j,p}\left(\mathcal{M}_{4^j\cdot p}(\mu_0^{\mathrm{EK}}), J, \kappa_y, \kappa_\Psi, \kappa_h, \ell_\Psi, \ell_h, \Sigma, \Gamma\right)$ such that

$$\sup_{N\in\mathbb{N}} \sqrt{N}\left(\mathbb{E}\left|u_j^{(1)} - \overline{u}_j^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant C_{j,p}. \tag{3.81}$$

By Lipschitz continuity of Ψ and h, we deduce immediately that

$$\sup_{N\in\mathbb{N}} \sqrt{N}\left(\mathbb{E}\left|v_{j+1}^{(1)} - \overline{v}_{j+1}^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant \ell_\Psi C_{j,p}, \qquad \sup_{N\in\mathbb{N}} \sqrt{N}\left(\mathbb{E}\left|y_{j+1}^{(1)} - \overline{y}_{j+1}^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant \ell_h\ell_\Psi C_{j,p}. \tag{3.82}$$

Now we write

$$\begin{aligned} u_{j+1}^{(1)} - \overline{u}_{j+1}^{(1)} = &\left(v_{j+1}^{(1)} - \overline{v}_{j+1}^{(1)}\right) + \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\left(\overline{y}_{j+1}^{(1)} - y_{j+1}^{(1)}\right) \\ &+ \left(\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right)\left(y_{j+1}^{\dagger} - y_{j+1}^{(1)}\right). \end{aligned}$$

Thus, by the triangle and Hölder inequalities,

$$\begin{aligned} \sqrt{N}\left(\mathbb{E}\left|u_{j+1}^{(1)} - \overline{u}_{j+1}^{(1)}\right|^p\right)^{\frac{1}{p}} \leqslant &\ \ell_\Psi C_{j,p} + \left|\mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right| \ell_h\ell_\Psi C_{j,p} \\ &+ \sqrt{N}\left(\mathbb{E}\left|\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|^{2p}\right)^{\frac{1}{2p}}\left(\mathbb{E}\left|y_{j+1}^{\dagger} - y_{j+1}^{(1)}\right|^{2p}\right)^{\frac{1}{2p}}. \end{aligned}$$

The first two terms on the right-hand side are bounded uniformly in N in view of (3.79). In order to bound the last term, we note that the term $\mathbb{E}\left|y_{j+1}^{\dagger} - y_{j+1}^{(1)}\right|^{2p}$ is bounded uniformly in N by a constant which depends on $\mathcal{M}_{4^j\cdot 2p}(\mu_0^{\mathrm{EK}})$; this follows by (3.78) and (3.82). To complete the inductive step, it remains to show that

$$\sup_{N\in\mathbb{N}} \sqrt{N}\left(\mathbb{E}\left|\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|^{2p}\right)^{\frac{1}{2p}} < \infty.$$

To this end, in line with the classical propagation of chaos approach, we decompose

$$\begin{aligned} \sqrt{N}\left(\mathbb{E}\left|\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|^{2p}\right)^{\frac{1}{2p}} \leqslant &\sqrt{N}\left(\mathbb{E}\left|\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right|^{2p}\right)^{\frac{1}{2p}} \\ &+ \sqrt{N}\left(\mathbb{E}\left|\mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|^{2p}\right)^{\frac{1}{2p}}, \end{aligned} \tag{3.83}$$

where we introduced the empirical measure associated with the mean field particles:

$$\overline{\pi}_{j+1}^{\mathrm{EK},N} := \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\overline{v}_{j+1}^{(i)}, \overline{y}_{j+1}^{(i)}\right)}.$$

To conclude the proof of propagation of chaos, we must prove that

- The first term on the right-hand side of (3.83) is bounded uniformly in N, which is achieved by employing the induction hypothesis (3.81), as well as (3.82).
- The second term on the right-hand side of (3.83) is also bounded uniformly in N. This follows from classical arguments based on the law of large number in L^p spaces.

Let us first bound the second term. To this end, note that

$$\begin{aligned}\mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right) = C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\left(\left(\Gamma + C^{hh}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right)^{-1} - \left(\Gamma + C^{hh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right)^{-1}\right) \\ + \left(C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right)\left(\Gamma + C^{hh}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right)^{-1}.\end{aligned}$$

Therefore, using that $A^{-1} - B^{-1} = B^{-1}(B-A)A^{-1}$, we have that

$$\begin{aligned}\left|\mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right| &\leqslant \left|C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right| |\Gamma^{-1}|^2 \left|C^{hh}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - C^{hh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right| \\ &\quad + |\Gamma^{-1}|\left|C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right) - C^{uh}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right| \\ &\leqslant C(\Gamma, H)\left(1 + \left|C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|\right)\left|C^{uu}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|.\end{aligned}$$

The term $\left|C^{uu}(\overline{\pi}_{j+1}^{\mathrm{EK}})\right|$ is bounded by (3.78), and the L^p norm of the second term tends to zero with rate $N^{-\frac{p}{2}}$ by classical law of large number arguments, see for example Ding and Q. Li, 2021c, Lemma 3 and U. Vaes, 2024, Lemma 3; indeed it holds that

$$\mathbb{E}\left|C^{uu}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK}}\right)\right|^{2p} \leqslant \frac{C\left(\mathcal{M}_{4p}(\mu_0^{\mathrm{EK}})\right)}{N^p}.$$

It remains to bound the other term. Reasoning similarly to above, we have that

$$\left|\mathcal{K}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - \mathcal{K}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right| \leqslant C(\Gamma, H)\left(1 + \left|C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right|\right)\left|C^{uu}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right|. \tag{3.84}$$

By U. Vaes, 2024, Lemma 2, it holds that

$$\left|C^{uu}\left(\pi_{j+1}^{\mathrm{EK},N}\right) - C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right|^2 \leqslant 2\left(\frac{1}{N}\sum_{i=1}^{N}\left|v_{j+1}^{(i)} - \overline{v}_{j+1}^{(i)}\right|^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}\left|v_{j+1}^{(i)}\right|^2 + \frac{1}{N}\sum_{i=1}^{N}\left|\overline{v}_{j+1}^{(i)}\right|^2\right).$$

Using the Cauchy-Schwarz and Jensen's inequalities, moment bounds and finally the induction hypothesis (3.80) enables to conclude that

$$\left(\mathbb{E}\left|C^{uu}\left(\pi_{j+1}^{\mathrm{EK},N}\right)-C^{uu}\left(\overline{\pi}_{j+1}^{\mathrm{EK},N}\right)\right|^{2p}\right)^{\frac{1}{2p}} \leqslant \frac{C\left(\mathcal{M}_{4^{j+1}p}\left(\mu_0^{\mathrm{EK}}\right), J, \kappa_y, \kappa_\Psi, \kappa_h, \ell_\Psi, \ell_h, \Sigma, \Gamma\right)}{\sqrt{N}}.$$

Step 5: concluding the proof Using the propagation of chaos result from (3.80) in (3.76) gives that

$$\left(\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N}\phi\left(u_J^{(i)}\right)-\phi\left(\overline{u}_J^{(i)}\right)\right|^2\right)^{\frac{1}{2}} \leqslant \frac{C}{\sqrt{N}},$$

for some constant C depending on $\mathcal{M}_q\left(\mu_0^{\mathrm{EK}}\right), J, L_\phi, \varsigma, \kappa_y, \kappa_\Psi, \kappa_h, \ell_\Psi, \ell_h, \Sigma, \Gamma$ with $q = \max\{4^{J+1}, 4\varsigma\}$. Combining this result with (3.75) yields the desired result. □

Part II

Operator Learning for Nonlinear Data Assimilation

Chapter 4

TRANSFORMER NEURAL OPERATORS

4.1 Introduction

Attention was introduced in the context of neural machine translation in Bahdanau, Cho, and Bengio (2014) to effectively model correlations in sequential data. Many attention mechanisms used in conjunction with recurrent networks were subsequently developed, for example long short-term memory (Hochreiter and Schmidhuber, 1997) and gated recurrent units (Chung et al., 2014), which were then employed for a wide range of applications, see for example the review Chung et al. (2014). The seminal work Vaswani et al. (2017) introduced the now ubiquitous transformer architecture, which relies solely on the attention mechanism and not on the recurrence of the network in order to effectively model nonlocal, long-range correlations. In its full generality, the attention function described in Vaswani et al. (2017) defines an operation between sequences $\{u_i\}_{i=1}^N \subset \mathbb{R}^{d_u}$, $\{v_i\}_{i=1}^M \subset \mathbb{R}^{d_v}$ of lengths $N, M \in \mathbb{N}$ respectively, represented as tensors $u \in \mathbb{R}^{N\times d_u}$, $v \in \mathbb{R}^{M\times d_v}$, where learnable weights $Q \in \mathbb{R}^{d_K\times d_u}, K \in \mathbb{R}^{d_K\times d_v}, V \in \mathbb{R}^{d_V\times d_v}$ parametrize

$$\texttt{attention}(u, v) := \texttt{softmax}(uQ^T Kv^T)vV^T. \tag{4.1}$$

In this context, we recall that the function $\texttt{softmax} : \mathbb{R}^{N\times M} \to \mathbb{R}^{N\times M}$ is defined component-wise via its action on the input as

$$\big(\texttt{softmax}(w)\big)_{ij} = \frac{\exp(w_{ij})}{\sum_{j=1}^M \exp(w_{ij})}, \tag{4.2}$$

for any $w \in \mathbb{R}^{N\times M}$. Therefore, in this form the attention function defines a mapping $\texttt{attention} : \mathbb{R}^{N\times d_u} \times \mathbb{R}^{M\times d_v} \to \mathbb{R}^{N\times d_V}$.[1] Alternatively, we can view the output of the attention function as a sequence $\{\texttt{attention}(u, v)_i\}_{i=1}^N \subset \mathbb{R}^{d_V}$.

Viewed more generally, attention defines a mapping from two sequences $\{u_i\}_{i=1}^N$ and $\{v_i\}_{i=1}^M$ into a third sequence $\{\texttt{attention}(u, v)_i\}_{i=1}^N$; all three sequences take values in $\mathbb{R}^r$ for various values of r. In this chapter, we generalize this in two ways: (a) we replace index i by index $x \in D$, where D is a bounded open subset in $\mathbb{R}^d$; (b) we retain a discrete index i but allow the sequences to take values in Banach spaces

[1] We note that in most implementations, the input of the softmax function is rescaled by a factor of $1/\sqrt{d_K}$. In our context, continuum attention, a different scaling will be used.

of $\mathbb{R}^r$-valued functions defined over D, where D is a bounded open subset in $\mathbb{R}^d$. We find the first generalization (a) most useful in settings where $d = 1$, that is for time series. However, when discretizing the index set $D \subset \mathbb{R}^d$ with $N := n^d$ points, the quadratic scaling of the attention mechanism in N can be prohibitive. This motivates generalization (b) where we retain the discrete index but allow sequences to be function valued, corresponding to patching of functions defined over $D \subset \mathbb{R}^d$, with $d \geqslant 2$.

Our motivation for these generalizations is operator learning and, in particular, design of discretization invariant neural operator architectures. Such architectures disentangle model parameters from any particular discretization of the input functional data; learning is focused on the intrinsic continuum problem and not any particular discretization of it. This allows models to obtain consistent errors when trained with different resolution data, as well as enabling training, testing, and inference at different resolutions. We consider the general setting where $D \subset \mathbb{R}^d$ is a bounded open set and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$ and $\mathcal{Z} = \mathcal{Z}(D; \mathbb{R}^{d_z})$ are separable Banach spaces of functions. To train neural operators we consider the following standard data scenario. The set $\{u^{(j)}, z^{(j)}\}_{j=1}^J$ of input-output observation pairs is such that $\left(u^{(j)}, z^{(j)}\right) \sim \pi$ is an i.i.d. sequence distributed according to the measure π, which is supported on $\mathcal{U} \times \mathcal{Z}$. We let μ and π denote the marginals of π on $\mathcal{U}$ and $\mathcal{Z}$ respectively, so that $u^{(j)} \sim \mu$ and $z^{(j)} \sim \pi$. In what follows, we consider $\Psi^\dagger : \mathcal{U} \to \mathcal{Z}$ to be an arbitrary map so that

$$z^{(j)} = \Psi^\dagger\left(u^{(j)}\right), \tag{4.3}$$

noting that such maps will typically be both nonlocal (with respect to domain D) and nonlinear. We assume for simplicity that the data generation mechanism is compatible with (4.3) so that $\pi = \Psi^\dagger_\sharp \mu$, i.e., the data generation mechanism is not corrupted by noise. Neural operators construct an approximation of $\Psi^\dagger$ picked from the parametric class

$$\Psi : \mathcal{U} \times \Theta \to \mathcal{Z};$$

equivalently we may write $\Psi_\theta : \mathcal{U} \to \mathcal{Z}$, for $\theta \in \Theta$. The set Θ is assumed to be a finite dimensional parameter space from which a $\theta^* \in \Theta$ is selected so that $\Psi(\cdot, \theta^*) = \Psi_{\theta^*} \approx \Psi^\dagger$. We then define a cost functional $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ and define θ^* by

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}_{u \sim \mu}\Big[c\left(\Psi(u, \theta), \Psi^\dagger(u)\right)\Big]. \tag{4.4}$$

In practice, we will only have access to the values of each $u^{(j)}$ and $z^{(j)}$ at a set of discretization points of the domain D, which we denote as $\{x_i\}_{i=1}^N$. We seek

discretization invariant approaches to operator learning which allow for sharing of learned parameters θ^* between different discretizations.

The choice of an architecture defining the approximation family Ψ_θ is sometimes known as surrogate modeling. We use continuum attention concepts to make this choice. The resulting transformer based neural operator methodology that we develop here can be used to learn solution operators to parametric PDEs, to solve PDE inverse problems, and may also be applied to solve a class of data assimilation problems. In the context of parametric PDEs we will explore how this algorithmic framework may be employed in operator learning for Darcy flow and 2d Navier-Stokes problems. As a particular case of operator learning, we will also explore a class of data assimilation problems that involve recovering unobserved trajectories of possibly chaotic dynamical systems. Namely, we illustrate examples involving the Lorenz-63 dynamical system and a controlled ODE.

4.1.1 Literature Review

The attention mechanism, as introduced in Vaswani et al. (2017) and defined in (4.1), has shown widespread success for a variety of tasks. In particular, given the suitability of attention and transformers for applications involving sequential data, there has been a large body of work extending the methodology from natural language processing, the setting in which it was originally introduced, to the more general context of time series. Namely, as surveyed in Wen et al. (2023), transformers and variants thereof have been applied to time series forecasting, classification and anomaly detection tasks. This generalization has involved modifications and innovations relating to the positional encoding and the design of the attention function itself. We refer the reader to Wen et al. (2023) for a detailed review of the large body of work and architectural variants of transformers that have been developed for time series applications.

Attention in Vision The quadratic scaling in the input of the attention mechanism has made transformers a computationally prohibitive architecture for applications involving long sequences. Nonetheless, through techniques such as patching, the attention mechanism and transformers have also enjoyed success in computer vision, where high-resolution image data presents an increased computational cost. In fact, in vision transformers (ViT), introduced in Dosovitskiy et al. (2021), the input image is subdivided into patches; attention is then applied across the shorter sequence of patches, thus decreasing the overall cost.

Neural Operators Given the success of the methodology in computer vision, there has been growing interest in extending the transformer methodology to the operator learning context. This operator setting concerns learning mappings between infinite dimensional spaces of functions. In this case, the domains on which the functions are defined (spatial and temporal) may inherently be a continuum. Neural operators (N. Kovachki et al., 2023a), generalizations of neural networks that learn operators mapping between infinite-dimensional spaces, have been developed for this task. As they parametrize operators acting on spaces of functions, the learned parameters of neural operators may be applied to any discretization of the input function; this property of neural operators is referred to as "discretization invariance". Discretization invariance is desirable both at a theoretical and practical level. Indeed, for a neural network to approximate an operator acting on functions defined on a continuum, its parameters cannot depend on the discretizations of the input functions that are used for learning the parameters themselves. On the other hand, discretization invariance allows to scalably train models on coarser discretizations of functions and to deploy or finetune trained models at different resolutions for downstream tasks. A recent example of the success of this approach is the GenCast medium-range weather forecasting model (Price et al., 2025a), which because of discretization invariance properties is pretrained on low resolutions and finetuned at the desired predictive resolution.

Transformers for Operator Learning In the following paragraphs, we discuss several approaches in the literature employing transformers for operator learning. We first discuss some approaches leading to neural operators, followed by a discussion of architectures that are not discretization invariant.

Neural Operators. An integral kernel interpretation of the attention mechanism appearing for example in Tsai et al. (2019), Martins et al. (2020), Moreno et al. (2022), Wright and Gonzalez (2021), Cao (2021), and N. Kovachki et al. (2023a) inspired a range of attention-based architectures for operator learning. For example, the work of Cao (2021) leverages insight from the integral kernel formulation to design novel attention mechanisms that do not employ the softmax function. The author designs attention mechanisms using integral kernels which may be approximated by a Fourier-type projection, leading to quadratic complexity, and by a Galerkin projection, leading to linear complexity. While the resulting attention maps are discretization invariant, the full architecture proposed uses convolutional neural networks to downsample the input to a feature map of sufficiently small

resolution for the method to be scalable. Hence the full architecture is not a neural operator. Furthermore, the lack of the softmax nonlinearity leads to a mechanism lacking the probabilistic description outlined in this chapter, thus differing from the original definition of attention in Vaswani et al. (2017). In Zhijie Li et al., 2024 the authors take the Fourier attention approach as in Cao, 2021 to design a neural operator for large eddy simulation, while in Luo, Qian, and Yoon, 2024 the authors take the Galerkin attention approach from Cao, 2021 to design a novel neural operator architecture. In Zijie Li, Meidani, and Farimani (2023) the authors present "OFormer" a neural operator with an encoder-decoder structure that employs cross-attention to obtain the latent embeddings and then stacked self-attention blocks; the model uses the attention from Cao (2021) and random Fourier features for query coordinate encodings while using a recurrent multi-layer perceptron for unrolling in time-dependent problems as opposed to masked attention in Vaswani et al. (2017). In the recent work Rahman et al. (2024), the authors propose a neural operator architecture that employs an attention mechanism that acts across channels of the function in the latent embedding, and do so in the context of pre-training a foundation model for PDE solution operators. We demonstrate how such an architecture can be derived from our formulation in Remark 4.3.4. As in this chapter, various attention-related concepts are defined in a function-space setting, but our formulations are more general; this issue is discussed in detail at the relevant point in the chapter. We note that the architectures introduced in this chapter rely solely on the attention mechanism defined in the continuum, hence they are neural operators; furthermore, because of the simple design, with a slight modification to our architectures we may prove the first universal approximation theorem for transformer operators.

Non discretization invariant architectures. In Guibas et al. (2022) the authors propose a modification of the Fourier neural operator that involves mixing of tokens in Fourier space. This is achieved by first applying the FFT to the sequence of tokens and then applying a MLP to each token, before inverting back to the original space. This approach differs fundamentally from the original definition of attention in Vaswani et al. (2017) and lacks the probabilistic description outlined in this chapter. Furthermore, because of learnable positional embeddings and patching via convolution, the AFNO is not discretization invariant. Various other approaches have been developed. The work Hao et al. (2023) introduces "GNOT", an architecture employing a normalized attention layer that handles multiple inputs. This design, along with a learned gating mechanism for input coordinates, allows the model to be applied to irregular meshes and multi-scale problems; however, this

architecture employs an attention mechanism that does not involve rescaling by the grid, hence not directly allowing for discretization invariance. In Ovadia et al. (2024) the authors propose to use an architecture composed of a ViT in the latent space of a U-Net (Ronneberger, Fischer, and Brox, 2015). The proposed model takes as input the input field evaluated on a grid with associated coordinate values and outputs an approximation of the solution to PDE inverse problems. While it is shown that the architecture achieves state-of-the-art accuracy, the parameters in the scheme are not invariant to the input function resolution. In Wang et al. (2025a) the authors design "Continuous ViT", an architecture consisting of a ViT encoder and a cross-attention decoder. The method handles input functions defined on space and time and the method is shown to present evidence of discretization invariance, with test discretizations close to training discretizations. However, while the architecture is discretization invariant in time, it is not by design in space. This is because the tokenization procedure employed by standard ViTs is not discretization invariant; we elaborate on this point in Remark 4.3.3.

Transformers for Continuum Data The use of attention-based transformer models in the context of dynamical systems and data assimilation problems, where spatial and temporal domains of the systems are continuous, is at its infancy. For a comprehensive review on the application of machine learning to data assimilation we refer the reader to Cheng et al. (2023a). In this context, the efficacy of transformers has been explored for forecasting in numerical weather prediction. In Chattopadhyay et al. (2022), a transformer module is integrated in the latent space of a U-Net (Ronneberger, Fischer, and Brox, 2015) architecture in order to leverage the permutation equivariance property of the transformer to improve physical consistency and forecast accuracy. The work in Bi et al. (2023) achieves state-of-the-art performance among end-to-end learning methods for numerical weather prediction by applying a ViT-like architecture and employing the shifted window methodology from Z. Liu et al. (2021). These approaches, however, are not discretization invariant by design.

Mathematics of Transformers Mathematical foundations for the methodology are only now starting to emerge, however most analysis has been preformed in the finite-dimensional setting. In the recent work Geshkovski, Letrouit, et al. (2023), the authors study the attention dynamics by viewing tokens as particles in an interacting particle system. The formulation of the continuous-time limit of the dynamics, where the limit is taken in "layer time" allows to investigate the emergence of clusters among tokens. In Yun et al. (2020), the authors prove that transformers are uni-

versal approximators of continuous permutation equivariant sequence-to-sequence functions with compact support. More broadly, developing a firm theoretical framework for the subject is necessary to understand the empirical performance of the methodology and to inform the design of novel attention-based architectures.

This Work This chapter bridges the gap between the recent theoretical developments for the attention mechanism and practical application in operator learning. In fact, starting from a description of the attention mechanism as a map on spaces of functions, we build discretization invariant transformer blocks that may be directly deployed for operator learning tasks. A numerical investigation shows that the ensuing transformer neural operators are competitive with state-of-the-art architectures for a range of operator learning tasks. The framework developed may be used more generally to design larger, more complex architectures for which the universal approximation theorem proved in this chapter may potentially be generalized.

4.1.2 Contributions and Outline

Motivated by the success of transformers for sequential data, we set out to formulate the attention mechanism and transformer architectures as mappings between infinite dimensional spaces of functions. In this work we introduce a continuum perspective on the transformer methodology. The resulting theoretical framework enables the design of attention-based neural operator architectures, which find a natural application to operator learning problems. The strength of the approach of devising implementable discrete architectures from the continuum perspective stems from the discretization invariance property that the schemes inherit. Indeed, the resulting neural operator architectures are designed as mappings between infinite dimensional functions spaces that are invariant to the particular discretization, or resolution, at which the input functions are defined. Notably, this allows for zero-shot generalization to different function discretizations. With these insights in view, our main contributions can be summarized as follows:

1. Formulation of attention as a mapping between function spaces.

2. Approximation theorem that quantifies the error between application of the attention operator to a continuous function and the result of applying its finite dimensional analogue to a discretization of the same function. The relevant results may be found in Theorems 4.2.6 and 4.2.12.

3. A formulation of the transformer architecture as a neural operator, hence as a mapping between function spaces; the resulting scheme is resolution invariant and allows for zero-shot generalization to irregular, non-uniform meshes.

4. A first universal approximation theorem for transformer neural operators. The relevant results may be found in Theorems 4.5.1 and 4.5.2.

5. Formulation of patch-attention as a mapping between spaces of functions acting on patch indices. This continuum perspective leads to the design of efficient attention-based neural operator architectures.

6. Numerical evidence of the competitiveness of transformer-based neural operators with state-of-the-art operator learning architectures.

Section 4.2 is focused on contributions 1 and 2, formulating attention as acting on functions defined on a continuous domain. In Section 4.3 we formulate the attention mechanism as acting on a sequence of "patches" of a function, addressing contribution 5. In Section 4.4 we make use of the operator frameworks developed in Sections 4.2 and 4.3 to formulate transformer neural operator architectures acting on function space. Thus we address contribution 3, whilst the last two subsections concern the algorithmic aspect of extending to patching and address contribution 5. In Section 4.5 we state and prove a first universal approximation theorem for a transformer-based neural operator, hence addressing contribution 4. Finally, in Section 4.6 we explore applications of the transformer neural operator architectures to a variety of operator learning problems (contribution 6).

4.2 Continuum Attention

In this section we formulate a continuum analogue of the attention mechanism described in Vaswani et al. (2017). We note that in the transformer architecture of Vaswani et al. (2017), a distinction is made between the self-attention and cross-attention functions. The former, a particular instance of (4.1) where $u = v$, can be used to produce a representation of the input that captures intra-sequence correlations. On the other hand, cross-attention is used to output a representation of the input u that models cross-correlations between both inputs u and v. Indeed, in Vaswani et al. (2017) cross-attention is employed in the decoder component of the transformer architecture so that the input to the decoder attends to the output of the encoder. For natural language applications in particular, the form of the cross-attention function has allowed for sequences of arbitrary length N to be outputted

from a transformer architecture while attending to sequences of different lengths, e.g. as outputs of length M from a self-attention encoder block.

For pedagogical purposes we subdivide the presentation between the self-attention operator and the cross-attention operator, which are treated in Subsection 4.2.1 and Subsection 4.2.2 respectively. In Subsection 4.2.1.1 we define self-attention for discrete sequences indexed on finite bounded sets. Here, we outline the interpretation of self-attention as an expectation under a family of carefully defined discrete probability measures, each acting on the space of sequence indices. We show that under this formulation we recover the standard definition of self-attention from Vaswani et al. (2017). We proceed in Subsection 4.2.1.2 by formulating self-attention as an operator mapping between spaces of functions defined on bounded uncountable sets. This interpretation is natural, as functions may be thought of as "sequences" indexed over an uncountable domain. As in the discrete case, we formulate the self-attention operator as an expectation under a carefully defined probability measure acting on the domain of the input. We obtain an approximation result given by Theorem 4.2.6 that shows that for continuous functions the self-attention mechanism as implemented in practice may be viewed as a Monte Carlo approximation of the self-attention operator described. We mimic these constructions in Subsections 4.2.2.1 and 4.2.2.2 for the discrete and continuous formulations of cross-attention, respectively, and obtain an analogous approximation result given by Theorem 4.2.12. As in (4.1), throughout this section $Q \in \mathbb{R}^{d_K \times d_u}, K \in \mathbb{R}^{d_K \times d_v}, V \in \mathbb{R}^{d_V \times d_v}$ are learnable weights that parametrize the attention operators. When we consider self-attention we will have $d_v = d_u$.

4.2.1 Self-Attention

In the following discussion we define the self-attention operator in the discrete setting and in the continuum setting, in Subsections 4.2.1.1 and 4.2.1.2, respectively.

4.2.1.1 Sequences over $D^N \subset \mathbb{Z}$

We begin by considering the finite, bounded set $D^N = \{1, \cdots, N\}$. We let $u : D^N \to \mathbb{R}^{d_u}$ be a sequence and $j \in D^N$ an index so that $u(j) \in \mathbb{R}^{d_u}$. We will now focus on formulating the attention mechanism in a manner that allows generalization to more general domains D^N; at the end of this subsection we connect our formulation with the more standard one used in the literature, which is specific to the current choice of domain $D^N = \{1, \cdots, N\}$. We define the following discrete probability measure.

Definition 4.2.1. *We define a probability measure on D^N, parameterized by (u, j), via the probability mass function $p(\cdot; u, j) : D^N \to [0, 1]$ defined by*

$$p(k; u, j) = \frac{\exp\Big(\big\langle Qu(j), Ku(k)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\sum_{\ell \in D^N} \exp\Big(\big\langle Qu(j), Ku(\ell)\big\rangle_{\mathbb{R}^{d_K}}\Big)},$$

for any $k \in D^N$.

We may now define the self-attention operator as an expectation over this measure p of the linear transformation $V \in \mathbb{R}^{d_V \times d_u}$ applied to u.

Definition 4.2.2. *We define the self-attention operator A^N as a mapping from a $\mathbb{R}^{d_u}$-valued sequence over D^N, u, into a $\mathbb{R}^{d_V}$-valued sequence over D^N, $\mathsf{A}^N(u)$, that takes the form*

$$\mathsf{A}^N(u)(j) := \mathbb{E}_{k \sim p(k;u,j)}[Vu(k)],$$

for any $j \in D^N$.

To connect these definitions with the standard formulation of transformers, note that sequence u can be reformulated as the matrix $u \in \mathbb{R}^{N \times d_u}$. Then

$$\{uQ^T Ku^T\}_{jk} = \big\langle Qu_j, Ku_k\big\rangle_{\mathbb{R}^{d_K}};$$

that is, the $\{j, k\}-$entry of $uQ^T Ku^T \in \mathbb{R}^{N \times N}$ is the right-hand side of the preceding identity, where u_ℓ denotes the ℓ'th row of $u \in \mathbb{R}^{N \times d_u}$. It is then apparent that applying the `softmax` function along rows of $uQ^T Ku^T$ delivers the vector of probabilities defined in Definition 4.2.1, indexed by k; note that j is a parameter in this vector of probabilities, indicating the row of the matrix $uQ^T Ku^T$ along which the `softmax` operation is applied. Finally the attention operation from Definition 4.2.2 can be rewritten to act between matrices as

$$\mathsf{A}^N(u) = \texttt{attention}(u, u) := \texttt{softmax}(uQ^T Ku^T)uV^T,$$

so that $\texttt{attention}(u, u) \in \mathbb{R}^{N \times d_V}$.

Remark 4.2.3 (Sequences over $D^N \subset \mathbb{Z}^d$). *Once the attention mechanism is formulated as in Definitions 4.2.1, 4.2.2, it is straightforward to extend to (generalized) sequences indexed over bounded subsets of $\mathbb{Z}^d$; note, for example, that pixellated images are naturally indexed over bounded subsets of $\mathbb{Z}^2$. Let $D^N \subset \mathbb{Z}^d$ possess cardinality $|D^N| = N$. Let $u : D^N \to \mathbb{R}^{d_u}$ be a sequence and $j \in D^N$ an index. Then*

the definition of probability on D^N, indexed by (u, j), and the resulting definition of self-attention operator A, is exactly as in Definitions 4.2.1, 4.2.2. This shows the power of non-standard notation we have employed here.

4.2.1.2 Sequences over $D \subset \mathbb{R}^d$

We note that the preceding two formulations of the attention mechanism view it as a transformation defined as an expectation applied to the input sequence. The expectation is with respect to a probability measure supported on an index of the input sequence. Furthermore, the expectation itself depends on the input sequence to which it is applied, and is parameterized by the input sequence index, resulting in a new output sequence of the same length as the input sequence. The formulation using an expectation results from the softmax part of the definition of the attention mechanism; the query and key linear operations define the expectation, and the value linear operation lifts the output sequence to take values in a (possibly) different Euclidean space than the input sequence. These components may be readily extended to work with (generalized) sequences defined over open subsets of $\mathbb{R}^d$. (These would often be referred to as functions, but we call them generalized sequences to emphasize similarity with the discrete case.)

Let $D \subseteq \mathbb{R}^d$ be an open set. Let $u : D \to \mathbb{R}^{d_u}$ be a sequence (function) and $x \in D$ an index.

Definition 4.2.4. *We define a probability measure on D, parameterized by (u, x), via the probability density function $p(\cdot; u, x) : D \to \mathbb{R}^+$ defined by*

$$p(y; u, x) = \frac{\exp\Big(\big\langle Qu(x), Ku(y)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\int_D \exp\Big(\big\langle Qu(x), Ku(s)\big\rangle_{\mathbb{R}^{d_K}}\Big)\,\mathrm{d}s},$$

for any $y \in D$.

Definition 4.2.5. *We define the self-attention operator A as a mapping from a $\mathbb{R}^{d_u}$-valued sequence (function) over D, u, into a $\mathbb{R}^{d_V}$-valued sequence (function) over D, $\mathsf{A}(u)$, as follows:*

$$\mathsf{A}(u)(x) = \mathbb{E}_{y\sim p(y;u,x)}[Vu(y)],$$

for any $x \in D$.

A connection between the discrete and continuum formulations of self-attention is captured in the following theorem.

Theorem 4.2.6. *The self-attention operator* A *may be viewed as a mapping* $\mathsf{A} : L^\infty(D;\mathbb{R}^{d_u}) \to L^\infty(D;\mathbb{R}^{d_V})$ *and thus as a mapping* $\mathsf{A} : C(\overline{D};\mathbb{R}^{d_u}) \to C(\overline{D};\mathbb{R}^{d_V})$. *Furthermore, for any compact set* $B \subset C(\overline{D};\mathbb{R}^{d_u})$,

$$\lim_{N\to\infty} \sup_{u\in B} \mathbb{E} \left\| \mathsf{A}(u) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Ku(y_j)\rangle\big) Vu(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Ku(y_\ell)\rangle\big)} \right\|_{C(\overline{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\overline{D})$.

Proof. The proof of this result is developed in Appendix 4.A.1. □

Remark 4.2.7. *This result connects our continuum formulation of self-attention with the standard definition, using Monte Carlo approximation of the relevant integrals; a similar result could be proved using finite difference approximation on, for example, a uniform grid. We note that it is not possible to obtain a uniform convergence rate for all* $u \in B$ *in the norm of* $C(\overline{D};\mathbb{R}^{d_V})$ *as the Riemann integral may exhibit arbitrarily slow convergence depending on the regularity of the integrand.*

4.2.2 Cross-Attention

In the following discussion we define the cross-attention operator in the discrete setting and in the continuum setting, in Subsections 4.2.2.1 and 4.2.2.2, respectively.

4.2.2.1 Sequences over $D^N \subset \mathbb{Z}^d, E^M \subset \mathbb{Z}^e$

We begin by letting $D^N \subseteq \mathbb{Z}^d$ and $E^M \subseteq \mathbb{Z}^e$ be finite sets of points with cardinalities N and M respectively. Let $u : D^N \to \mathbb{R}^{d_u}$ be a sequence (function), $v : E^M \to \mathbb{R}^{d_v}$ another sequence (function), and $j \in D^N$, $k \in \mathbb{E}^M$ indices so that $u(j) \in \mathbb{R}^{d_u}$ and $v(k) \in \mathbb{R}^{d_v}$. We define the following discrete probability measure.

Definition 4.2.8. *We define a probability measure on* D^N, *parameterized by* (u, v, j), *via the probability mass function* $q(\cdot; u, v, j) : D^N \to [0, 1]$ *defined by*

$$q(k; u, v, j) = \frac{\exp\Big(\big\langle Qu(j), Kv(k)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\sum_{\ell\in E^M} \exp\Big(\big\langle Qu(j), Kv(\ell)\big\rangle_{\mathbb{R}^{d_K}}\Big)},$$

for any $k \in E^M$.

We may now define the cross-attention operator as an expectation over this measure q of the linear transformation $V \in \mathbb{R}^{d_V \times d_v}$ applied to v.

Definition 4.2.9. *We define the cross-attention operator* C^N *as a mapping from a* $\mathbb{R}^{d_u}$*-valued sequence (function) over* D^N*, u, and a* $\mathbb{R}^{d_v}$*-valued sequence (function) over* E^M*, v, into a* $\mathbb{R}^{d_V}$*-valued sequence (function) over* D^N*,* $\mathsf{C}^N(u, v)$*, as follows:*

$$\mathsf{C}^N(u, v)(j) = \mathbb{E}_{k\sim q(k;u,v,j)}[Vv(k)],$$

for any $j \in D^N$*.*

4.2.2.2 Sequences over $D \subset \mathbb{R}^d, E \subset \mathbb{R}^e$

The framework described in Subsection 4.2.2.1 may be readily extended to work with generalized sequences (functions) defined over open subsets of $\mathbb{R}^d$. Indeed, let $D \subseteq \mathbb{R}^d$ and $E \subseteq \mathbb{R}^e$ be open sets. Let $u : D \to \mathbb{R}^{d_u}$ be a sequence (function), $v : E \to \mathbb{R}^{d_v}$ another sequence (function), and $x \in D$, $y \in E$ indices. We define the following probability measure over D before providing the definition of the cross-attention operator.

Definition 4.2.10. *We define a probability measure on D, parameterized by* (u, v, x)*, via the probability density function* $q(\cdot; u, v, x) : D \to \mathbb{R}^+$ *defined by*

$$q(y; u, v, x) = \frac{\exp\Big(\big\langle Qu(x), Kv(y)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\int_E \exp\Big(\big\langle Qu(x), Kv(s)\big\rangle_{\mathbb{R}^{d_K}}\Big)\,\mathrm{d}s},$$

for any $y \in E$*.*

Definition 4.2.11. *We define the cross-attention operator* C *as a mapping from a* $\mathbb{R}^{d_u}$*-valued sequence (function) over D, u, and a* $\mathbb{R}^{d_v}$*-valued sequence (function) over E, v, into a* $\mathbb{R}^{d_V}$*-valued sequence (function) over D,* $\mathsf{C}(u, v)$*, as follows:*

$$\mathsf{C}(u, v)(x) = \mathbb{E}_{y\sim q(y;u,v,x)}[Vv(y)],$$

for any $x \in D$*.*

A connection between the discrete and continuum formulations of cross-attention is captured in the following theorem.

Theorem 4.2.12. *The cross-attention operator* C *may be viewed as a mapping* $\mathsf{C} : L^\infty(D; \mathbb{R}^{d_u}) \times L^\infty(E; \mathbb{R}^{d_v}) \to L^\infty(D; \mathbb{R}^{d_V})$*, and so as a mapping* $\mathsf{C} : C(\overline{D}; \mathbb{R}^{d_u}) \times C(\overline{E}; \mathbb{R}^{d_v}) \to C(\overline{D}; \mathbb{R}^{d_V})$*. Furthermore, for any compact set* $B \subset C(\overline{D}; \mathbb{R}^{d_u}) \times C(\overline{E}; \mathbb{R}^{d_v})$

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E} \left\| \mathsf{C}(u,v) - \frac{\sum_{j=1}^{N} \exp\big(\langle Qu(\cdot), Kv(y_j)\rangle\big) Vv(y_j)}{\sum_{\ell=1}^{N} \exp\big(\langle Qu(\cdot), Kv(y_\ell)\rangle\big)} \right\|_{C(\overline{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^{N} \sim \mathsf{unif}(\overline{E})$.

Proof. The proof of this result is developed in Appendix 4.A.2, which contains straightforward generalizations of the techniques in Appendix 4.A.1. □

4.3 Continuum Patched Attention

Patches of a function are defined as the restriction of the function itself to elements of a partition of the domain. We may then define patch-attention as a mapping between spaces of functions defined on patch indices. Interest in applying the attention mechanism for computer vision led to the development of methodologies to overcome its prohibitive computational cost. Patching, as employed in vision transformers (ViT), involves subdividing the data space domain into $P \in \mathbb{N}$ "patches" and applying attention to a sequence of flattened patches. This step drastically reduces the complexity of the attention mechanism, which is quadratic in the input sequence length. An analogue of the patching strategy in vision transformers can be considered in the function space setting. Indeed, in image space, attention can be applied across subsets of pixels (patches); in function space, attention can be applied across the function defined on elements of a partition of the domain. Such a generalization to functions defined on the continuum allows for the development of architectures that are mesh-invariant. We describe the patched-attention methodology in the continuum framework developed thus far.

Throughout this section, we let $D \subset \mathbb{R}^d$ be a bounded open set and let $D := D_1 \cup \cdots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ are congruent for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We note that we require a uniform partition, namely a partition of subsets of equivalent finite volume, in order to define operators that can be applied to functions defined on each of these subsets; however, for practical application we may have a different number of discretization points within each subset. We consider a function defined on the full domain D, denoted as $u \in \mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$. We define a mapping

$$\widetilde{u} : [\![1, P]\!] \to \mathcal{U}(D', \mathbb{R}^{d_u}), \tag{4.5}$$

that defines the patched version of the function $u \in \mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$, so that

$$\widetilde{u}(p) = u|_{D_p}, \tag{4.6}$$

for any $p \in [\![1, P]\!]$. We proceed to establish the definitions of the patched self-attention operator $\mathsf{A_P}$ in Subsection 4.3.1 and for completeness the patched cross-attention operator $\mathsf{C_P}$ in Subsection 4.3.2.

4.3.1 Patched Self-Attention

We begin with the definition of the patched self-attention operator $\mathsf{A_P}$. Indeed, we let the operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ be defined such that $\mathsf{Q} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{K} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{V} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to \mathcal{U}(D', \mathbb{R}^{d_V})$ and let $j \in [\![1, P]\!]$ be an index.

Definition 4.3.1. *Define a probability measure on $[\![1, P]\!]$, parameterized by $(\widetilde{u}, j)$, via the probability density function $p(\cdot; \widetilde{u}, j) : [\![1, P]\!] \to [0, 1]$ defined by*

$$p(k; \widetilde{u}, j) = \frac{\exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{u}(k) \big\rangle_{L^2(D', \mathbb{R}^{d_K})}\Big)}{\sum_{\ell=1}^{P} \exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{u}(\ell) \big\rangle_{L^2(D', \mathbb{R}^{d_K})}\Big)},$$

for any $k \in [\![1, P]\!]$.[2]

Definition 4.3.2. *The self-attention operator $\mathsf{A_P}$ maps the function taking a patch index to a $\mathbb{R}^{d_u}$-valued function over D', $\widetilde{u}$, into a function taking a patch index to a $\mathbb{R}^{d_V}$-valued function over D', $\mathsf{A_P}(\widetilde{u})$, as follows:*

$$\mathsf{A_P}(\widetilde{u})(j) = \mathbb{E}_{k \sim p(k; \widetilde{u}, j)}[\mathsf{V}\widetilde{u}(k)],$$

for any $j \in [\![1, P]\!]$.

Remark 4.3.3 (Patched Attention in ViT). *An image may be viewed as a mapping $u : D \to \mathbb{R}^{d_u}$, where $D \subset \mathbb{Z}^2$, and where $d_u = 1$ for grey-scale and $d_u = 3$ for RGB-valued images. In the context of vision transformers (Dosovitskiy et al., 2021), the input to the attention mechanism is a sequence of P "flattened" patches of the input image; we next outline the details to this procedure.*

Letting N be the total number of discretization points in D and P the number of patches, in the discrete setting each function patch $\widetilde{u}(p) : D' \to \mathbb{R}^{d_u}$ may be represented as an $\mathbb{R}^{N/P \times d_u}$ matrix. The flattening procedure of each patch in ViT involves reshaping this matrix into an $\mathbb{R}^{N/P \cdot d_u}$ vector. The full image is thus represented as a sequence of P vectors of dimension $N/P \cdot d_u$. Each patch vector is then linearly lifted to an embedding space of dimension d_{model}. The

[2] We note that as Q is linear, $\mathsf{Q}u(j)$ is shorthand notation for $\mathsf{Q}\big(u(j)\big)$, and similarly for K and V.

embedded image, to which attention is applied, may hence be viewed as a matrix $\widetilde{u} \in \mathbb{R}^{P \times d_{model}}$*. Therefore, ViT may be cast in the setting of the discrete interpretation of the self-attention operator outlined in Subsection 4.2.1.1, where the domain on which attention is applied is* D^P*. A key observation to the ensuing discussion is that the application of a linear transformation to a "flattened" patch in ViT breaks the mesh-invariance of the architecture; however, this operation may be viewed as a nonlocal transformation on each patch. We leverage this insight in Section 4.4 to design discretization invariant transformer neural operators that employ patching. We further note that in practice, patching in ViT is often also achieved via strided convolutions, with stride equivalent to the kernel size; however, because the kernel size determines the number of parameters, such a procedure also breaks discretization invariance as it leads to architectures with parameters dependent on resolution of the input.*

Remark 4.3.4 (Codomain Attention). *The work of Rahman et al. (2024) employs an attention mechanism that acts across the channels of the function* $u \in \mathcal{U}\big(D; \mathbb{R}^{d_{model}}\big)$ *in the latent space of a neural operator architecture. Indeed the "codomain" attention mechanism defined may be viewed as a particular variant of Definition 4.3.2, where* $D' \cong D$*, where* $\mathsf{A}_\mathsf{P}(\widetilde{u})(j)$ *is defined for* $j \in [\![1, d_{model}]\!]$ *and where* $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ *are implemented as integral operators. In particular the indexing is by channels and not by patch. For problems in computer vision, similar ideas have been explored with discrete versions of the* $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ *operators (L. Chen et al., 2017).*

4.3.2 Patched Cross-Attention

For completeness, we outline the definition of the patched cross-attention operator. We consider $E \subset \mathbb{R}^e$ to be a bounded open set and $E \coloneqq E_1 \cup \ldots \cup E_O$ a uniform partition of E so that $E_j \cong E' \cong D'$ for all $j \in [\![1, O]\!]$ for some $E' \subset E$ and $O \in \mathbb{N}$. We note that for the following construction we require $E' \cong D'$, but it may hold in general that $P \neq O$. As done for $\widetilde{u}$ we define an operator

$$\widetilde{v} : [\![1, O]\!] \to \mathcal{U}(E', \mathbb{R}^{d_v}), \tag{4.7}$$

that defines the patched version of the function $v \in \mathcal{U}\big(E; \mathbb{R}^{d_v}\big)$, so that

$$\widetilde{v}(o) = v|_{E_o}, \tag{4.8}$$

for any $o \in [\![1, O]\!]$. Furthermore, we let the operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ be defined such that $\mathsf{Q} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{K} : \mathcal{U}(E', \mathbb{R}^{d_v}) \to L^2(E', \mathbb{R}^{d_K}), \mathsf{V} : \mathcal{U}(E', \mathbb{R}^{d_v}) \to \mathcal{U}(E', \mathbb{R}^{d_V})$ and let $j \in [\![1, P]\!]$ be an index.

Definition 4.3.5. *We define a probability measure on D, parameterized by $(\tilde{u}, \tilde{v}, j)$, via the probability density function $q(\cdot; \widetilde{u}, \widetilde{v}, j) : [\![1, P]\!] \to [0, 1]$ defined by*

$$q(k; \widetilde{u}, \widetilde{v}, j) = \frac{\exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{v}(k) \big\rangle_{L^2(E', \mathbb{R}^{d_K})}\Big)}{\sum_{\ell=1}^{O} \exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{v}(\ell) \big\rangle_{L^2(E', \mathbb{R}^{d_K})}\Big)},$$

for any $k \in [\![1, O]\!]$.

Definition 4.3.6. *The cross-attention operator C maps the operator taking a patch index to a $\mathbb{R}^{d_u}$-valued function over D', $\widetilde{u}$, and the operator taking a patch index to a $\mathbb{R}^{d_v}$-valued function over E', $\widetilde{v}$, into an operator taking a patch index to a $\mathbb{R}^{d_V}$-valued function over D', $\mathsf{C}_\mathsf{P}(\widetilde{u}, \widetilde{v})$, as follows:*

$$\mathsf{C}_\mathsf{P}(\widetilde{u}, \widetilde{v})(j) = \mathbb{E}_{k \sim q(k; \widetilde{u}, \widetilde{v}, j)}[\mathsf{V}\widetilde{v}(k)],$$

for any $j \in [\![1, P]\!]$.

4.4 Transformer Neural Operators

In this section we turn our focus to describing how to devise transformer-based neural operators using the continuum attention operator A as defined in the function space setting in Section 4.2, and the continuum patched-attention operator A_P acting on function space from Section 4.3. We make use of the operator frameworks developed in Sections 4.2 and 4.3 to formulate transformer neural operator architectures acting on function space. We start in Subsection 4.4.1, setting-up the framework. In Subsection 4.4.2 we formulate an analogue of the transformer from Vaswani et al. (2017) as an operator mapping an input function space $\mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ to the solution function space $\mathcal{Z}\big(D; \mathbb{R}^{d_z}\big)$. The resulting scheme, which we define as transformer neural operator (TNO) is invariant to the discretization of the input function and allows for zero-shot generalization to irregular, non-uniform meshes. However, this architecture exhibits quadratic complexity with respect to the number of discretization points of the input functions. Hence, motivated by the need to develop new efficient architectures to employ attention on longer sequences, or in our case higher resolution discretizations of functions, in Subsection 4.4.3 we describe a generalization of the ViT methodology to the function space setting, inspired by the work in Dosovitskiy et al. (2021). In particular we generalize the procedure of lifting flattened patches to the embedding dimension, as done in ViT, to the function space setting. This is achieved by lifting patches of functions to an embedding dimension using a nonlocal linear operator, namely, an integral operator. Unlike ViT, the resulting neural operator architecture, which we call ViT neural

operator (ViTNO), is discretization invariant and allows for zero-shot generalization to different resolutions. In Subsection 4.4.4 we describe a different patching-based transformer neural operator acting on function space, the Fourier attention neural operator (FANO). The architecture is discretization invariant and is based on $\mathsf{Q}, \mathsf{K}, \mathsf{V}$, learnable parameters in the attention mechanism, being defined as integral operators acting themselves on function space. We demonstrate in Section 4.6 that the approach involving parametrizing $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ as integral operators yields an architecture with greater parameter count for a similar computational complexity. We show that this architecture produces improved results in the context of a problem with smooth inputs.

4.4.1 Set-Up

In the subsections that follow we describe transformer neural operators of the form $\Psi : \mathcal{U}(D; \mathbb{R}^{d_u}) \times \Theta \to \mathcal{Z}(D; \mathbb{R}^{d_z})$, which serve as approximations to an operator $\Psi^\dagger : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{Z}(D; \mathbb{R}^{d_z})$. The neural operators we devise have the general form

$$\Psi(u, \theta) \coloneqq \Big(\mathsf{T}_{\text{out}} \circ \mathsf{E}_L \circ \mathsf{T}_{\text{in}}\Big)(u, \theta), \tag{4.9}$$

for any $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ and $\theta \in \Theta$. In (4.9), the operators $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \times \Theta \to \mathcal{V}$ and $\mathsf{T}_{\text{out}} : \mathcal{V} \times \Theta \to \mathcal{Z}(D; \mathbb{R}^{d_z})$ are defined for some appropriate embedding function space $\mathcal{V}$. We will make explicit the definitions of the general operators T_{in}, T_{out} and the embedding function space $\mathcal{V}$ in the context of each architecture. Throughout this section, we drop explicit dependencies on θ for brevity unless we wish to stress a parametric dependence; for example, we may abuse notation and write $\mathsf{T}_{\text{in}}(u) := \mathsf{T}_{\text{in}}(u;\ \theta)$. The operator $\mathsf{E}_L : \mathcal{V} \times \Theta \to \mathcal{V}$ defines the neural operator analogue of the transformer encoder block from Vaswani et al. (2017). Its action on the input $v^{(0)} \coloneqq \mathsf{T}_{\text{in}}(u) \in \mathcal{V}$ is summarized by the iteration

$$v^{(l-1)} \hookleftarrow \mathsf{W}_1 v^{(l-1)} + \mathsf{A}_{\text{MultiHead}}\big(v^{(l-1)}\big), \tag{4.10a}$$

$$v^{(l-1)} \hookleftarrow \mathsf{F}_{\text{LayerNorm}}\big(v^{(l-1)}\big), \tag{4.10b}$$

$$v^{(l-1)} \hookleftarrow \mathsf{W}_2 v^{(l-1)} + \mathsf{F}_{\text{NN}}\big(v^{(l-1)}\big), \tag{4.10c}$$

$$v^{(l)} \hookleftarrow \mathsf{F}_{\text{LayerNorm}}\big(v^{(l-1)}\big), \tag{4.10d}$$

for $l = 1, \ldots, L$ layers, so that $\mathsf{E}_L\big(v^{(0)}\big) = v^{(L)} \in \mathcal{V}$ [3]. We note that for notational convenience, we have suppressed dependence of the operators on the layer ℓ; indeed,

[3] We note that we use v to denote an arbitrary function in the space $\mathcal{V}$. This is not to be confused with v used for cross-attention in Section 4.2. We also denote the output of the ℓ'th encoder layer by $v^{(\ell)}$, not to be confused with the notation used for the data pairs.

each of the operators $\mathsf{A}_{\text{MultiHead}}$, $\mathsf{W}_1, \mathsf{W}_2$, F_{NN} and $\mathsf{F}_{\text{LayerNorm}}$, will have different learnable parametrizations for each encoder layer ℓ. Before we delve into the specific neural operator architectures, we summarize the definitions of the elements of the transformer encoder described in (4.10), which are common to all the subsequent architectures. The multi-head self-attention operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is defined as the composition of a linear transformation $W_{\text{MultiHead}}$ applied pointwise to its input and a concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators so that

$$\Big(\mathsf{A}_{\text{MultiHead}}(v)\Big)(x) = W_{\text{MultiHead}}\Big(\mathsf{A}_{\bullet}(v;\theta_1)(x), \dots, \mathsf{A}_{\bullet}(v;\theta_H)(x)\Big), \tag{4.11}$$

for any $v \in \mathcal{V}$. Here the notation $\mathsf{A}_{\bullet}$ indicates the ambiguity in the definition of self-attention. We note that we will make explicit which definition of self-attention operator from the previous sections we employ for each architecture; we will also make clear the definitions of the parameters $\theta_h \in \Theta$ defining each attention operator. The operators $\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$ are pointwise linear operators.[4] Next, the operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ defines the layer normalization and the operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ defines a feed-forward neural network layer. Both operators are applied pointwise to the input; further details are provided in the context of each architecture.

A key aspect of the ensuing discussion will be the distinction between local and nonlocal linear operators. We note that for an arbitrary function space $\mathcal{U}\big(D;\mathbb{R}^r\big)$ where $D \subset \mathbb{R}^d$, the space of linear operators $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ includes both local and nonlocal linear operators. In the following, we will consider pointwise linear operators in $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ that admit a finite-dimensional representation $W \in \mathbb{R}^{r'\times r}$. On the other hand, we will also consider linear integral operators in $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$, a class of nonlocal operators. Namely, we consider operators W that are integral operators learned from data of the form $\mathsf{W} : \Theta_{\mathsf{W}} \to \mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$.

Definition 4.4.1 (Integral Operator W). *The integral operator* $\mathsf{W} : \Theta_{\mathsf{W}} \to \mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ *is defined as the mapping given by*

$$\big(\mathsf{W}(\theta)u\big)(x) := \int_D \kappa_\theta(x,y)u(y)\mathrm{d}y, \qquad \forall u \in \mathcal{U}\big(D;\mathbb{R}^r\big),\ \forall x \in D, \tag{4.12}$$

where $\kappa_\theta : D \to \mathbb{R}^{r'\times r}$ *is a neural network parametrized by* $\theta \in \Theta_{\mathsf{W}}$.

Assuming $\kappa_\theta(x,y) = \kappa_\theta(x-y)$, hence the kernel to be periodic, then (4.12) may be viewed as a convolution operator, which makes it possible to use the Fast Fourier

[4] We note that for all experiments in Section 4.6 we set $\mathsf{W}_1, \mathsf{W}_2$ to be the identity operators.

Transform (FFT) to compute (4.12) and to parametrize W in Fourier space. Following Zongyi Li et al. (2021), this insight leads to the following formulation of the operator given in Definition 4.4.1.

Definition 4.4.2 (Fourier Integral Operator W). *We define the Fourier integral operator as*

$$\big(\mathsf{W}(\theta)u\big)(x) = \mathcal{F}^{-1}\Big(R_{\mathsf{W}}(\theta)\cdot(\mathcal{F}u)\Big)(x) \qquad \forall u \in \mathcal{U}\big(D;\mathbb{R}^r\big),\ \forall x \in D. \tag{4.13}$$

To link to the definition (4.12) *with translation-invariant kernel we take $R_{\mathsf{W}}(\theta)$ to be the Fourier transform of the periodic function $\kappa_\theta : D \to \mathbb{R}^{r'\times r}$ parametrized by $\theta \in \Theta_{\mathsf{W}}$.*

For uniform discretizations $n_1 \times \cdots \times n_d = N$ of the space D, the Fourier transform can be replaced with the Fast Fourier Transform (FFT). Indeed, for the Fourier transform we have $\mathcal{F}u : \mathbb{R}^d \to \mathbb{C}^r$ and $R_{\mathsf{W}}(\theta) : \mathbb{R}^d \to \mathbb{C}^{r'\times r}$. On the other hand, for the FFT we have $\mathcal{F}u : \mathbb{Z}^d \to \mathbb{C}^r$ and $R_{\mathsf{W}}(\theta) : \mathbb{Z}^d \to \mathbb{C}^{r'\times r}$. In practice, it is possible to select a finite-dimensional parametrization by choosing a finite set of wave numbers

$$\big|Z_{k_{\max}}\big| := \Big|\big\{k \in \mathbb{Z}^d : |k_j| \leqslant k_{\max,j},\ \text{for } j = 1,\dots,d\big\}\Big|.$$

We can therefore parametrize R_{W} by a complex $\big((2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times r'\times r\big)$-tensor. We will use the notation $k_{\max} = \big|Z_{k_{\max}}\big|$ when discussing parameter scalings in the subsequent sections.

These integral operator definitions will be used to define architectures for the Vision Transformer Neural Operator in Subsection 4.4.3 and the Fourier Attention Neural Operator in Subsection 4.4.4, but are not needed to define the basic Transformer Neural Operator in Subsection 4.4.2. In the context of each architecture, we make explicit the parametrizations of the integral operators employed.

4.4.2 Transformer Neural Operator

The transformer neural operator architecture follows the general structure of Equations (4.9) and (4.10); here, we make explicit the choices for $\mathsf{T}_{\text{in}}, \mathsf{T}_{\text{out}}, \mathsf{E}_L$ that instantiate it. In Figure 4.1 we provide a schematic representation of the full transformer neural operator architecture. We consider model inputs $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \mathcal{U}(D;\mathbb{R}^{d_{\text{model}}})$, and model output space $\mathcal{Z}(D;\mathbb{R}^{d_z})$, with $D \subset \mathbb{R}^d$. The function $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ is first concatenated with the identity

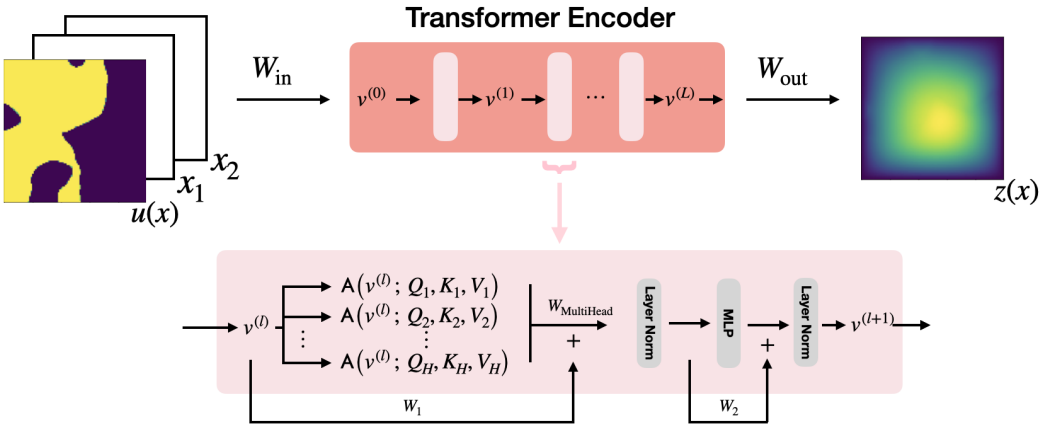


Figure 4.1: Transformer Neural Operator.

defined on the domain D, so that the resulting function $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u+d})$ is defined by its action on $x \in D$ as

$$u_{\text{in}}(x) := \big(u(x), x\big). \tag{4.14}$$

In practice, this step defines a positional encoding that is appended to the input function. A linear transformation $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u+d)}$ is then applied pointwise to lift the function u_{in} into the embedding space $\mathcal{V}$. The input operator $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$ is hence defined as

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(x) := W_{\text{in}} u_{\text{in}}(x) \tag{4.15}$$

for $u_{\text{in}} \in \mathcal{U}\big(D; \mathbb{R}^{d_u+d}\big)$ defined as in (4.14), where $x \in D$, $u(x) \in \mathbb{R}^{d_u}$, and $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u+d)}$. Observe that T_{in} acts as a local, pointwise linear operator on u_{in}.

Next, we define the details of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in (4.10) by specifying a definition of $\mathsf{A}_{\text{MultiHead}}$, $\mathsf{W}_1, \mathsf{W}_2$, $\mathsf{F}_{\text{LayerNorm}}$, and F_{NN} for the transformer neural operator setting. The operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (4.11) based on the continuum self-attention from Definition 4.2.5. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by the learnable linear transformations $Q_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K \times d_{\text{model}}}$, for $h = 1, \dots, H$, where for implementation purposes d_K is chosen as $d_K := d_{\text{model}}/H$.[5] The multi-head attention operator is thus defined by the application of a linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to the concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators $\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$

[5] We note that d_{model} and H should be chosen so that d_{model} is a multiple of H.

are pointwise linear operators and hence admit finite-dimensional representations $W_1, W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, so that they define a map

$$v(x) \mapsto W_1 v(x),$$

for each $x \in D$, and similarly for W_2. The operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\Big(\mathsf{F}_{\text{LayerNorm}}(v; \gamma, \beta)(x)\Big)_k = \gamma_k \cdot \frac{\big(v(x)\big)_k - m\big(v(x)\big)}{\sqrt{\sigma^2\big(v(x)\big) + \varepsilon}} + \beta_k, \tag{4.16}$$

for $k = 1, \ldots, d_{\text{model}}$, any $v \in \mathcal{V}$, and any $x \in D \subset \mathbb{R}^d$, where the notation $(\,\cdot\,)_k$ is used to denote the k'th entry of the vector. In equation (4.16), $\varepsilon \in \mathbb{R}^+$ is a fixed parameter, $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters and m, σ are defined as

$$\begin{aligned} m(y) &\coloneqq \frac{1}{d_{\text{model}}} \sum_{k=1}^{d_{\text{model}}} y_k, \\ \sigma^2(y) &\coloneqq \frac{1}{d_{\text{model}}} \sum_{k=1}^{d_{\text{model}}} \big(y_k - m(y)\big)^2, \end{aligned} \tag{4.17}$$

for any $y \in \mathbb{R}^{d_{\text{model}}}$. The operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\mathsf{F}_{\text{NN}}(v; W_3, W_4, b_1, b_2)(x) = W_3 f\big(W_4 v(x) + b_1\big) + b_2, \tag{4.18}$$

for any $v \in \mathcal{V}$ and $x \in D$, where $W_3, W_4 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $b_1, b_2 \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters and where f is a nonlinear activation function.

Finally, we define the output operator $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$ as a local linear operator applied pointwise, so that

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) = W_{\text{out}} v(x), \tag{4.19}$$

for $x \in D$, $v \in \mathcal{V}$ and $W_{\text{out}} \in \mathbb{R}^{d_z \times d_{\text{model}}}$.

The composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the transformer neural operator. In Appendix 4.B.1 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

4.4.3 Vision Transformer Neural Operator

We introduce an analogue of the vision transformer architecture as a neural operator using the patching in the continuum framework developed in Section 4.3. We note

that in the context of the ViT from Dosovitskiy et al. (2021), performing patch "flattening" before a local lifting to the embedding dimension introduces a nonlocal transformation on the patches, as described in Remark 4.3.3. This procedure as implemented in Dosovitskiy et al. (2021) violates the desirable (for neural operators) discretization invariance property. Hence, for a neural operator analogue, we substitute the lifting to embedding dimension with a nonlocal integral operator. The ViT neural operator (ViT NO) introduced is thus discretization invariant. We describe this methodology within the framework provided by making explicit choices for $\mathsf{T}_{\text{in}}, \mathsf{T}_{\text{out}}, \mathsf{E}_L$ that instantiate Equations (4.9) and (4.10) appropriately. In Figure 4.2 we provide a schematic for the architecture, with a graphical representation of the encoder layer in Figure 4.3.

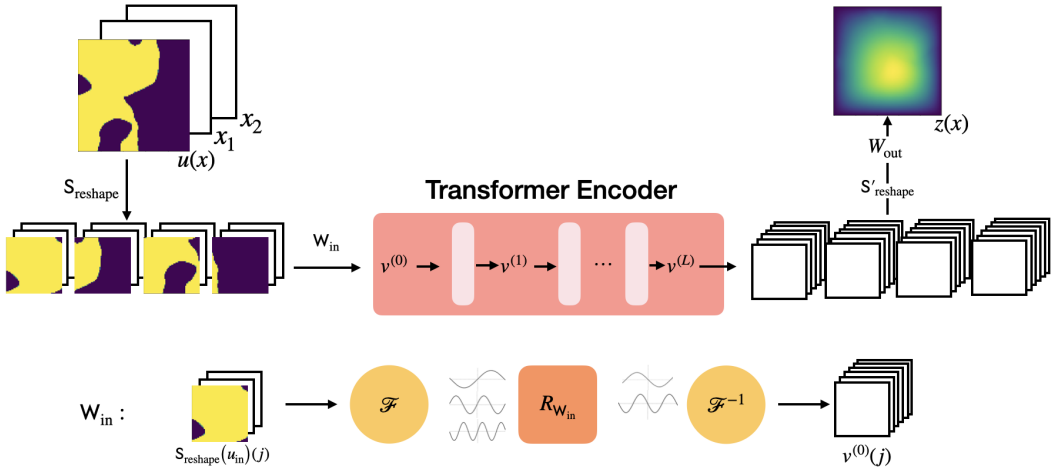


Figure 4.2: Vision Transformer Neural Operator.

Let $D \subset \mathbb{R}^d$ be a bounded open set. Consider $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ and let $D := D_1 \cup \cdots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ have equivalent finite volume for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We consider model inputs $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \{v \mid v : [\![1, P]\!] \to \mathcal{U}(D';\mathbb{R}^{d_{\text{model}}})\}$, which is a space of functions acting on patch indices, and model output space $\mathcal{Z}(D;\mathbb{R}^{d_z})$, with domain $D \subset \mathbb{R}^d$.

We begin by defining the input function $\mathsf{T}_{\text{in}} : \mathcal{U}(D;\mathbb{R}^{d_u}) \to \mathcal{V}$. The function $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ is first concatenated with the identity defined on the domain D, so that the resulting function $u_{\text{in}} \in \mathcal{U}(D;\mathbb{R}^{d_u+d})$ is defined as in (4.14); recall that this defines a positional encoding. An operator Σ_{reshape} is applied to the function $u_{\text{in}} \in \mathcal{U}(D;\mathbb{R}^{d_u+d})$ that acts on the input so that the resulting function is of the form

$$\Sigma_{\text{reshape}}(u_{\text{in}}) : [\![1, P]\!] \to \mathcal{U}(D', \mathbb{R}^{d_u+d}). \tag{4.20}$$

We apply a nonlocal linear operator $\mathsf{W}_{\text{in}} \in \mathcal{L}\Big(\mathcal{U}(D';\mathbb{R}^{d_u+d})\,;\,\mathcal{U}\big(D';\mathbb{R}^{d_{\text{model}}}\big)\Big)$ that acts on any $u : [\![1,P]\!] \to \mathcal{U}(D';\mathbb{R}^{d_u+d})$ so that for any $j \in [\![1,P]\!]$

$$u(j) \mapsto \mathsf{W}_{\text{in}} u(j). \tag{4.21}$$

The above nonlocal linear operator is chosen as an integral operator as in Definition 4.4.2 and the subsequent discussion. The composition of the operators defined in Equations (4.14), (4.20) and (4.21) defines our choice of T_{in}. Namely, we define $\mathsf{T}_{\text{in}} : \mathcal{U}(D;\mathbb{R}^{d_u}) \to \mathcal{V}$ via

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(j) \coloneqq \mathsf{W}_{\text{in}}\Big(\Sigma_{\text{reshape}}\big(u_{\text{in}}\big)(j)\Big), \tag{4.22}$$

where $u_{\text{in}} \in \mathcal{U}(D;\mathbb{R}^{d_u+d})$ is defined as in (4.14), $j \in [\![1,P]\!]$ and W_{in} is defined as in (4.21).

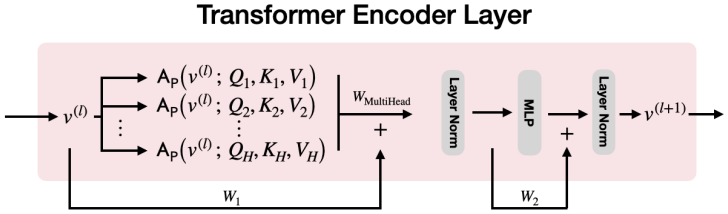


Figure 4.3: Vision Transformer Neural Operator Encoder Layer.

Next, we define the details of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in Equation (4.10) by specifying a definition of $\mathsf{A}_{\text{MultiHead}}$, $\mathsf{F}_{\text{LayerNorm}}$, $\mathsf{W}_1, \mathsf{W}_2$, and F_{NN} in this patched setting. In each layer the operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (4.11) based on the self-attention from Definition 4.3.2. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable local (pointwise) linear operators $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ for $h = 1,\dots,H$. In this setting, these are chosen to be linear operators applied pointwise, they can be represented by finite dimensional linear transformations $Q_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K \times d_{\text{model}}}$, for $h = 1,\dots,H$. [6] For implementation purposes d_K is chosen as $d_K \coloneqq d_{\text{model}}/H$. [7] The multi-head attention operator is thus defined by the application of a local linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to the concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators

[6] We note that in the context of the Fourier attention neural operator, presented in the next Subsection 4.4.4, we generalize this definition to $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ for $h = 1,\dots,H$ being nonlocal linear integral operators.

[7] We note that d_{model} and H should be chosen so that d_{model} is a multiple of H.

$\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$ are pointwise linear operators and hence admit finite-dimensional representations $W_1, W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, so that they define a map

$$v(x; p) \mapsto W_1 v(x; p), \tag{4.23}$$

for each $x \in D'$ and $p \in [\![1, P]\!]$, and similarly for W_2. The operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\Big(\mathsf{F}_{\text{LayerNorm}}(v; \gamma, \beta)(x; p)\Big)_k = \gamma_k \cdot \frac{\big(v(x; p)\big)_k - m\big(v(x; p)\big)}{\sqrt{\sigma^2\big(v(x; p)\big) + \varepsilon}} + \beta_k, \tag{4.24}$$

for $k = 1, \ldots, d_{\text{model}}$, any $p \in [\![1, P]\!]$ and any $x \in \mathbb{R}^d \subset D$, where we use the notation $(\,\cdot\,)_k$ to denote the k'th entry of the vector. In equation (4.24), $\varepsilon \in \mathbb{R}^+$ is a fixed parameter, $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters and m, σ are defined as in (4.17). The operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined by

$$\mathsf{F}_{\text{NN}}(v; W_3, W_4, b_1, b_2)(x; p) = W_3 f\big(W_4 v(x; p) + b_1\big) + b_2, \tag{4.25}$$

for any $x \in D$ and any $p \in [\![1, P]\!]$, where $W_3, W_4 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $b_1, b_2 \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters and f is a nonlinear activation function.

Finally, we define the action of $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$. This is given by first applying a reshaping operator Σ'_{reshape} to the output of the encoder, where this map is defined as

$$v \mapsto \Sigma'_{\text{reshape}}\big(v\big) \in \mathcal{U}(D; \mathbb{R}^{d_{\text{model}}}). \tag{4.26}$$

We note that the operator Σ'_{reshape} may be viewed as an inverse of Σ_{reshape}. A pointwise linear transformation $W_{\text{out}} \in \mathbb{R}^{d_z \times d_{\text{model}}}$ defined by

$$v(x) \mapsto W_{\text{out}} v(x) \in \mathcal{Z}(D; \mathbb{R}^{d_z}), \tag{4.27}$$

for any $x \in D$, is then applied. This procedure completes the definition of the operator $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$, which is given by

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) \coloneqq W_{\text{out}}\Big(\Sigma'_{\text{reshape}}\big(v\big)(x)\Big), \tag{4.28}$$

for any $v \in \mathcal{V}$ and $x \in D$.

Remark 4.4.3. *Discontinuities may arise when the patching architecture is used and may be visible in error plots and can also interfere with self-composition of maps learned as solution operators of time-dependent PDEs. Ameliorating such discontinuities can be achieved using a smoothing operator as a final layer of the*

network. To this end, assuming $\mathcal{Z}(D;\mathbb{R}^{d_z}) \subset H^s(D;\mathbb{R}^{d_z})$ *for some* $s \geqslant 0$*, we define for* $\varepsilon > 0$ *and* $\alpha > 1$ *the operator* $(I - \varepsilon\Delta)^{-\alpha} : \mathcal{Z}(D;\mathbb{R}^{d_z}) \to H^{s+2\alpha}(D;\mathbb{R}^{d_z})$*, and define the last layer of the architecture as*

$$z \mapsto (I - \varepsilon\Delta)^{-\alpha}(z), \tag{4.29}$$

for $z := \left(\mathsf{T}_{\text{out}} \circ \mathsf{E}_L \circ \mathsf{T}_{\text{in}}\right)(u)$.

The composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the ViT neural operator. In Appendix 4.B.2 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

4.4.4 Fourier Attention Neural Operator

In this subsection, we introduce a different transformer neural operator based on patching, the Fourier attention neural operator (FANO). In this setting, to gain additional expressivity, we replace the local linear operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ in the attention mechanism with nonlocal integral operators. In Figure 4.4 we provide a schematic for the Fourier attention neural operator architecture, with a graphical representation of the encoder layer in Figure 4.5. We outline the details of the architecture in the following discussion.

We again let $D \subseteq \mathbb{R}^d$ be a bounded open set. Consider $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ and let $D := D_1 \cup \ldots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We consider model inputs $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \left\{v \mid v : [\![1, P]\!] \to \mathcal{U}(D';\mathbb{R}^{d_{\text{model}}})\right\}$ which is a space of functions acting on patch indices , and model output space $\mathcal{Z}(D;\mathbb{R}^{d_z})$, with domain $D \subset \mathbb{R}^d$.

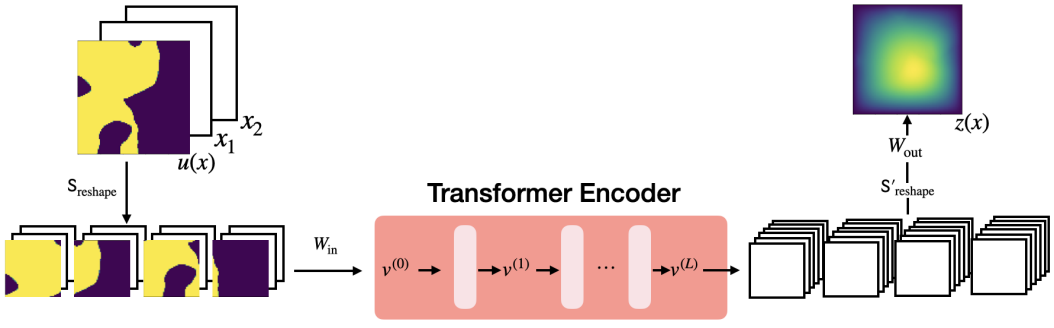


Figure 4.4: Fourier Attention Neural Operator.

We define T_{in} similarly to (4.22) in the previous section, via a composition of Equations (4.14), (4.20) and (4.21); however here the linear lifting operator is chosen to be a pointwise linear transformation $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u + d)}$. Namely, we define $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$ via

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(j)(x) \coloneqq W_{\text{in}}\Big(\Sigma_{\text{reshape}}\big(u_{\text{in}}\big)(j)\Big)(x), \tag{4.30}$$

for any $j \in [\![1, P]\!]$ and any $x \in D$, where $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u + d})$ is defined as in (4.14), Σ_{reshape} as in (4.20) and the nonlocal operation (4.21) is replaced by a local pointwise one.

Next, we define the particulars of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in Equation (4.10). The key distinction between the Fourier attention neural operator that we define here and the ViT neural operator from the previous subsection 4.4.3 lies in the nonlocality of operations performed within $\mathsf{A}_{\text{MultiHead}}$. Here, in each layer the operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (4.11) based on the self-attention operator from Definition 4.3.2. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable nonlocal linear integral operators $\mathsf{Q}_h, \mathsf{K}_h \in \mathcal{L}\Big(\mathcal{U}(D', \mathbb{R}^{d_{\text{model}}}); L^2(D', \mathbb{R}^{d_K})\Big)$ and $\mathsf{V}_h \in \mathcal{L}\Big(\mathcal{U}(D', \mathbb{R}^{d_{\text{model}}}); \mathcal{U}(D', \mathbb{R}^{d_K})\Big)$, for $h = 1, \ldots, H$, where for implementation purposes d_K is chosen as $d_K \coloneqq d_{\text{model}}/H$. The specific formulation of these linear integral operators may be found in Definition 4.4.2 and the subsequent discussion. The multi-head attention operator is thus defined by the application of a pointwise linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to a concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators $\mathsf{W}_1, \mathsf{W}_2$ are defined as in (4.23). The layer normalization operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined as in (4.24) so that it is applied to every point in every patch. Furthermore, the operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined as in (4.25).

We define T_{out} identically to Equation (4.28). We refer to Remark 4.4.3 for an additional operation that can be applied to ameliorate the effect of patch discontinuities. Finally, the composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the Fourier attention neural operator. In Appendix 4.B.3 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

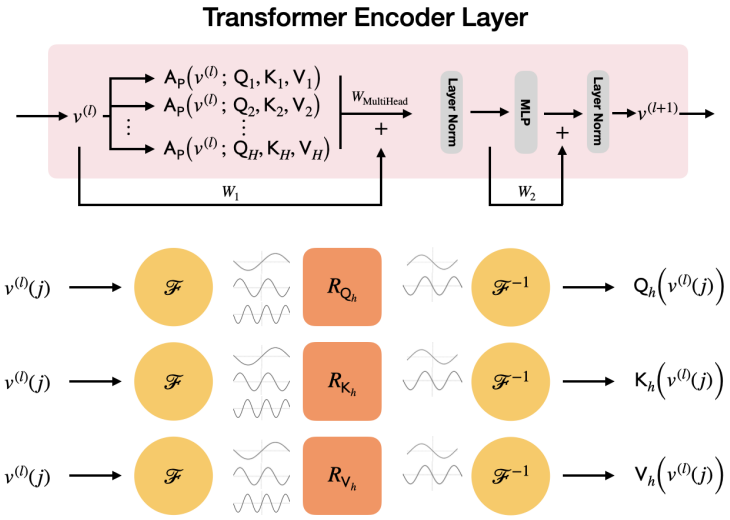


Figure 4.5: Encoder Layer of the Fourier Attention Neural Operator.

4.5 Universal Approximation by Transformer Neural Operators

Universal approximation is a minimal requirement for neural operators. When the operator to be approximated maps between spaces of functions defined over Euclidean domains, universal approximation necessarily requires both nonlocality and nonlinearity. In the recent paper Lanthaler, Zongyi Li, and Andrew M. Stuart (2025), a minimal architecture possessing these two properties is exhibited. In this section, rather than trying to prove universal approximation for the various discretization-invariant neural operators introduced in the chapter, we construct a simple canonical setting that exhibits the universality of the attention mechanism on function space, allowing direct exploitation of the ideas in Lanthaler, Zongyi Li, and Andrew M. Stuart (2025). We build on the notation used in Subsection 4.4.1. In particular A denotes the self-attention operator from Definition 4.2.5 and f an activation function. Throughout this section, we consider activation functions $f \in C^\infty(\mathbb{R})$ which are non-polynomial and Lipschitz continuous.

To be specific, we consider neural operators $\Psi : \mathcal{U}\big(D;\mathbb{R}^{d_u}\big) \times \Theta \to \mathcal{Z}\big(D;\mathbb{R}^{d_z}\big)$ of the form

$$\Psi(u;\theta) := \Big(\mathsf{T}_{\text{out}} \circ \mathsf{E} \circ \mathsf{T}_{\text{in}}\Big)(u;\theta), \tag{4.31}$$

where $\mathsf{T}_{\text{in}} : \mathcal{U}\big(D;\mathbb{R}^{d_u}\big) \to \mathcal{V}\big(D;\mathbb{R}^{d_{\text{model}}}\big)$ and $\mathsf{T}_{\text{out}} : \mathcal{V}\big(D;\mathbb{R}^{d_{\text{model}}}\big) \to \mathcal{Z}\big(D;\mathbb{R}^{d_z}\big)$ are defined by neural networks of the form

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(x) = R_2 f\Big(R_1\big(u(x),x\big) + b_R\Big) + b'_R, \tag{4.32}$$

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) = P_2 f\Big(P_1\big(v(x),x\big) + b_P\Big) + b'_P, \tag{4.33}$$

where $\big(u(x), x\big) \in \mathbb{R}^{d_u+d}$ and $\big(v(x), x\big) \in \mathbb{R}^{d_{\text{model}}+d}$, where R_1, R_2, P_1, P_2 are learned linear transformations of appropriate dimensions and b_R, b'_R, b_P, b'_P are learned vectors. We define the operator $\mathsf{E} : \mathcal{V}\big(D; \mathbb{R}^{d_{\text{model}}}\big) \to \mathcal{V}\big(D; \mathbb{R}^{d_{\text{model}}}\big)$ as a variant of the layer defined by the iteration step in Equation (4.10), given by the two-step map acting on its inputs $v \in \mathcal{V}$ as

$$v(x) \leftarrow\!\shortmid\; W_1 v(x) + \mathsf{A}\big(v; Q, K, V\big)(x), \tag{4.34a}$$

$$v(x) \leftarrow\!\shortmid\; W_2 v(x) + W_3 f\big(W_4 v(x) + b_1\big) + b_2, \tag{4.34b}$$

for any $x \in D$. Note that the neural operator thus defined does not include layer normalizations; it is thus a variant of the transformer neural operator from Subsection 4.4.2. We may now apply the result of Lanthaler, Zongyi Li, and Andrew M. Stuart (2025) to show two universal approximation theorems for the resulting transformer neural operator.

Theorem 4.5.1. *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary, and fix integers $s, s' \geqslant 0$. If $\Psi^\dagger : C^s\big(\overline{D}; \mathbb{R}^r\big) \to C^{s'}\big(\overline{D}; \mathbb{R}^{r'}\big)$ is a continuous operator and $K \subset C^s\big(\overline{D}; \mathbb{R}^r\big)$ a compact set, then for any $\varepsilon > 0$, there exists a transformer neural operator $\Psi(\cdot; \theta) : K \subset C^s\big(\overline{D}; \mathbb{R}^r\big) \to C^{s'}\big(\overline{D}; \mathbb{R}^{r'}\big)$ so that*

$$\sup_{u \in K} \big\| \Psi^\dagger(u) - \Psi(u; \theta) \big\|_{C^{s'}} \leqslant \varepsilon. \tag{4.35}$$

Proof. We begin by noting that for $Q, K = 0$ and $V = I$, the self-attention mapping reduces to

$$\mathsf{A}\big(v; Q, K, V\big)(\cdot) = \mathbb{E}_{y \sim p(y; v, x)}[Vv(\cdot)] = \frac{1}{|D|} \int v(x)\, \mathrm{d}x. \tag{4.36}$$

For weights $W_2 = 0$, $W_3 = W_4 = I$ and $b_2 = 0$, the transformer encoder neural operator layer (4.34) reduces to the mapping

$$v(\cdot) \mapsto f\left(W_1 v(\cdot) + b_1 + \frac{1}{|D|} \int v(x)\, \mathrm{d}x\right). \tag{4.37}$$

The existence of $W_1, R_1, R_2, P_1, P_2, b_1, b_R, b'_R, b_P, b'_P$ so that $\Psi(\cdot\,; \theta)$ satisfies (4.35) then follows from Lanthaler, Zongyi Li, and Andrew M. Stuart (2025, Theorem 2.1), which also involves the application of the universality result for two-layer neural networks of Pinkus (1999, Theorem 4.1). □

The analysis in Lanthaler, Zongyi Li, and Andrew M. Stuart (2025) allows the derivation of an analogous universal approximation theorem for functions belonging

to Sobolev spaces. This generalization is the content of the next theorem. The proof follows easily by applying the same argument as in the proof of Theorem 4.5.1 and the result of Lanthaler, Zongyi Li, and Andrew M. Stuart (2025, Theorem 2.2).

Theorem 4.5.2. *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary and fix integers $s, s' \geqslant 0$, $p, p' \in [1, \infty)$. If $\Psi^\dagger : W^{s,p}\big(D; \mathbb{R}^r\big) \to W^{s',p'}\big(D; \mathbb{R}^{r'}\big)$ is a continuous operator and $K \subset W^{s,p}\big(D; \mathbb{R}^r\big)$ a compact set of bounded functions so that $\sup_{u \in K} \|u\|_{L^\infty} < \infty$, then for any $\varepsilon > 0$, there exists a transformer neural operator $\Psi(\cdot; \theta) : K \subset W^{s,p}\big(D; \mathbb{R}^r\big) \to W^{s',p'}\big(D; \mathbb{R}^{r'}\big)$ so that*

$$\sup_{u \in K} \big\| \Psi^\dagger(u) - \Psi(u; \theta) \big\|_{W^{s',p'}} \leqslant \varepsilon. \tag{4.38}$$

4.6 Numerical Experiments

In this section we illustrate, through numerical experiments, the capability of the transformer neural operator architectures described in Section 4.4. Throughout, we consider the supervised learning problem described by the data model in (4.3) and take the viewpoint of surrogate modeling to construct approximations of the operators $\Psi^\dagger$. In Subsection 4.6.1 we consider problems given by dynamical systems and ordinary differential equations, namely operators acting on functions defined on a one-dimensional time domain. In this context, we explore the use of the transformer neural operator for operator learning problems given by the Lorenz '63 dynamical system and a controlled ODE. On the other hand, in Subsection 4.6.2 we consider problems given by partial differential equations equations, namely operators acting on functions defined on a two-dimensional spatial domain. In this context we explore the use of the transformer neural operator and of the more efficient ViT neural operator and Fourier attention neural operator architectures for operator learning problems given by the Darcy flow and Navier-Stokes equations. The code repository for the experiments in this chapter may be found at https://github.com/EdoardoCalvello/TransformerNeuralOperators.

Remark 4.6.1 (Relative Loss). *For each neural operator architecture we optimize for the parameters $\theta \in \Theta$ according to the relative loss, i.e.*

$$\inf_{\theta \in \Theta} \frac{1}{J} \sum_{j=1}^{J} \left(\frac{\|\Psi\big(u^{(j)}; \theta\big) - \Psi^\dagger\big(u^{(j)}\big)\|_{\mathcal{Z}}}{\|\Psi^\dagger\big(u^{(j)}\big)\|_{\mathcal{Z}}} \right), \tag{4.39}$$

for data points $\{u^{(j)}\}_{j=1}^J$.

4.6.1 1D Time Domain

In this subsection we consider problems where attention is applied to functions in the one-dimensional time domain. The first setting we consider is that of the Lorenz '63 dynamical system (Edward N Lorenz, 1963a), a simplified model for atmospheric convection. The dynamics are given by

$$\begin{aligned} dx &= \sigma(y - x)dt, \\ dy &= \big(x(\rho - z) - y\big)dt, \\ dz &= \big(xy - \beta z\big)dt, \end{aligned} \tag{Lorenz '63}$$

with parameters set to $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. We consider learning the operator $\Psi^\dagger : C\big([0,T];\mathbb{R}^{d_x}\big) \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_z} \to C\big([0,T];\mathbb{R}^{d_y}\big) \times C\big([0,T];\mathbb{R}^{d_z}\big)$ given by

$$\Psi^\dagger : \{x(t), y(0), z(0)\}_{t\in[0,T]} \mapsto \{y(t), z(t)\}_{t\in[0,T]}. \tag{4.40}$$

In this case, $\Psi^\dagger$ is known to exist and given by the solution operator for the system of linear ODEs arising from the second and third components of (Lorenz '63), given known first component. We note that including the initial condition of the unobserved trajectories in the model introduces a discrepancy in dimension within the input function to the model. Namely, denoting by u the input to the model, $u(0) \in \mathbb{R}^3$ while $u(t) \in \mathbb{R}$ for $t \neq 0$. We overcome this issue by learning separate embedding linear transformations W_{in_0} and W_{in}, for the initial condition and the rest of the trajectory, respectively. In Figure 4.1 we display the results obtained by applying a transformer neural operator model on a test data set of 1000 samples with the same discretization ($\Delta t = 0.01$ and $T = 2$) as the training set of 8000 samples on the operator learning problem described by (4.40). The operator is successfully learned, and displays median relative L^2 error of less than 1%, with worst case rising to just below 10%.

We also experiment with a setting for which $\Psi^\dagger$ is not well-defined: we study the recovery of unobserved trajectory $\{y(t)\}_{t\in[0,T]}$ from the observed trajectory $\{x(t)\}_{t\in[0,T]}$ so that $\Psi^\dagger : C\big([0,T];\mathbb{R}^{d_x}\big) \to C\big([0,T];\mathbb{R}^{d_y}\big)$ and

$$\Psi^\dagger : \{x(t)\}_{t\in[0,T]} \mapsto \{y(t)\}_{t\in[0,T]}.$$

This operator would only be well-defined if $(y(0), z(0))$ were included as inputs. We nonetheless expect accurate recovery of the hidden trajectory by the property of synchronization (Pecora and Carroll, 1990; Hayden, Olson, and Titi, 2011).

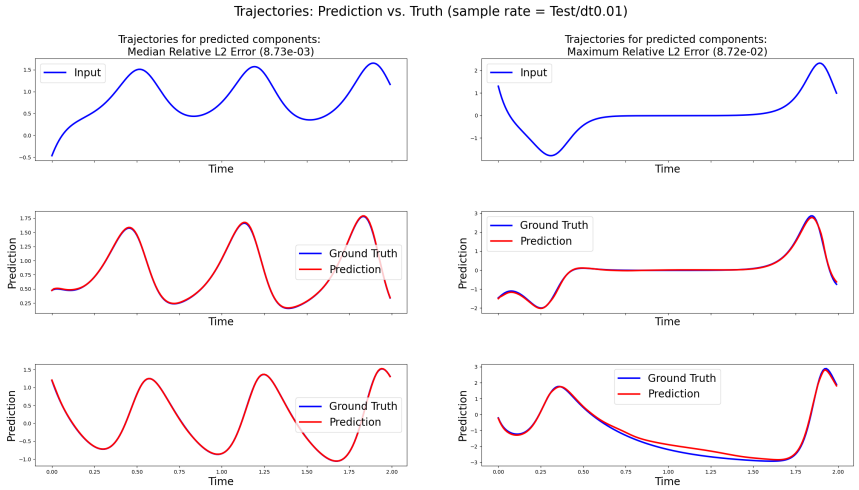


Figure 4.1: The panel displays the performance of the transformer neural operator when applied to the Lorenz '63 operator learning problem of recovering both unobserved y and z trajectories. In each column, the top plot shows the input x trajectory data while the second and third row display the predicted against true y and z trajectories, respectively; the left column concerns the testing example achieving a median relative L^2 error, and the right columns shows the test example achieving the worst error.

We investigate an additional operator learning problem given by the setting of the controlled ODE

$$\begin{aligned} dz &= \sin(z)du, \qquad & z(0) &= z_0, \\ u(t) &= \sum_{j=1}^{J} \xi_j \sin(\pi\eta_j t), \quad & \xi_j &\sim \mathsf{N}(0,1) \text{ i.i.d. }, \eta_j \sim \mathsf{unif}([0,J]) \text{ i.i.d. }. \end{aligned} \tag{CDE}$$

In this case we aim to approximate the operator $\Psi^\dagger : C\big([0,T];\mathbb{R}^{d_u}\big) \to C\big([0,T];\mathbb{R}^{d_z}\big)$ defined by

$$\Psi : \{u(t);\xi,\eta\}_{t\in[0,T]} \mapsto \{z(t);\xi,\eta\}_{t\in[0,T]}.$$

For this problem we make the choice $J = 10$.

The experimental settings we consider for the continuous time operator learning problem may be viewed through the lens of data assimilation as smoothing problems (Sanz-Alonso, A. Stuart, and Taeb, 2023a). Indeed, the mappings to be learned represent the recovery of an unobserved trajectory $\{z(t)\}_{t\in[0,T]}$ conditioned on the observation of a whole trajectory $\{u(t)\}_{t\in[0,T]}$.

In Figure 4.2 we demonstrate the mesh-invariance property of the transformer neural operator in the context of the operator learning problems arising from equations

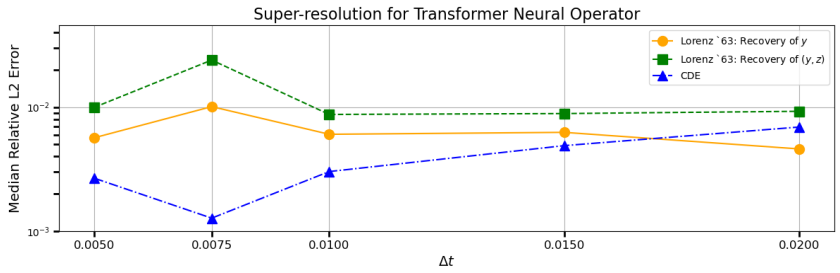


Figure 4.2: The panel displays the performance in relative L^2 errors (in log-scale) of the transformer neural operator when applied to the Lorenz '63 and controlled ODE operator learning test sets of different resolutions at inference time (without retraining). All models were parametrized by $d_{\text{model}} = 128$ and 6 encoder layers, and were trained with 8000 trajectories with $T = 2$, sampled according to $\Delta t = 0.01$.

(Lorenz '63) and (CDE). A key motivation for considering the continuum formulation of attention and the related transformer neural operator is the property of zero-shot generalization to different non-uniform meshes. To demonstrate this, in Figure 4.3 we consider the operator learning problem of mapping the observed x trajectory of (Lorenz '63) to the unobserved y trajectory. We display the median relative L^2 error and worst error samples from a test set consisting of trajectories $\{x(t)\}_{t\in\mathfrak{T}_{\text{test}}}$, where the test time index set $\mathfrak{T}_{\text{test}}$ is defined as

$$\mathfrak{T}_{\text{test}} := \{n\Delta t\}_{n=0}^{N/2} \cup \{2n\Delta t\}_{n=N/2+1}^{3N/4}, \tag{4.41}$$

for $\Delta t := T/N$, where the model deployed is trained on trajectories indexed at a set $\mathfrak{T}_{\text{train}}$ defined by

$$\mathfrak{T}_{\text{train}} := \{n\Delta t\}_{n=0}^{N}. \tag{4.42}$$

In other words, the second half of the testing domain is up-sampled by a factor of 2 compared to the training discretization. As seen qualitatively in Figure 4.3 testing on the non-uniform discretization based on a model trained on the uniform grid is successful. The use of different grids in test and train data does lead to a larger error than the one incurred when they match (Figure 4.2), but the errors remain small. It is our continuum formulation of attention from Subsection 4.B.1 that enables this direct generalization; indeed, as displayed in Figure 4.4, the transformer from Vaswani et al., 2017 when applied to the same setting yields larger errors than the transformer neural operator, due to the TNO's reweighting based on the discretization of the grid.

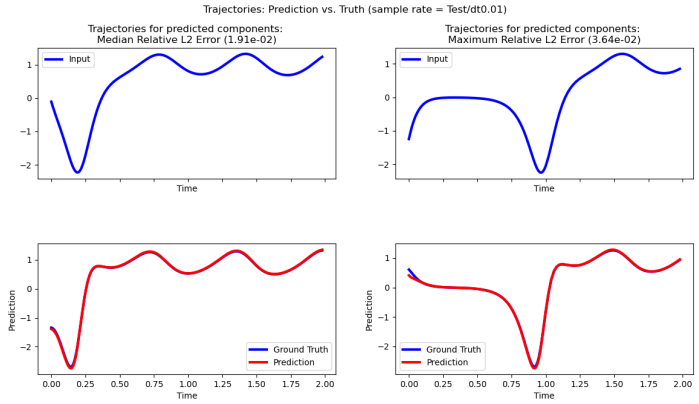


Figure 4.3: The panel displays the performance in relative L^2 errors of the transformer neural operator when applied to the Lorenz '63 operator learning problem of recovering the unobserved y trajectory. The results displayed concern a model trained on trajectories indexed on the index set (4.42) deployed on a test of trajectories indexed on (4.41). In each column, the top plot shows the input data and the bottom compares the prediction and ground truth; the left column shows the testing example achieving a median error, and the right columns shows the test example achieving the worst error.

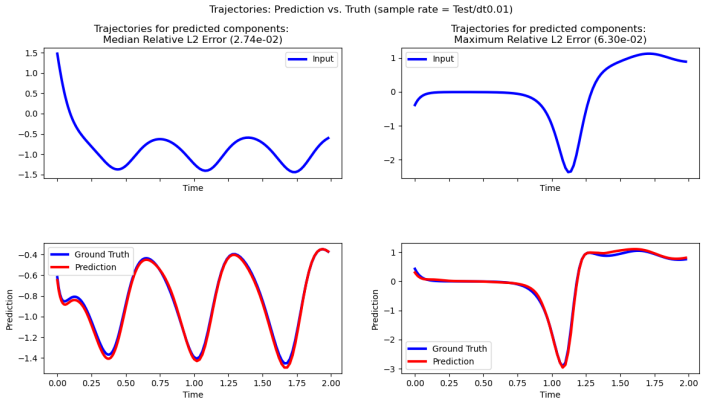


Figure 4.4: The panel displays the performance in relative L^2 errors of the transformer from Vaswani et al., 2017 when applied to the Lorenz '63 operator learning problem of recovering the unobserved y trajectory. The results displayed concern a model trained on trajectories indexed on the index set (4.42) deployed on a test of trajectories indexed on (4.41). In each column, the top plot shows the input data and the bottom compares the prediction and ground truth; the left column shows the testing example achieving a median error, and the right columns shows the test example achieving the worst error.

4.6.2 2D Spatial Domain

In this subsection we study two different 2D operator learning problems — flow in a porous medium, governed by the Darcy model, and incompressible viscous fluid

mechanics governed by the Navier–Stokes equation, in particular, a Kolmogorov flow. We apply the three architectures developed in Sections 4.2 and 4.3, concentrating on the patched architectures. In the first Subsection 4.6.2.1 we report the results obtained by applying the neural operators described to an array of operator learning problems, where the domain of the functions is two-dimensional i.e. $D \subset \mathbb{R}^2$. In Subsection 4.6.2.2 we describe the settings of the two Darcy flow operator learning problems considered, along with the parametrization and training details for the various architectures. Similarly in Subsection 4.6.2.3, we describe the Kolmogorov flow operator learning problem and relevant implementation details. Subsections 4.6.2.2 and 4.6.2.3 also contain further numerical results illustrating the behavior of the proposed transformer neural operators, and discretization invariance in particular.

4.6.2.1 Results

The results presented concern the two variants of Darcy flow, with lognormal and piecewise constant inputs, and the Kolmogorov flow. The operator learning architectures we deploy include the three transformer-based methods introduced in this chapter, and the implementations of the Fourier neural operator from the "NeuralOperator" library (Zongyi Li et al., 2021; N. Kovachki et al., 2023a), the Galerkin transformer from the "galerkin-transformer" library (Cao, 2021), and the AFNO (Guibas et al., 2022) from the "physics-nemo" library (PhysicsNeMo Contributors, 2023). We note that in order to obtain a numerical comparison of the attention mechanisms themselves, in the context of the Galerkin transformer architecture we only use the Galerkin transformer encoder, with lifting and projections achieved by linear layers as in TNO. This allows for a more direct comparison between the Fourier attention from Cao (2021) and our continuum attention, without potential expressivity gain from additional architectural components. In the following discussion, we do not include experiments performed using the Codomain-attention neural operator (Rahman et al., 2024) in the high-resolution settings: the memory usage with the implementation from the "NeuralOperator" library did not allow for comparably sized models, in terms of parameters; the performance of the models we trained was hence incomparably inferior to the others.

Our first results compares the transformer neural operator from Subsection 4.4.2 with FNO, AFNO and the Galerkin transformer (employing Fourier attention) on the lognormal Darcy Flow problem in a low 64×64 resolution setting. Table 4.1

provides a brief summary of the test relative L^2 errors obtained. It is notable that the transformer architecture obtains the best performance with an order of magnitude fewer parameters. However, since higher resolution renders the architecture from Subsection 4.4.2 impractical in dimensions $d \geqslant 2$ the remainder of our experiments use the patched ViT and Fourier attention neural operators from Subsections 4.4.3 and 4.4.4 respectively.

Architecture	Number of Parameters	Darcy Flow Lognormal Input
Transformer NO	$5.98 \cdot 10^5$	$1.19 \cdot 10^{-2}$
FNO	$5.70 \cdot 10^6$	$1.96 \cdot 10^{-2}$
AFNO	$1.10 \cdot 10^6$	$2.57 \cdot 10^{-2}$
Galerkin Transformer	$6.04 \cdot 10^5$	$7.30 \cdot 10^{-2}$

Table 4.1: Median relative L^2 error for each architecture applied to a low resolution experimental setting.

We now consider training two patched architectures on the Darcy Flow problem with piecewise constant inputs and the Kolmogorov Flow problem; in both cases we use a 416×416 resolution setting. We compare cost versus accuracy for the different neural operators implemented, defining cost in terms of both number of parameters and number of FLOPS. In Table 4.2 we present the FLOPS for each method; details of the derivation are provided in Appendix 4.C. We note that the parameter scaling for the transformer neural operator is given by $O\big(d_{\text{model}}^2\big)$, while the scaling for the FNO and the Fourier attention neural operator is $O\big(d_{\text{model}}^2 \cdot k_{\max}\big)$, where $k_{\max}$ is the total number of Fourier modes used. The parameter scaling for the ViT neural operator is given by $O\big(d_{\text{model}} \cdot k_{\max} + d_{\text{model}}^2\big)$. We note that the parameter scaling for the AFNO architecture is given by $O\big(N d_{\text{model}} + d_{\text{model}}^2\big)$, where the $N d_{\text{model}}$ appears because of the learnable positional encoding. Further details are provided in Appendix 4.C.

Figures 4.5 and 4.6 display the cost-accuracy trade-off. Table 4.3 provides the test relative L^2 errors values, depicted in the cost-accuracy analysis figures, obtained with the models with the highest parameter count. Figure 4.5 demonstrates the parameter-efficiency of the ViT neural operator in comparison with the FNO, AFNO and Fourier attention neural operator. Once cost is defined in terms of FLOPS it is apparent that both patched transformer architectures outperform the FNO; furthermore in the Kolmogorov flow problem, the FNO exhibits accuracy-saturation, caused by the finite data set, whereas the patched architectures are less affected, in the ranges we test here. This suggests that the patched transformer architectures use the

Architecture	Scaling
FNO	$O\big(d_{\text{model}} N \log(\sqrt{N}) + N d_{\text{model}}^2\big)$
AFNO	$O\big(d_{\text{model}} N \log(\sqrt{N}) + N d_{\text{model}}^2 / b\big)$
Transformer NO	$O\big(d_{\text{model}} N^2 + N d_{\text{model}}^2\big)$
ViT NO	$O\Big(d_{\text{model}} N \big(P + \log(\sqrt{N/P})\big) + N \log(\sqrt{N}) + N d_{\text{model}}^2\Big)$
FANO	$O\Big(d_{\text{model}} N \big(P + \log(\sqrt{N/P})\big) + N \log(N) + N d_{\text{model}}^2\Big)$

Table 4.2: Evaluation cost (measured in FLOPS) for the three proposed transformer neural operators compared to the Fourier neural operator (Zongyi Li et al., 2021) and adaptive Fourier neural operator (Guibas et al., 2022). We include the scaling of the AFNO as implemented for our use case, i.e. not employing patching and not employing mode truncation; we further note that in the context of this architecture b is a chosen integer hyperparameter. We give further details on the derivation of these scalings in Appendix 4.C.

information content in the data more efficiently. The AFNO is comparable in terms of parameter count to FANO but more costly than ViTNO. On the other hand, even with no patching, the AFNO is cheaper in terms of evaluation complexity. Both ViTNO and FANO outperform the AFNO in terms of parameter-accuracy tradeoff in the context of the Darcy flow problem. For the same problem ViTNO and FANO bring an improvement in accuracy, albeit at a higher computational complexity. In the context of the Kolmogorov flow problem, ViTNO and FANO are competitive with AFNO in parameter-accuracy tradeoff. However for this problem, the AFNO can achieve higher accuracy with a lower complexity. It is important to note that because of the learnable positional embedding, and potential strided convolutions (if using patching), the AFNO is not a neural operator. This means that applying this architecture to different resolutions would require retraining, thus significantly increasing computational complexity. This is a fundamental advantage of the other architectures as their parameters may be deployed across different resolutions for training and inference. We recall the example of GenCast (Price et al., 2025a), where a medium-range weather forecasting model is trained using a neural operator at low resolutions, and then finetuned at higher resolutions, thus drastically reducing the cost for such a large scale application.

4.6.2.2 Darcy Flow

We consider the linear, second-order elliptic PDE defined on the unit square

$$\begin{aligned} -\nabla \cdot \big(a(x)\nabla u(x)\big) &= f(x), \quad x \in (0,1)^2, \\ u(x) &= 0, \qquad x \in \partial(0,1)^2, \end{aligned} \tag{Darcy Flow}$$

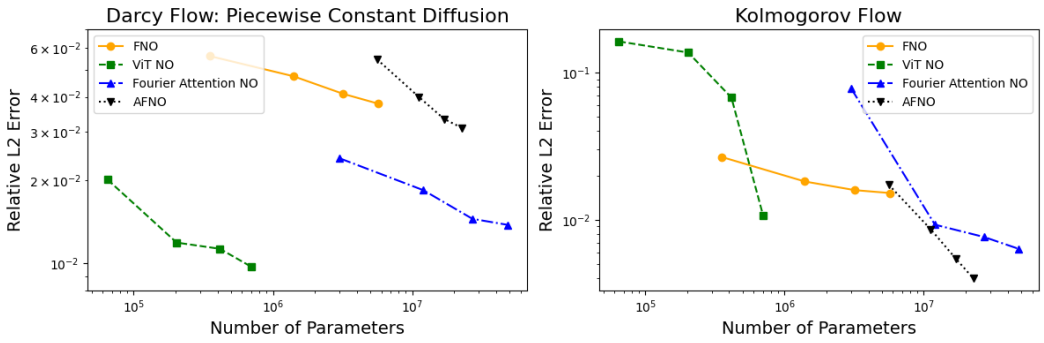


Figure 4.5: The two panels display the test errors (in log-scale) against number of parameters for the architectures with the channel widths of 32, 64, 96, and 128.

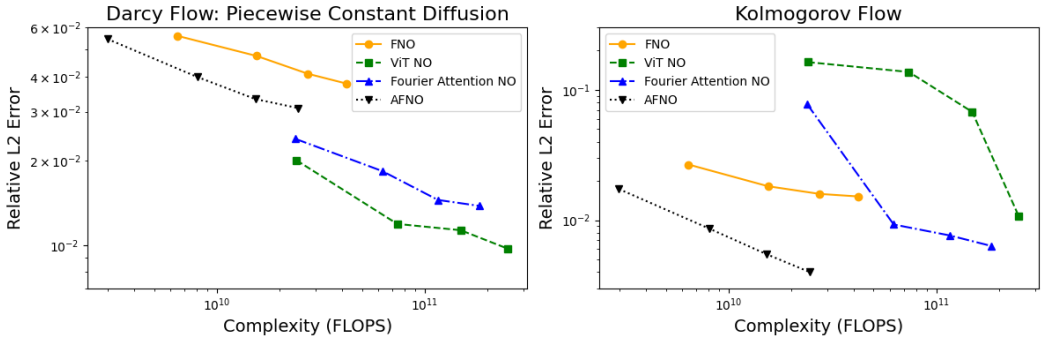


Figure 4.6: The two panels display the test errors (in log-scale) against evaluation cost in FLOPS for the architectures with the channel widths of 32, 64, 96, and 128.

Architecture	Darcy Flow Lognormal Input	Darcy Flow Piecewise Constant Input	Kolmogorov Flow
ViT NO	$7.68 \cdot 10^{-3}$	$9.72 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$
FANO	$8.39 \cdot 10^{-3}$	$1.38 \cdot 10^{-2}$	$6.34 \cdot 10^{-3}$
FNO	$2.38 \cdot 10^{-2}$	$3.78 \cdot 10^{-2}$	$1.52 \cdot 10^{-2}$
AFNO	$2.66 \cdot 10^{-2}$	$3.09 \cdot 10^{-2}$	$3.99 \cdot 10^{-3}$

Table 4.3: Median relative L^2 error for each architecture applied to the different high resolution experimental settings.

which is the steady-state of the 2d Darcy flow equation. The equation is equipped with Dirichlet boundary conditions and is well-posed when the diffusion coefficient $a \in L^\infty\big((0,1)^2;\mathbb{R}_+\big)$ and the forcing function is $f \in L^2\big((0,1)^2;\mathbb{R}\big)$. In this context, we will consider learning the nonlinear operator $\Psi^\dagger : L^\infty\big((0,1)^2;\mathbb{R}_+\big) \to H_0^1\big((0,1)^2;\mathbb{R}\big)$ defined as

$$\Psi^\dagger : a \mapsto u.$$

We consider two different inputs. In both experimental settings the forcing function is chosen as $f \equiv 1$.

In the first the data points $\{a^{(j)}\}_{j=1}^{J}$ are sampled from the probability measure

$$\mu_1 \coloneqq (T_1)_\sharp \mathsf{N}\big(0, C\big),$$

where $\mathsf{N}\big(0, C\big)$ is a Gaussian measure with covariance operator C defined as

$$C = 12^2\big(-\Delta + 6^2 I\big)^{-2},$$

where the Laplacian is equipped with zero Neumann boundary conditions and viewed as acting between spaces of spatially mean-zero functions, and $T_1(\cdot) = \exp(\cdot)$ is the exponential function. We hence refer to the data inputs $a^{(j)}$ as being *lognormal*. In the second setting $\{a^{(j)}\}_{j=1}^{J}$ are sampled from the probability measure

$$\mu_2 \coloneqq (T_2)_\sharp \mathsf{N}\big(0, C\big),$$

where the covariance operator is defined as before and

$$T_2(x) = \begin{cases} 12, & x \geqslant 0, \\ 3, & x < 0. \end{cases}$$

We hence refer to the data inputs $a^{(j)}$ as being *piecewise constant*.

In the context of the lognormal experimental setting, we investigate a scenario with low resolution training samples, for which pointwise attention and hence the transformer neural operator from Subsection 4.4.2 is suitable, and a high resolution scenario where patching becomes necessary for application of the patched-based architectures of Subsections 4.4.3 and 4.4.4.

For the low resolution setting, we train the FNO, AFNO, Galerkin transformer and TNO architectures using 3600 independent samples of resolution 64×64, we use 200 samples for validation and 200 samples for testing. The Fourier neural operator is parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. On the other hand, the transformer neural operator is parametrized by 128 channels in the encoder and is trained using a batch size of 2 and learning rate of 10^{-3}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. The Galerkin transformer uses the Fourier type attention from Cao, 2021, with quadratic complexity, making it prohibitive for higher resolutions. It uses 4 layers of 128 hidden channels and is trained using a learning rate of 10^{-3} and batch size of 2. On this problem we train all neural operators using the relative H^1 loss.

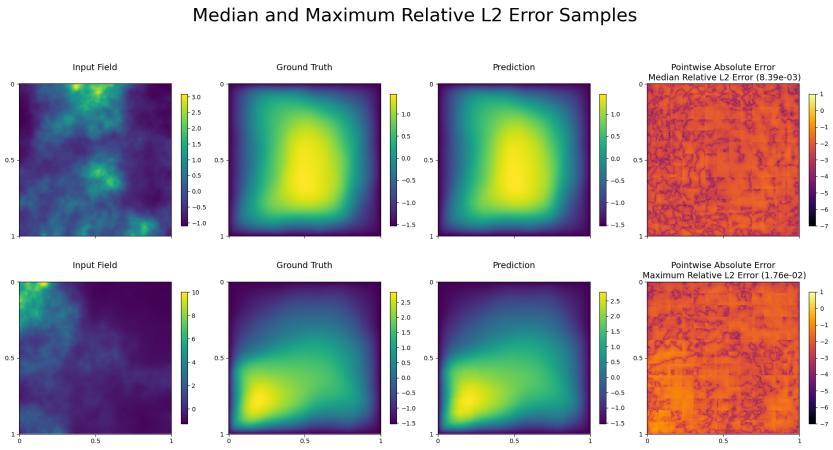


Figure 4.7: The panel displays the result of the application of the Fourier attention neural operator on the Darcy flow experiment with lognormal diffusion for the median and maximum relative L^2 error samples. The first row shows the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. On the other hand, the second row displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input diffusion coefficient $a(x)$ of the relevant sample. The second column shows the true solution $u(x;a)$, while the third column displays the the predicted solution. The last column displays the pointwise absolute error (in log-scale) between the prediction and truth.

For the high resolution setting, we train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures using 3600 independent samples of resolution 416×416; we use 200 samples for validation and 200 samples for testing. The Fourier neural operator is again parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both architectures employ a final spectral convolution layer (see Remark 4.4.3) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 1 and learning rate of 10^{-3}. On this problem we train all the neural operators

with the relative H^1 loss. Figure 4.7 displays the high resolution test results obtained by applying the Fourier attention neural operator.

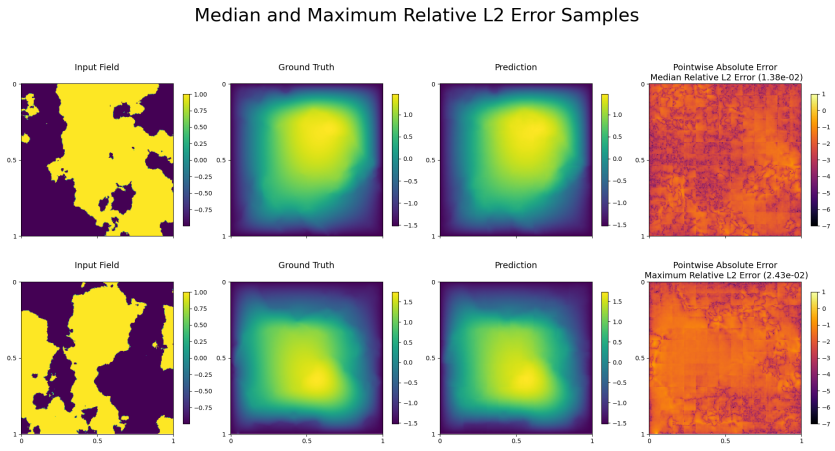


Figure 4.8: The panel displays the result of the application of the Fourier attention neural operator on the Darcy flow experiment with piecewise constant diffusion for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. On the other hand, the second row displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input diffusion coefficient $a(x)$ of the relevant sample. The second column shows the true solution $u(x;a)$, while the third column displays the the predicted solution. The last column displays the pointwise absolute error (in log-scale) between the prediction and truth.

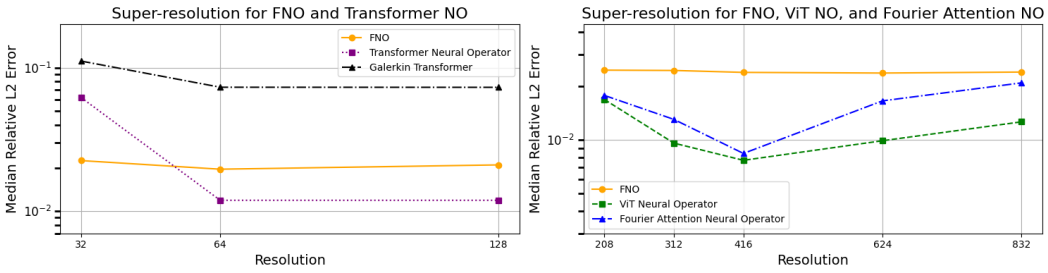


Figure 4.9: The panel displays the performance in relative L^2 errors (in log-scale) of the various architectures when applied to Darcy flow with lognormal diffusion test sets of different resolutions at inference time (without retraining). The left panel concerns FNO, the transformer neural operator and the Galerkin transformer when trained on data of resolution 64×64. On the other hand, the right panel concerns the FNO, ViT neural operator and Fourier attention neural operator architectures trained on data of resolution 416×416.

In the context of the piecewise constant experimental setting, we train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures using 3600 independent samples of resolution 416×416, we use 200 samples for validation and 200 samples for testing. The Fourier neural operator is again parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both architectures employ a final spectral convolution layer (see Remark 4.4.3) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 1 and learning rate of 10^{-3}. On this problem we train all the neural operators with the relative H^1 loss. Figure 4.8 displays the high resolution test results obtained by applying the Fourier attention neural operator.

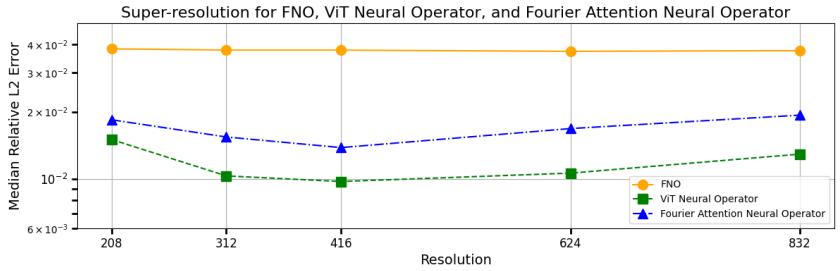


Figure 4.10: The panel displays the performance in relative L^2 errors (in log-scale) of the FNO, ViT neural operator and Fourier attention neural operator when applied to Darcy flow with piecewise constant diffusion test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

In Figures 4.9 and 4.10 we display the results of applying the neural operator architectures for the lognormal and piecewise input settings, respectively, to test samples of different resolutions than the one used for training. The results demonstrate the zero-shot generalization to different resolutions capability of the transformer neural operator architectures, which does not require retraining of the models. In the lognormal setting the FNO exhibits invariance to discretization that is more stable to changing resolution than are the ViT neural operator and the Fourier attention neural operator.

4.6.2.3 Kolmogorov Flow

We consider the two-dimensional Navier-Stokes equation for a viscous, incompressible fluid,

$$\begin{aligned} \frac{\partial u}{\partial t} + u \cdot \nabla u + \nabla p &= \pi \Delta u + \sin(ny)\widehat{x}, && (x,t) \in [0,2\pi]^2 \times (0,\infty), \\ \nabla \cdot u &= 0, && (x,t) \in [0,2\pi]^2 \times [0,\infty), \\ u(\cdot,0) &= u_0, && x \in [0,2\pi]^2, \end{aligned} \tag{KF}$$

where u denotes the velocity, p the pressure, and π the kinematic viscosity. The particular choice of forcing function $\sin(ny)$ leads to a particular example of a Kolmogorov flow. We equip the domain $[0, 2\pi]$ with periodic boundary conditions. We assume

$$u_0 \in \mathcal{U} := \left\{ u \in \dot{L}^2_{\text{per}}\big([0,2\pi]^2; \mathbb{R}^2\big) : \nabla \cdot u = 0 \right\}.$$

The dot on L^2 denotes the space of spatially mean-zero functions. The vorticity is defined as $w = (\nabla \times u)\widehat{z}$ and the stream function f as the solution to the Poisson equation $-\Delta f = w$. Existence of the semigroup $S_t : \mathcal{U} \to \mathcal{U}$ is shown in Temam (2012, Theorem 2.1). We generate the data by solving (KF) in vorticity-streamfunction form by applying the pseudo-spectral split step method from Chandler and Kerswell (2013). In our experimental set-up $n = 4$ and $\pi = 1/70$ are chosen. Random initial conditions are sampled from the Gaussian measure $\mathsf{N}(0, C)$ where the covariance operator is given by

$$C = 7^3\big(-\Delta + 49I\big)^{-5},$$

where the Laplacian is equipped with periodic boundary conditions on $[0, 2\pi]^2$, and viewed as acting between spaces of spatially mean-zero functions. The input output-data pairs for the Kolmogorov flow experiment are given by

$$\left\{ \omega\big(x, T; \omega_0^{(j)}\big); \omega\big(x, T + \Delta t; \omega_0^{(j)}\big) \right\}_{j=1}^{J},$$

for $\omega_0^{(j)} \sim \mathsf{N}(0, C)$ being i.i.d. samples. Indeed, given any $r > 0$ we aim to approximate the nonlinear operator $\Psi^\dagger : H^r\big([0,2\pi]^2; \mathbb{R}\big) \to H^r\big([0,2\pi]^2; \mathbb{R}\big)$ defined by

$$\Psi^\dagger : \omega\big(x, T; \omega_0\big) \mapsto \omega\big(x, T + \Delta t; \omega_0\big).$$

In our experimental setting, we set $T = 11$ and $\Delta t = 0.1$. We train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures on this problem using 9000 independent samples of resolution 416×416, 500 samples for

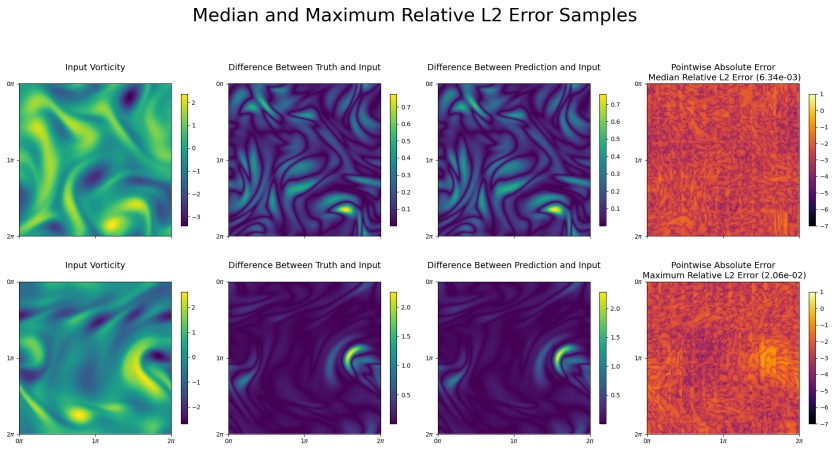


Figure 4.11: The panel displays the result of the application of the Fourier attention neural operator on the Kolmogorov flow experiment for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. The second row on the other hand displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input vorticity $w(x, T)$ of the relevant sample. The second column shows the absolute difference between the true vorticity $w(x, T + \Delta t)$ and the input vorticity $w(x, T)$, while the third column displays the absolute difference between the predicted vorticity at time $T + \Delta t$ and the input $w(x, T)$. The last column shows the pointwise absolute error (in log-scale) between the prediction and truth.

validation, and 500 samples for testing. The Fourier neural operator is parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 4 and learning rate of 10^{-3}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both architectures employ a final spectral convolution layer (Remark 4.4.3) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 2 and learning rate of 10^{-3}. On this problem we train all the neural operators with the relative H^1 loss. Figure 4.11 displays the test results using the Fourier attention neural operators.

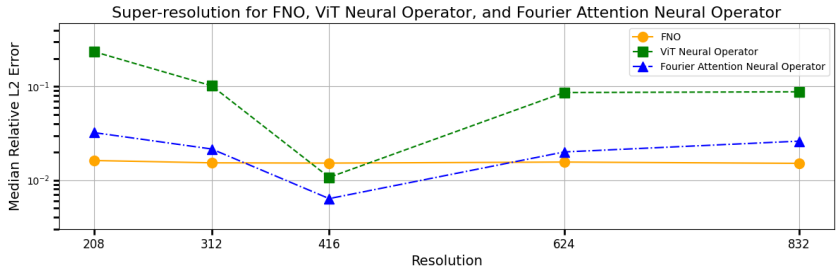


Figure 4.12: The panel displays the performance in relative L^2 errors (in log-scale) of the FNO, ViT neural operator and Fourier attention neural operator when applied to Kolomgorov flow test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

In Figure 4.12 we display the results of applying the three architectures to test samples of resolutions different to the 416×416 training resolution. The results demonstrate the zero-shot generalization capability of the three neural operator architectures to different discretizations, which does not require retraining of the models. Again we note that FNO exhibits invariance to discretization that is more stable to changing resolution than the methods introduced here; but for all the neural operators it is nonetheless clear that intrinsic properties of the continuum limit are learned and may be transferred between discretizations.

A few additional considerations are in order. The mode truncation employed in the Fourier neural operator make the architecture computationally efficient but yields an over-smoothing effect that is not suitable for problems at high resolutions, where high frequency detail is prominent. Attention-based neural operators offer a possible solution to this issue, as demonstrated by the performance of the transformer neural operators proposed when applied to the PDE operator learning problems considered, which involve rough diffusion coefficients and rough initial conditions. On the other hand, patching leads to more efficient attention-based neural operators, but also introduces possible discontinuities at patch-intersections. This issue has been observed to be resolved in the large data regime. Furthermore, it is possible to employ problem specific smoothing operators as the one introduced in Remark 4.4.3. In Figure 4.13 we display the result of applying a smoothing operator as the last layer in the training of the FANO architecture in the context of the Kolmogorov flow problem. The smoothing via the inverse of the negative Laplacian is applied using the Fourier basis, given the periodic boundary conditions. Here, we make the specific choice $\varepsilon = 10^{-3}$ and $\alpha = 1.001$, in the context of Remark 4.4.3. Such an approach is effective at reducing the discontinuities introduced by patching and

reducing the evaluation error, but is problem dependent and requires knowledge of boundary conditions. Such discontinuity issues constitute an inherent limitation of patching and highlights the need for more suitable efficient attention mechanisms.

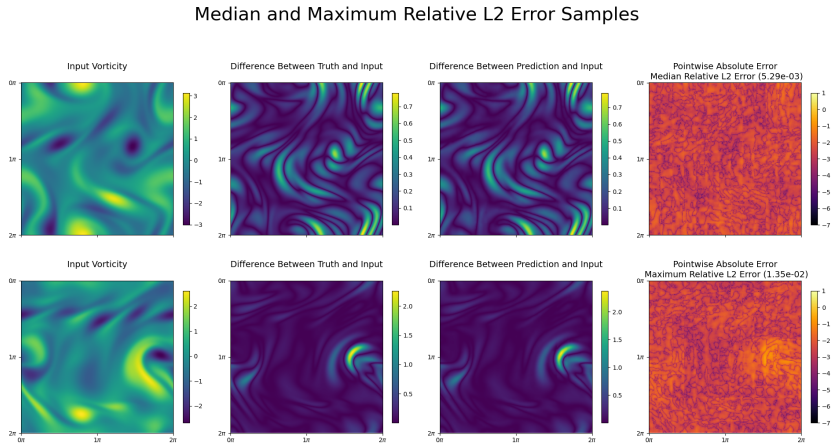


Figure 4.13: The panel displays the result of the application of the Fourier attention neural operator employing a final smoothing layer on the Kolmogorov flow experiment for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. The second row on the other hand displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input vorticity $w(x, T)$ of the relevant sample. The second column shows the absolute difference between the true vorticity $w(x, T + \Delta t)$ and the input vorticity $w(x, T)$, while the third column displays the absolute difference between the predicted vorticity at time $T + \Delta t$ and the input $w(x, T)$. The last column shows the pointwise absolute error (in log-scale) between the prediction and truth.

Furthermore, it is observed that in some experimental settings the patching-based neural operators exhibit worse stability to changing resolution than FNO. This is likely due to the effect of patching which may then be further amplified for FANO due to the application of a higher number of FFT(s) to non-periodic domains. In the context of the FANO architecture, such issues can be ameliorated using periodic extensions for Fourier-based parametrizations. In Figure 4.14 we demonstrate the discretization invariance of the standard FANO compared to a FANO architecture that involves building a periodic extension of each patch. The Fourier integral operator is applied to each extended patch and the extension is then removed from the output. Such a technique leads to an increase in complexity and runtime but yields more stable discretization invariance. Other remedies could potentially include

other kinds of parametrizations for the nonlocal operations, or fine-tuning at different resolutions.

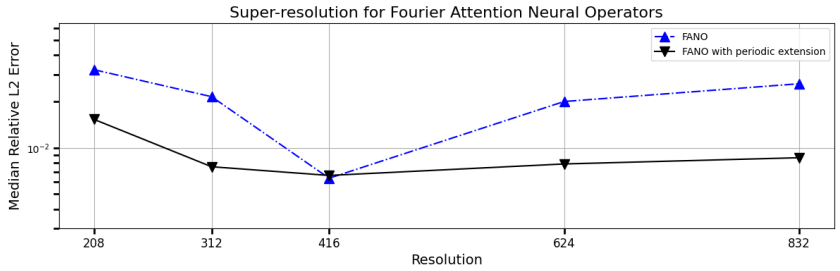


Figure 4.14: The panel displays the performance in relative L^2 errors (in log-scale) of the Fourier attention neural operator and Fourier attention neural operator with periodic extensions of patches when applied to Kolomgorov flow test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

4.7 Conclusions

In this work we have introduced a continuum formulation of the attention mechanism from the seminal work Vaswani et al. (2017). Our continuum formulation can be used to design transformer neural operators. Indeed, this continuum formulation leads to discretization invariant implementations of attention and schemes that exhibit zero-shot generalization to different resolutions. In the first result of its kind for transformers, with a slight modification to the architecture implemented in practice, the resulting neural operator architecture mapping between infinite-dimensional spaces of functions is shown to be a universal approximator of continuous functions and functions of Sobolev regularity defined over a compact domain. We extend the continuum formulation to patched attention, which makes it possible to design efficient transformer neural operators. To this end, we introduce a neural operator analogue of the vision transformer Dosovitskiy et al. (2021) and a more expressive architecture, the Fourier attention neural operator. Through a cost-accuracy analysis, we demonstrate the power of the methodology in the context of a range of operator learning problems. In the following we highlight potential avenues for further work.

1. It is of interest to extend the analysis of Theorems 4.2.6 and 4.2.12 to quantify the norm-dependent rates of convergence of discretized attention to its continuum limit. Furthermore, obtaining the results over i.i.d. samples from arbitrary strictly positive probability distributions would be of interest.

2. The Fourier attention neural operator architecture uses operator parametrizations for $\mathsf{Q}, \mathsf{K}, \mathsf{V}$. Other design choices for these operators not employing the Fourier basis are possible, for example using the wavelet basis as in Tripura and Chakraborty, 2023 or using a multigrid approach as in He, X. Liu, and Xu, 2024. An investigation of other attention neural operator architectures arising from this consideration would be of interest.

3. It is of interest to investigate the discontinuity issues arising from patching. In particular, it is desirable to design schemes that impose continuity amongst these throughout the architecture.

4. Deriving universal approximation theorems for the patch-based transformer neural operators would be of great theoretical interest.

4.A Proofs of Approximation Theorems

4.A.1 Proof of Self-Attention Approximation Theorem

We introduce the following auxiliary result that we will apply to prove Theorem 4.2.6.

Lemma 4.A.1. *Let $D \subset \mathbb{R}^d$ be a bounded open set and let $\{y_j\}_{j=1}^N \sim U(\overline{D})$ be an i.i.d. sequence. If $f \in C(\overline{D} \times \overline{D})$ then, with the expectation taken over the data $\{y_j\}_{j=1}^N$,*

$$\mathbb{E}\left\| \int_D f(\cdot, y)\, \mathrm{d}y - \frac{|D|}{N} \sum_{j=1}^N f(\cdot, y_j) \right\|_{L^2} \leqslant |D|^{3/2} \|f\|_{C(\overline{D}\times\overline{D})} N^{-1/2}.$$

Proof. Notice that, for any $x \in \overline{D}$,

$$\int_D f(x, y)\, \mathrm{d}y = |D| \mathbb{E}_{y\sim U(\overline{D})}\big[f(x, y)\big].$$

Hence, by expanding the square, we can observe that

$$\mathbb{E}\left(\int_D f(x, y)\, \mathrm{d}y - \frac{|D|}{N} \sum_{j=1}^N f(x, y_j) \right)^2 \leqslant \frac{|D|^2}{N} \mathbb{E}_{y\sim U(\overline{D})}\big[f(x, y)^2\big].$$

Bounding f in L^∞, and noting that it is a continuous function, gives

$$\begin{aligned} \mathbb{E}\left\| \int_D f(\cdot, y)\, \mathrm{d}y - \frac{|D|}{N} \sum_{j=1}^N f(\cdot, y_j) \right\|_{L^2}^2 &\leqslant \frac{|D|^2}{N} \int_D \mathbb{E}_{y\sim U(\overline{D})}\big[f(x, y)^2\big]\, \mathrm{d}x \\ &\leqslant |D|^3 \|f\|_{C(\overline{D}\times\overline{D})}^2 N^{-1}. \end{aligned}$$

The result follows by Jensen's inequality. □

As part of the proof of Theorem 4.2.6, we first show that the the self-attention operator is a mapping of the form $\mathsf{A} : L^\infty(D;\mathbb{R}^{d_u}) \to L^\infty(D;\mathbb{R}^{d_V})$ and hence $\mathsf{A} : C(\overline{D};\mathbb{R}^{d_u}) \to C(\overline{D};\mathbb{R}^{d_V})$.

Lemma 4.A.2. *Let $u \in L^\infty(D;\mathbb{R}^{d_u})$, then it holds that $\mathsf{A}(u) \in L^\infty(D;\mathbb{R}^{d_V})$. Furthermore, for $u \in C(\overline{D};\mathbb{R}^{d_u})$ it holds that $\mathsf{A}(u) \in C(\overline{D};\mathbb{R}^{d_V})$.*

Proof. The result follows readily from the fact that p is a probability density function. For completeness, we show the boundedness of the normalization constant of the function p. Indeed we note that

$$\begin{aligned}\int_D \exp\big(\langle Qu(x), Ku(s)\rangle\big)\,\mathrm{d}s &\leqslant \int_D \exp\big(|Q||K||u(x)||u(y)|\big)\mathrm{d}s \\ &\leqslant \int_D \exp\left(\frac{1}{2}|Q|^2|K|^2|u(x)|^2\right)\exp\left(\frac{1}{2}|u(y)|^2\right)\mathrm{d}s \\ &\leqslant |D|\exp\big(R\|u\|_{L^\infty}^2\big),\end{aligned}$$

for some constant $R > 0$. From definition, it is clear that

$$\int_D p(y;u,x)\,\mathrm{d}y = 1;$$

hence p is indeed a valid probability density function. It hence follows that

$$\|A(u)\|_{L^\infty} \leqslant |V|\|u\|_{L^\infty}$$

hence $A : L^\infty(D;\mathbb{R}^{d_u}) \to L^\infty(D;\mathbb{R}^{d_V})$. Similarly, it follows that $A : C(\overline{D};\mathbb{R}^{d_u}) \to C(\overline{D};\mathbb{R}^{d_V})$. We note that the continuity of u is preserved by A under the continuity of the inner product in its first argument and the continuity of the exponential. □

We are now ready to establish the full result of Theorem 4.2.6 which states that

$$\lim_{N\to\infty}\sup_{u\in B}\mathbb{E}\left\|\mathsf{A}(u) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Ku(y_j)\rangle\big)Vu(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Ku(y_\ell)\rangle\big)}\right\|_{C(\overline{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\overline{D})$.

Proof of Theorem 4.2.6. Before commencing the proof, we first establish useful shorthand notation. Letting $u \in B$ we define

$$f(x,y;u) \coloneqq \exp\big(\langle Qu(x), Ku(y)\rangle\big), \quad g_l(x,y;u) \coloneqq \exp\big(\langle Qu(x), Ku(y)\rangle\big)\big(Vu(y)\big)_l$$

for all $x, y \in \overline{D}$ and $l \in [\![1, d_V]\!]$. For $u \in B$ we also define

$$a(x;u) \coloneqq \int_D f(x,y;u)\,\mathrm{d}y, \qquad a^{(N)}(x;u) \coloneqq \frac{|D|}{N}\sum_{j=1}^{N} f(x,y_j;u),$$

and

$$b_l(x;u) \coloneqq \int_D g_l(x,y;u)\,\mathrm{d}y, \qquad b_l^{(N)}(x;u) \coloneqq \frac{|D|}{N}\sum_{j=1}^{N} g_l(x,y_j;u)$$

for any $x \in \overline{D}$ and any $l \in [\![1, d_V]\!]$. We set $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$. Given this notation, to prove the result we must show

$$\lim_{N\to\infty} \sup_{u\in B} \mathbb{E}\left\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\right\|_{C(\overline{D};\mathbb{R})} = 0, \tag{4.43}$$

for any $l \in [\![1, d_V]\!]$. We divide the proof in the following key steps. We first show that for fixed $u \in B$, it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\|_{L^2} = 0, \tag{4.44}$$

for any $l \in [\![1, d_V]\!]$. By using compactness and applying the Arzelà-Ascoli theorem, we then show that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\|_{C(\overline{D})} = 0, \tag{4.45}$$

for any $l \in [\![1, d_V]\!]$. We then use the compactness of B and continuity of the mappings $u \mapsto \alpha_\ell(\cdot\,;u)$ and $u \mapsto \alpha_\ell^{(N)}(\cdot\,;u)$ to deduce the result given by (4.43). We now focus on establishing (4.44). Since B is bounded, there exists a constant $M > 0$ such that

$$\sup_{u\in B} \|u\|_{L^\infty} \leqslant M.$$

Therefore, we have

$$\sup_{u\in B} \max\{\|f\|, \|g_1\|, \ldots, \|g_m\|\} \leqslant \max\left\{\exp\big(RM^2\big), M|V|\exp\big(RM^2\big)\right\} \coloneqq J,$$

where $\|\cdot\| = \|\cdot\|_{C(\overline{D}\times\overline{D})}$. Clearly,

$$\sup_{u\in B} \max\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \leqslant |D|^{3/2} J$$

Notice that, since $|\langle Qu(x), Ku(s)\rangle| \leqslant RM^2$, we have $f(x,y) \geqslant \exp\big(-RM^2\big)$. It follows that

$$\inf_{u\in B} \min\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \geqslant |D|^{3/2}\exp\big(-RM^2\big) \coloneqq |D|^{3/2} I > 0.$$

Similarly,

$$\sup_{u\in B}\max_{l\in[\![1,d_V]\!]}\{\|b_l\|_{L^2},\|b_l^{(N)}\|_{L^2}\}\leqslant |D|^{3/2}J.$$

Applying Lemma 4.A.1, for fixed $u\in B$ we find that

$$\mathbb{E}\left\|\frac{b_l}{a}-\frac{b_l^{(N)}}{a^{(N)}}\right\|_{L^2}\leqslant \mathbb{E}\frac{\|a^{(N)}\|_{L^2}\|b_l-b_l^{(N)}\|_{L^2}+\|b_l^{(N)}\|_{L^2}\|a-a^{(N)}\|_{L^2}}{\|a^{(N)}\|_{L^2}\|a\|_{L^2}} \tag{4.46a}$$

$$\leqslant \frac{2J^2}{I^2}N^{-1/2}. \tag{4.46b}$$

Setting $\alpha_l=b_l/a$ and $\alpha_l^{(N)}=b_l^{(N)}/a^{(N)}$, from (4.46a) we deduce that

$$\lim_{N\to\infty}\mathbb{E}\|\alpha_l(\cdot\,;u)-\alpha_l^{(N)}(\cdot\,;u)\|_{L^2}=0, \tag{4.47}$$

for any $l\in[\![1,d_V]\!]$. We now proceed to the second step of the proof, i.e. showing that (4.45) holds. Using similar reasoning as before, we notice that

$$\inf_{u\in B}\min\{\|a\|_{C(\overline{D})},\|a^{(N)}\|_{C(\overline{D})}\}\geqslant |D|I, \tag{4.48a}$$

$$\sup_{u\in B}\max_{l\in[\![1,d_V]\!]}\{\|b_l\|_{C(\overline{D})},\|b_l^{(N)}\|_{C(\overline{D})}\}\leqslant |D|J; \tag{4.48b}$$

hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly bounded in N. Now, for fixed $u\in B$ the sequence $\{\alpha_l^{(N)}(\cdot\,;u)\}$ is also uniformly equicontinuous with probability 1 over the choice $\{y_j\}_{j=1}^N\sim U(\overline{D})$. Indeed, we note that

$$\left|\frac{b_l^{(N)}(r)}{a^{(N)}(r)}-\frac{b_l^{(N)}(t)}{a^{(N)}(t)}\right|=\left|\frac{\sum_{j=1}^N g_l(r,y_j)}{\sum_{k=1}^N f(r,y_k)}-\frac{\sum_{j=1}^N g_l(t,y_j)}{\sum_{k=1}^N f(t,y_k)}\right| \tag{4.49a}$$

$$\leqslant \frac{1}{N^2I}\left|\sum_{j,n=1}^N g_l(r,y_j)f(t,y_n)-g_l(t,y_j)f(r,y_n)\right| \tag{4.49b}$$

$$\leqslant \frac{J}{N^2I}\sum_{j,n=1}^N\Big(\big|g_l(r,y_j)-g_l(t,y_j)\big|+\big|f(t,y_n)-f(r,y_n)\big|\Big). \tag{4.49c}$$

The result then follows from the uniform continuity of f and g_l in their first arguments, due to the compactness of $\overline{D}$. Therefore, since $\overline{D}\subset\mathbb{R}^d$ is compact, by the Arzelà–Ascoli theorem there exists a subsequence of indices $(N_k)_{k\in\mathbb{N}}$ such that $\alpha_l^{(N_k)}$ converges in $C(\overline{D})$ to some $\alpha_l^{(\infty)}$. Since,

$$\mathbb{E}\|\alpha_l^{(\infty)}-\alpha_l^{(N_k)}\|_{L^2}\leqslant |D|^{1/2}\mathbb{E}\|\alpha_l^{(\infty)}-\alpha_l^{(N_k)}\|_{C(\overline{D})},$$

by uniqueness of limits, it is readily observed that $\alpha_l^{(\infty)} = \alpha_l$. Therefore, since the limit is independent of the subsequence, it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\|_{C(\overline{D})} = 0,$$

for any $l \in [\![1, d_V]\!]$. We now turn our attention to establishing (4.43). By reasoning as before and by using the continuity of the exponential and the inner product in its first argument, it is straightforward to show that as mappings of the form $\alpha_l : B \to C(\overline{D};\mathbb{R}^{d_V})$ and $\alpha_l^{(N)} : B \to C(\overline{D};\mathbb{R}^{d_V})$ where B is compact, α_l and $\alpha_l^{(N)}$ are continuous and hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly equicontinuous. Therefore, we can find moduli of continuity ω_1, ω_2 such that

$$\|\alpha_l(u)-\alpha_l(v)\|_{C(\overline{D})} \leqslant \omega_1\big(\|u-v\|_{C(\overline{D})}\big), \quad \|\alpha_l^{(N)}(u)-\alpha_l^{(N)}(v)\|_{C(\overline{D})} \leqslant \omega_2\big(\|u-v\|_{C(\overline{D})}\big)$$

for any $u, v \in B$ and $N \in \mathbb{N}$. Fix $\varepsilon > 0$. Since B is compact, we can find a number $L = L(\varepsilon) \in \mathbb{N}$ and functions $\phi_1, \ldots, \phi_L \in B$ such that, for any $u \in B$, there exists $j \in [\![1, L]\!]$ such that

$$\|u - \phi_j\|_{C(\overline{D})} < \varepsilon.$$

Furthermore, we can find a number $S = S(\varepsilon) \in \mathbb{N}$ such that, for any $j \in [\![1, L]\!]$, we have, for all $N \geqslant S$,

$$\mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\overline{D})} < \varepsilon.$$

It follows by triangle inequality that

$$\begin{aligned}\mathbb{E}\|\alpha_l(u) - \alpha_l^{(N)}(u)\|_{C(\overline{D})} &\leqslant \mathbb{E}\|\alpha_l(u) - \alpha_l(\phi_j)\|_{C(\overline{D})} + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(u)\|_{C(\overline{D})} \\ &\leqslant |D|\omega_1(\varepsilon) + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\overline{D})} + \mathbb{E}\|\alpha_l^{(N)}(\phi_j) - \alpha_l^{(N)}(u)\|_{C(\overline{D})} \\ &\leqslant |D|\omega_1(\varepsilon) + \varepsilon + |D|\omega_2(\varepsilon).\end{aligned}$$

Therefore the result follows since the argument is uniform over all $l \in [\![1, d_V]\!]$. □

4.A.2 Proof of Cross-Attention Approximation Theorem

We make straightforward modifications to Lemmas 4.A.1, 4.A.4 and to the proof of Theorem 4.2.6 in order to establish Theorem 4.2.12.

Lemma 4.A.3. *Let $D \subset \mathbb{R}^d$ and $E \subset \mathbb{R}^e$ be bounded open sets and $f \in C(\overline{D} \times \overline{E})$. Then*

$$\mathbb{E}\left\|\int_E f(\cdot, y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^{N} f(\cdot, y_j)\right\|_{L^2} \leqslant |D|^{1/2}|E|\|f\|_{C(\overline{D}\times\overline{E})}N^{-1/2}$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim U(\overline{E})$.

Proof. Notice that, for any $x \in \overline{D}$,

$$\int_E f(x,y)\,\mathrm{d}y = |E|\mathbb{E}_{y\sim U(\overline{E})}\big[f(x,y)\big].$$

Hence, by expanding the square, we can observe that

$$\mathbb{E}\left(\int_E f(x,y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^{N} f(x,y_j)\right)^2 \leqslant \frac{|E|^2}{N}\mathbb{E}_{y\sim U(\overline{E})}\big[f(x,y)^2\big].$$

Bounding f in L^∞, and noting that it is a continuous function, gives

$$\begin{aligned}\mathbb{E}\left\|\int_E f(\cdot,y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^{N} f(\cdot,y_j)\right\|_{L^2}^2 &\leqslant \frac{|E|^2}{N}\int_D \mathbb{E}_{y\sim U(\overline{E})}\big[f(x,y)^2\big]\,\mathrm{d}x \\ &\leqslant |D||E|^2\|f\|^2_{C(\overline{D}\times\overline{E})}N^{-1}.\end{aligned}$$

The result follows by Jensen's inequality. □

As part of the proof of Theorem 4.2.12, we first show that the the cross-attention operator is a mapping of the form $\mathsf{C} : L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v}) \to L^\infty(D;\mathbb{R}^{d_V})$ and hence $\mathsf{A} : C(\overline{D};\mathbb{R}^{d_u}) \times C(\overline{E};\mathbb{R}^{d_v}) \to C(\overline{D};\mathbb{R}^{d_V})$.

Lemma 4.A.4. *Let $u \in L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v})$, then it holds that $\mathsf{A}(u) \in L^\infty(D;\mathbb{R}^{d_V})$. Furthermore, for $u \in C(\overline{D};\mathbb{R}^{d_u}) \times C(\overline{E};\mathbb{R}^{d_v})$ it holds that $\mathsf{A}(u) \in C(\overline{D};\mathbb{R}^{d_V})$.*

Proof. The result follows readily from the fact that q is a probability density function. For completeness, we show the boundedness of the normalization constant of the function q. Indeed we note that

$$\begin{aligned}\int_E \exp\big(\langle Qu(x), Kv(s)\rangle\big)\,\mathrm{d}s &\leqslant \int_E \exp\big(|Q||K||u(x)||v(s)|\big)\mathrm{d}s \\ &\leqslant \int_E \exp\left(\frac{1}{2}|Q|^2|K|^2|u(x)|^2\right)\exp\left(\frac{1}{2}|v(s)|^2\right)\mathrm{d}s \\ &\leqslant |E|\exp\big(R_1\|u\|^2_{L^\infty}\big)\exp\big(R_2\|v\|^2_{L^\infty}\big) \\ &\leqslant |E|\exp\Big(R\big(\|u\|^2_{L^\infty} + \|v\|^2_{L^\infty}\big)\Big),\end{aligned}$$

for some constant $R \coloneqq \max\{R_1, R_2\}$. From definition, it is clear that

$$\int_E q(y;u,v,x)\,\mathrm{d}y = 1;$$

hence q is indeed a valid probability density function. It hence follows that

$$\|\mathsf{C}(u,v)\|_{L^\infty(D;\mathbb{R}^{d_V})} \leqslant |V|\|v\|_{L^\infty(E;\mathbb{R}^{d_v})};$$

hence $\mathsf{C} : L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v}) \to L^\infty(D;\mathbb{R}^{d_V})$. Similarly, it follows that $\mathsf{C} : C(\overline{D};\mathbb{R}^{d_u}) \times C(\overline{E};\mathbb{R}^{d_v}) \to C(\overline{D};\mathbb{R}^{d_V})$. We note that the continuity is preserved by C under the continuity of the inner product in its first argument and the continuity of the exponential. □

We are now ready to establish the full result of Theorem 4.2.12 which states that

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E}\left\| \mathsf{C}(u,v) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Kv(y_j)\rangle\big) Vv(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Kv(y_\ell)\rangle\big)} \right\|_{C(\overline{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\overline{E})$.

Proof of Theorem 4.2.12. Before commencing the proof, we first establish useful shorthand notation. Letting $(u,v) \in B$ we define

$$f(x,y;u,v) \coloneqq \exp\big(\langle Qu(x), Kv(y)\rangle\big), \quad g_l(x,y;u,v) \coloneqq \exp\big(\langle Qu(x), Kv(y)\rangle\big)\big(Vv(y)\big)_l$$

for all $x \in \overline{D}$, $y \in \overline{E}$ and $l \in [\![1, d_V]\!]$. For $(u,v) \in B$ we also define

$$a(x;u,v) \coloneqq \int_E f(x,y;u,v)\,\mathrm{d}y, \qquad a^{(N)}(x;u,v) \coloneqq \frac{|E|}{N}\sum_{j=1}^N f(x,y_j;u,v),$$

and

$$b_l(x;u,v) \coloneqq \int_E g_l(x,y;u,v)\,\mathrm{d}y, \qquad b_l^{(N)}(x;u,v) \coloneqq \frac{|E|}{N}\sum_{j=1}^N g_l(x,y_j;u,v)$$

for any $x \in \overline{D}$ and any $l \in [\![1, d_V]\!]$. We set $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$. Given this notation, to prove the result we must show

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E}\left\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\right\|_{C(\overline{D};\mathbb{R})} = 0, \tag{4.50}$$

for any $l \in [\![1, d_V]\!]$. We divide the proof in the following key steps. We first show that for fixed $(u,v) \in B$, it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\|_{L^2} = 0, \tag{4.51}$$

for any $l \in [\![1, d_V]\!]$. By using compactness and applying the Arzelà-Ascoli theorem, we then show that for $(u, v) \in B$

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u, v) - \alpha_l^{(N)}(\cdot\,; u, v)\|_{C(\overline{D})} = 0, \tag{4.52}$$

for any $l \in [\![1, d_V]\!]$. We then use the compactness of B and continuity of the mappings $(u, v) \mapsto \alpha_\ell(\cdot\,; u, v)$ and $(u, v) \mapsto \alpha_\ell^{(N)}(\cdot\,; u, v)$ to deduce the result given by (4.50). We now focus on establishing (4.51). Since B is bounded, there exists a constant $M > 0$ such that

$$\sup_{(u,v)\in B} \|u\|_{L^\infty} + \|v\|_{L^\infty} \leqslant M.$$

Therefore, we have

$$\sup_{(u,v)\in B} \max\{\|f\|, \|g_1\|, \dots, \|g_m\|\} \leqslant \max\left\{\exp\big(RM^2\big), M|V|\exp\big(RM^2\big)\right\} \coloneqq J,$$

where $\|\cdot\| = \|\cdot\|_{C(\overline{D}\times\overline{E})}$. Clearly, it holds that

$$\sup_{(u,v)\in B} \max\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \leqslant |D|^{1/2}|E|J.$$

Notice that, since $|\langle Qu(x), Kv(s)\rangle| \leqslant RM^2$, we have $f(x, y) \geqslant \exp\big(-RM^2\big)$. It follows that

$$\inf_{(u,v)\in B} \min\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \geqslant |D|^{1/2}|E|\exp\big(-RM^2\big) \coloneqq |D|^{1/2}|E|I > 0.$$

Similarly,

$$\sup_{(u,v)\in B} \max_{l\in[d_K]} \{\|b_l\|_{L^2}, \|b_l^{(N)}\|_{L^2}\} \leqslant |D|^{1/2}|E|J.$$

Applying Lemma 4.A.3, we find

$$\mathbb{E}\left\|\frac{b_l}{a} - \frac{b_l^{(N)}}{a^{(N)}}\right\|_{L^2} \leqslant \mathbb{E}\frac{\|a^{(N)}\|_{L^2}\|b_l - b_l^{(N)}\|_{L^2} + \|b_l^{(N)}\|_{L^2}\|a - a^{(N)}\|_{L^2}}{\|a^{(N)}\|_{L^2}\|a\|_{L^2}} \tag{4.53a}$$

$$\leqslant \frac{2J^2}{I^2} N^{-1/2}. \tag{4.53b}$$

Setting $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$, from (4.53a) we deduce that for $(u, v) \in B$ it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u, v) - \alpha_l^{(N)}(\cdot\,; u, v)\|_{L^2} = 0, \tag{4.54}$$

for any $l \in [\![1, d_V]\!]$. We now proceed to the second step of the proof, i.e. showing that (4.52) holds. Using similar reasoning as before, we notice that

$$\inf_{(u,v)\in B} \min\{\|a\|_{C(\overline{D})}, \|a^{(N)}\|_{C(\overline{D})}\} \geqslant |E|I, \tag{4.55a}$$

$$\sup_{(u,v)\in B} \max_{l\in[m]}\{\|b_l\|_{C(\overline{D})}, \|b_l^{(N)}\|_{C(\overline{D})}\} \leqslant |E|J; \tag{4.55b}$$

hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly bounded in N. Now, for fixed $(u, v) \in B$ the sequence $\{\alpha_l^{(N)}(\cdot\,; u, v)\}$ is also uniformly equicontinuous with probability 1 over the choice $\{y_j\}_{j=1}^N \sim U(\overline{D})$. Indeed, we note that

$$\left|\frac{b_l^{(N)}(r)}{a^{(N)}(r)} - \frac{b_l^{(N)}(t)}{a^{(N)}(t)}\right| = \left|\frac{\sum_{j=1}^N g_l(r, y_j)}{\sum_{k=1}^N f(r, y_k)} - \frac{\sum_{j=1}^N g_l(t, y_j)}{\sum_{k=1}^N f(t, y_k)}\right| \tag{4.56a}$$

$$\leqslant \frac{1}{N^2 I}\left|\sum_{j,n} g_l(r, y_j)f(t, y_n) - g_l(t, y_j)f(r, y_n)\right| \tag{4.56b}$$

$$\leqslant \frac{J}{N^2 I}\sum_{j,n}\big(\big|g_l(r, y_j) - g_l(t, y_j)\big| + \big|f(t, y_n) - f(r, y_n)\big|\big). \tag{4.56c}$$

The result then follows from the uniform continuity of f and g_l in their first arguments, due to the compactness of $\overline{D}$. Therefore, since $\overline{D} \subset \mathbb{R}^d$ is compact, by the Arzelà–Ascoli theorem there exists a subsequence of indices $(N_k)_{k\in\mathbb{N}}$ such that $\alpha_l^{(N_k)}$ converges in $C(\overline{D})$ to some $\alpha_l^{(\infty)}$. Since

$$\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{L^2} \leqslant |D|^{1/2}\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{C(\overline{D})},$$

by uniqueness of limits, it is readily observed that $\alpha_l^{(\infty)} = \alpha_l$. Therefore, since the limit is independent of the subsequence, for fixed $(u, v) \in B$ it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u, v) - \alpha_l^{(N)}(\cdot\,; u, v)\|_{C(\overline{D})} = 0,$$

for any $l \in [\![1, d_V]\!]$. We now turn our attention to establishing (4.50). By reasoning as before and by using the continuity of the exponential and of the inner product on the product space, it is straightforward to show that as mappings of the form $\alpha_l : B \to C(\overline{D}; \mathbb{R}^{d_V})$ and $\alpha_l^{(N)} : B \to C(\overline{D}; \mathbb{R}^{d_V})$ where B is compact, α_l and $\alpha_l^{(N)}$ are continuous and hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly equicontinuous. Therefore, we can find moduli of continuity ω_1, ω_2 such that

$$\|\alpha_l(u, v) - \alpha_l(\widetilde{u}, \widetilde{v})\|_{C(\overline{D})} \leqslant \omega_1\big(\|(u, v) - (\widetilde{u}, \widetilde{v})\|_{C(\overline{D})\times C(\overline{E})}\big),$$
$$\|\alpha_l^{(N)}(u, v) - \alpha_l^{(N)}(\widetilde{u}, \widetilde{v})\|_{C(\overline{D})} \leqslant \omega_2\big(\|(u, v) - (\widetilde{u}, \widetilde{v})\|_{C(\overline{D})\times C(\overline{E})}\big)$$

for any $(u, v), (\widetilde{u}, \widetilde{v}) \in B$ and $N \in \mathbb{N}$. Fix $\varepsilon > 0$. Since B is compact, we can find a number $L = L(\varepsilon) \in \mathbb{N}$ and functions $\phi_1, \ldots, \phi_L \in B$ such that, for any $(u, v) \in B$, there exists $j \in [\![1, L]\!]$ such that

$$\|(u, v) - \phi_j\|_{C(\overline{D})\times C(\overline{E})} < \varepsilon.$$

Furthermore, we can find a number $S = S(\varepsilon) \in \mathbb{N}$ such that, for any $j \in [\![1, L]\!]$, we have, for all $N \geqslant S$,

$$\mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\overline{D})} < \varepsilon.$$

It follows by triangle inequality that

$$\begin{aligned}
&\mathbb{E}\|\alpha_l(u, v) - \alpha_l^{(N)}(u, v)\|_{C(\overline{D})} \\
&\qquad \leqslant \mathbb{E}\|\alpha_l(u, v) - \alpha_l(\phi_j)\|_{C(\overline{D})} + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(u, v)\|_{C(\overline{D})} \\
&\qquad \leqslant |D|\omega_1(\varepsilon) + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\overline{D})} + \mathbb{E}\|\alpha_l^{(N)}(\phi_j) - \alpha_l^{(N)}(u, v)\|_{C(\overline{D})} \\
&\qquad \leqslant |E|\omega_1(\varepsilon) + \varepsilon + |E|\omega_2(\varepsilon).
\end{aligned}$$

Therefore the result follows since our argument is uniform over all $l \in [\![1, d_V]\!]$. □

4.B Transformer Neural Operators: The Discrete Setting

We describe how the transformer neural operators described in Section 4.4 are implemented in the discrete setting. Namely, in practice we will have access to evaluations of the input function $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ at a finite set of N discretization points $\{x_i\}_{i=1}^N \subset D$. To highlight how the neural operator methodology is applied in practical settings it is useful, abusing notation, to view the input to the neural operators as $u \in \mathbb{R}^{N\times d_u}$. In order to distinguish between functions in $\mathcal{U}(D; \mathbb{R}^{d_u})$, and discrete representations of these functions at a finite set of grid points, in this section we use different fonts for a function $\mathsf{u} \in \mathcal{U}(D; \mathbb{R}^{d_u})$ and its discrete analogue $u \in \mathbb{R}^{N\times d_u}$. We highlight that in practice, it is of interest to apply approximations of the attention operators A and A_P defined on the continuum, to $u \in \mathbb{R}^{N\times d_u}$. Through an abuse of notation, we will write $\mathsf{A}_\bullet(u)$ to denote the finite-dimensional approximation of $\mathsf{A}_\bullet$ acting on matrices, and similarly for E_L, F_{NN} and F_{LN}.

In Subsection 4.B.1 we describe the transformer neural operator in the discrete setting, in Subsection 4.B.2 the vision transformer neural operator and in Subsection 4.B.3 the Fourier attention neural operator.

4.B.1 Transformer Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.4.2 is implemented in the discrete setting. The input to the model is therefore

a tensor $u \in \mathbb{R}^{N \times d_u}$, which is then concatenated with the discretization points $\{x_i\}_{i=1}^N \subset D$, to form the tensor $u_{\text{in}} \in \mathbb{R}^{N \times (d_u + d)}$. The learnable linear transformation $W_{\text{in}}^\top \in \mathbb{R}^{(d_u+d) \times d_{\text{model}}}$ then acts on u_{in} to yield

$$v^{(0)} \coloneqq u_{\text{in}} W_{\text{in}}^\top,$$

which is then fed through the transformer encoder. In practice we solve the recurrence relation in (4.10) on a finite set of N grid points that arise naturally from the discrete input u_{in}. The multi-head attention operator $\mathsf{A}_{\text{MultiHead}}$ is defined by the application of the linear transformation $W_{\text{MultiHead}}^\top \in \mathbb{R}^{Hd_K \times d_{\text{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{N \times Hd_K}$ tensor, so that

$$\mathsf{A}_{\text{MultiHead}}(v) \coloneqq \big(\mathsf{A}(v; Q_1, K_1, V_1), \dots, \mathsf{A}(v; Q_H, K_H, V_H)\big) W_{\text{MultiHead}}^\top \in \mathbb{R}^{N \times d_{\text{model}}}, \tag{4.57}$$

for any $v \in \mathbb{R}^{N \times d_{\text{model}}}$. We recall that the self-attention operator in question, A, is defined in Definition 4.2.5 for functions. In the discrete setting, where $v \in \mathbb{R}^{N \times d_{\text{model}}}$, we compute the expectation and the integral arising from the normalization constant using the following trapezoidal approximations:

$$\mathsf{A}(v)_j := \frac{1}{2} \sum_{k=2}^{N} \Big(V v_k \cdot p(k; v, j) + V v_{k-1} \cdot p(k-1; v, j) \Big) \cdot \Delta x_k, \tag{4.58}$$

for any $j = 1, \dots, N$, with

$$p(k; v, j) \approx \frac{\exp\Big(\big\langle Q v_j, K v_k \big\rangle_{\mathbb{R}^{d_K}} \Big)}{\frac{1}{2} \sum_{\ell=2}^{N} \Big(\exp\big(\big\langle Q v_j, K v_\ell \big\rangle_{\mathbb{R}^{d_K}} \big) + \exp\big(\big\langle Q v_j, K v_{\ell-1} \big\rangle_{\mathbb{R}^{d_K}} \big) \Big) \Delta x_\ell}, \tag{4.59}$$

where by u_ℓ we have denoted the ℓ'th row of the matrix $v \in \mathbb{R}^{N \times d_{\text{model}}}$, and similarly for $\mathsf{A}(v)_\ell$. In equations (4.58) and (4.59) Δx_k denotes

$$\Delta x_k \coloneqq \prod_{i=1}^{d} |(x_k)_i - (x_{k-1})_i|,$$

for $k = 2, \dots, N$, where $(x_k)_i$ is the i'th component of the vector x_k. This trapezoidal approximation allows for a mathematically consistent implementation of attention for irregularly sampled data, as demonstrated in Section 4.6.

The linear transformations $W_1^\top, W_2^\top \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are applied pointwise. It is also straightforward to apply the definitions for $\mathsf{F}_{\text{LayerNorm}}$ in (4.16) and F_{NN} in (4.18)

to the finite representation of v as both involve pointwise linear transformations. We observe that the recurrence relation (4.10) is compatible with varying domain discretization lengths N, and will produce an output $v^{(L)}$ of dimension matching the input $v^{(0)}$. The resulting tensor $v^{(L)} \in \mathbb{R}^{N \times d_{\text{model}}}$ is then projected to the output dimension by applying the learnable linear transformation $W_{\text{out}}^{\top} \in \mathbb{R}^{d_{\text{model}} \times d_z}$, so that the output of the architecture is defined as

$$z \coloneqq v^{(L)} W_{\text{out}}^{\top} \in \mathbb{R}^{d_{\text{model}} \times d_z}.$$

4.B.2 Vision Transformer Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.4.3 is implemented in the discrete setting. In practice, we will have access to a discretization of the domain D containing N points and evaluations of the function $\mathsf{u} \in \mathcal{U}(D; \mathbb{R}^{d_u})$ at these points; the resolution of the input will be determined by $n_1 \times \ldots \times n_d = N$. The input to the model will therefore be a tensor $u \in \mathbb{R}^{n_1 \times \ldots \times n_d \times d_u}$. The input $u \in \mathbb{R}^{n_1 \times \ldots \times n_d \times d_u}$ is first concatenated with the discretization points $\{x_i\}_{i=1}^{N} \subset D$, to form the tensor $u_{\text{in}} \in \mathbb{R}^{n_1 \times \ldots \times n_d \times (d_u + d)}$. Denoting by $p_1 \times \ldots \times p_d$ the resolution of each patch, application of the reshaping operator Σ_{reshape} results in

$$\Sigma_{\text{reshape}}\left(u_{\text{in}}\right) \in \mathbb{R}^{P \times p_1 \times \ldots \times p_d \times (d_u + d)}.$$

Following Definition 4.4.2 and the subsequent discussion, the nonlocal linear operator W_{in} is implemented as

$$\mathsf{W}_{\text{in}}(\theta) u \coloneqq \mathcal{F}^{-1}\Big(R_{\mathsf{W}_{\text{in}}}(\theta) \cdot (\mathcal{F} u)\Big), \tag{4.60}$$

for any $u \in \mathbb{R}^{P \times p_1 \times \ldots \times p_d \times (d_u + d)}$, where we have made explicit the dependence on $\theta \in \Theta$ to highlight the learnable components in the implementation. The FFT is computed on the $p_1 \times \ldots \times p_d$ dimensions so that the resulting tensor is of the form $\mathcal{F} u \in \mathbb{C}^{P \times p_1 \times \ldots \times p_d \times (d_u + d)}$. Since u is convolved with a function that possess only $\left|Z_{k_{\max}}\right|$ Fourier modes, the higher Fourier modes may be truncated so that one may consider $\mathcal{F} u \in \mathbb{C}^{P \times (2k_{\max,1}+1) \times \ldots \times (2k_{\max,d}+1) \times (d_u + d)}$. The learnable tensor is $R_{\mathsf{W}_{\text{in}}} \in \mathbb{C}^{(2k_{\max,1}+1) \times \ldots \times (2k_{\max,d}+1) \times d_{\text{model}} \times (d_u + d)}$ and the operation in Fourier space is defined as

$$\Big(R \cdot \big(\mathcal{F} u\big)\Big)_{p, i_1, \ldots, i_d, \ell} = \sum_{k=1}^{d_u + d} R_{p, i_1, \ldots, i_d, \ell, k} \big(\mathcal{F} u\big)_{p, i_1, \ldots, i_d, k}, \tag{4.61}$$

for $p \in [\![1, P]\!]$, $1 \leqslant i_j \leqslant 2k_{\max,j} + 1$, $1 \leqslant \ell \leqslant d_{\text{model}}$. The inverse FFT is then applied along the dimensions $2k_{\max,1} + 1, \ldots, 2k_{\max,d} + 1$. The resulting tensor

defined as $v^{(0)} \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$ is then fed through the transformer encoder $\mathsf{E}_L : \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}} \longrightarrow \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$.

The multi-head attention is based on the self-attention operator acting on functions from Definition 4.3.2. While the expectation is computed with respect to a probability measure on the patch sequence, hence inducing a discrete probability mass function, the discretization of the domain introduces the need to approximate the L^2 inner product appearing in Definition 4.3.1. We resort to an approximation of the integral given by

$$\begin{aligned}\left\langle \mathsf{Q}_h\mathsf{v}(j), \mathsf{K}_h\mathsf{v}(k)\right\rangle_{L^2(D',\mathbb{R}^{d_K})} &= \int_{D'} \big(\mathsf{Q}_h\mathsf{v}(j)\big)(x)\cdot\big(\mathsf{K}\mathsf{v}(k)\big)(x)\,\mathrm{d}x &\text{(4.62a)}\\ &\approx \sum_{x_i\in D'} \Delta x_i\cdot\big(\mathsf{Q}_h\mathsf{v}(j)\big)(x_i)\cdot\big(\mathsf{K}_h\mathsf{v}(k)\big)(x_i), &\text{(4.62b)}\end{aligned}$$

where we have denoted by Δx_i the volume of the ith cell. We recall that in this case the linear operators $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ are pointwise local operators hence they admit finite dimensional representations $Q_h \in \mathbb{R}^{d_K\times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K\times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K\times d_{\text{model}}}$, for $h = 1,\ldots,H$. In the case of uniform patching and uniform discretization of the domain of size $s^d = N$, which is the setting we experiment with, the approximation reduces to the approximation of the integral of the form

$$\left\langle \mathsf{Q}_h\mathsf{v}(j), \mathsf{K}_h\mathsf{v}(k)\right\rangle_{L^2(D',\mathbb{R}^{d_K})} \approx \frac{1}{N}\sum_{i_1,\ldots,i_d=1}^{s}\sum_{\ell=1}^{d_K}\big(vQ_h^\top\big)_{j\times i_1\times\ldots\times i_d\times\ell}\big(vK_h^\top\big)_{k\times i_1\times\ldots\times i_d\times\ell}. \tag{4.63}$$

The multi-head attention operator is thus defined by the application of the linear transformation $W_{\text{MultiHead}}^\top \in \mathbb{R}^{Hd_K\times d_{\text{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations defined by the approximations above. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{P\times p_1\times\ldots\times p_d\times Hd_K}$ tensor, so that

$$\begin{aligned}\mathsf{A}_{\text{MultiHead}}(v) \coloneqq &\big(\mathsf{A}_{\mathsf{P}}(v;Q_1,K_1,V_1),\ldots,\mathsf{A}_{\mathsf{P}}(v;Q_H,K_H,V_H)\big)W_{\text{MultiHead}}^\top\\ &\in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}},\end{aligned} \tag{4.64}$$

for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$. The linear transformations $W_1^\top, W_2^\top \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ are applied pointwise. The layer normalization operator is applied as defined in (4.24) to every point in every patch so that $\mathsf{F}_{\text{LayerNorm}}(v) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$.

The operator F_{NN} is applied so that $\mathsf{F}_{\text{NN}}(v) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$, for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$. Indeed, in this finite-dimensional setting F_{NN} is defined as

$$\mathsf{F}_{\text{NN}}(v;W_3,W_4,b_1,b_2) = f\big(vW_4^\top + b_1\big)W_3^\top + b_2, \tag{4.65}$$

for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$, where f is a nonlinear activation function $W_3^\top, W_4^\top \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ are learnable local linear operators and $b_1, b_2 \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$ are learnable in the last dimension.

After computing the recurrence for E_L as described in Equation (4.10), a reshaping operator Σ'_{reshape} is applied to the resulting tensor so that

$$v \mapsto \Sigma'_{\text{reshape}}(v) \in \mathbb{R}^{n_1\times\ldots\times n_d\times d_{\text{model}}}, \tag{4.66}$$

for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$. A pointwise linear transformation $W_{\text{out}}^\top \in \mathbb{R}^{d_{\text{model}}\times d_z}$ is then applied according to

$$v \mapsto vW_{\text{out}}^\top, \tag{4.67}$$

for any $v \in \mathbb{R}^{n_1\times\ldots\times n_d\times d_{\text{model}}}$.

4.B.3 Fourier Attention Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.4.4 is implemented in the discrete setting. In practice, we consider a discretization of the domain D containing N points and evaluations of the function $u \in \mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$ at these points; the resolution of the input is defined by $n_1 \times \ldots \times n_d = N$. The input to the model is hence a tensor $u \in \mathbb{R}^{n_1\times\ldots\times n_d\times d_u}$. The input is concatenated with the discretization points $\{x_i\}_{i=1}^N \subset D$, to form the tensor $u_{\text{in}} \in \mathbb{R}^{n_1\times\ldots\times n_d\times(d_u+d)}$. Denoting by $p_1 \times \ldots \times p_d$ the resolution of each patch, application of the reshaping operator Σ_{reshape} results in

$$\Sigma_{\text{reshape}}\big(u_{\text{in}}\big) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times(d_u+d)}.$$

The learnable linear transformation $W_{\text{in}}^\top \in \mathbb{R}^{(d_u+d)\times d_{\text{model}}}$ is then applied as a map

$$\Sigma_{\text{reshape}}\big(u_{\text{in}}\big) \mapsto \Sigma_{\text{reshape}}\big(u_{\text{in}}\big)W_{\text{in}}^\top \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}.$$

The resulting tensor $\widetilde{u}^{(0)}$ is then fed through the transformer encoder

$$\mathsf{E}_L : \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}} \longrightarrow \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}.$$

In the recurrence relation that defines the encoder, $\mathsf{A}_{\text{MultiHead}}$ is the multi-head attention operator as defined in equation (4.11). In this setting, letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable linear integral operators $\mathsf{Q}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \longrightarrow \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$, $\mathsf{K}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \longrightarrow \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$, and $\mathsf{V}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \longrightarrow \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$ for $h =$

$1, \dots, H$. Following Definition 4.4.2 and the subsequent discussion we implement the linear integral operators as

$$\mathsf{Q}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{Q}_h}(\theta) \cdot (\mathcal{F}v)\Big), \tag{4.68a}$$

$$\mathsf{K}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{K}_h}(\theta) \cdot (\mathcal{F}v)\Big), \tag{4.68b}$$

$$\mathsf{V}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{V}_h}(\theta) \cdot (\mathcal{F}v)\Big), \tag{4.68c}$$

for $h = 1, \dots, H$ and any $v \in \mathbb{R}^{p_1 \times \dots \times p_d \times d_{\text{model}}}$, where we have made explicit the dependence on $\theta \in \Theta$ to highlight the learnable components in the implementation. In (4.68) the FFT is computed on the $p_1 \times \dots \times p_d$ so that the resulting tensor $\mathcal{F}v \in \mathbb{C}^{p_1 \times \dots \times p_d \times d_{\text{model}}}$. Since v is a function that possess only $\left|Z_{k_{\max}}\right|$ Fourier modes, the higher Fourier modes may be truncated so that one may consider $\mathcal{F}v \in \mathbb{C}^{(2k_{\max,1}+1) \times \dots \times (2k_{\max,d}+1) \times d_{\text{model}}}$. The learnable tensors are of dimension $R_{\mathsf{Q}_h}, R_{\mathsf{K}_h}, R_{\mathsf{V}_h} \in \mathbb{C}^{(2k_{\max,1}+1) \times \dots \times (2k_{\max,d}+1) \times d_K \times d_{\text{model}}}$ for $h = 1, \dots, H$, and the operation in Fourier space is defined as

$$\Big(R \cdot \big(\mathcal{F}v\big)\Big)_{i_1,\dots,i_d,\ell} = \sum_{k=1}^{d_{\text{model}}} R_{i_1,\dots,i_d,\ell,k} \big(\mathcal{F}v\big)_{i_1,\dots,i_d,k}, \tag{4.69}$$

for any $v \in \mathbb{R}^{p_1 \times \dots \times p_d \times d_{\text{model}}}$ and $1 \leqslant i_j \leqslant 2k_{\max,j} + 1$, $1 \leqslant \ell \leqslant d_K$. The inverse FFT is computed along the dimensions $2k_{\max,1}+1, \dots, 2k_{\max,d}+1$ so that $\mathsf{Q}_h v, \mathsf{K}_h v, \mathsf{V}_h v \in \mathbb{R}^{p_1 \times \dots \times p_d \times d_K}$ for $h = 1, \dots, H$. We recall that in this case the self-attention operator A is defined according to Definition 4.3.2. While the expectation is computed with respect to a probability measure on the patch sequence, hence inducing a discrete probability mass function, the discretization of the domain introduces the need to approximate the L^2 inner product appearing in Definition 4.3.1. We resort to an approximation of the integral given by (4.62) and (4.63). The multi-head attention operator is thus defined by the application of the linear transformation $W_{\text{MultiHead}}^{\top} \in \mathbb{R}^{Hd_K \times d_{\text{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{P \times p_1 \times \dots \times p_d \times Hd_K}$ tensor, so that

$$\begin{aligned} \mathsf{A}_{\text{MultiHead}}(v) \coloneqq &\big(\mathsf{A}_{\mathsf{P}}(v; \mathsf{Q}_1, \mathsf{K}_1, \mathsf{V}_1), \dots, \mathsf{A}_{\mathsf{P}}(v; \mathsf{Q}_H, \mathsf{K}_H, \mathsf{V}_H)\big) W_{\text{MultiHead}}^{\top} \\ &\in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}, \end{aligned} \tag{4.70}$$

for any $v \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$. The linear transformations $W_1^{\top}, W_2^{\top} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are applied pointwise. The layer normalization operator is applied as defined in (4.16) to every point in every patch so that $\mathsf{F}_{\text{LayerNorm}}(v) \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$.

Finally, the operator F_{NN} is parametrized and applied as in (4.65) so that $\mathsf{F}_{\mathrm{NN}}(v) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\mathrm{model}}}$, for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\mathrm{model}}}$.

A reshaping operator $\Sigma'_{\mathrm{reshape}}$ to the tensor resulting from the transformer encoder iteration as in Equation (4.66). A pointwise linear transformation $W_{\mathrm{out}}^{\top} \in \mathbb{R}^{d_{\mathrm{model}}\times d_z}$ is then applied as in Equation (4.67).

4.C Complexity of the Transformer Neural Operators

We summarize the expressions for the parameter complexity and evaluation cost (measured in floating point operations, FLOPS) of the neural operators from Section 4.4 and the Fourier neural operator (FNO) from Zongyi Li et al. (2021). These are provided in Tables 4.1 and 4.2, respectively. In Subsection 4.C.1 and 4.C.2 we provide further details for the computations of the parameter and evaluation cost complexities, respectively.

4.C.1 Parameter Complexity

Architecture	Parameter Complexity
FNO	$(d_u + d) \cdot d_{\mathrm{FNO}} + 2d_{\mathrm{FNO}} \cdot d_{\mathrm{model}} + d_{\mathrm{FNO}} \cdot d_z + 4 \cdot \left(d_{\mathrm{model}}^2 \cdot k_{\max} + d_{\mathrm{model}}^2\right)$
AFNO	$(d_u + d) \cdot d_{\mathrm{model}} + Nd_{\mathrm{model}} + d_{\mathrm{model}}d_z + 4 \cdot (8 + 4/b) \cdot d_{\mathrm{model}}^2$
Transformer NO	$(d_u + d) \cdot d_{\mathrm{model}} + d_{\mathrm{model}} \cdot d_z + 6 \cdot \left(6d_{\mathrm{model}}^2 + 2d_{\mathrm{model}}\right)$
ViT NO	$(d_u + d) \cdot d_{\mathrm{model}} \cdot k_{\max} + d_{\mathrm{model}} \cdot d_z + 6 \cdot \left(6d_{\mathrm{model}}^2 + 2d_{\mathrm{model}}\right)$
FANO	$(d_u + d) \cdot d_{\mathrm{model}} + d_{\mathrm{model}} \cdot d_z + 6 \cdot \left(3d_{\mathrm{model}}^2 \cdot k_{\max} + 3d_{\mathrm{model}}^2 + 2d_{\mathrm{model}}\right)$

Table 4.1: Parameter complexity for the three proposed transformer neural operators, as implemented in practice, compared to the Fourier neural operator (Zongyi Li et al., 2021). We note that for our implementations, d_{FNO}, which is the latent dimension of the two-layer lifting and projection MLP in FNO, is fixed to be 128, while the hyperparameter b in the AFNO is fixed to be 8. Parameters arising from possible bias terms, parameters arising from skip connections in FNO and parameters arising from normalizations in AFNO are omitted for ease of presentation.

4.C.2 Evaluation Cost

We note that a matrix vector product Wu for $W \in \mathbb{R}^{r\times r'}$ has cost $2rr'$. We approximate the cost of computing the FFT of a vector in $\mathbb{R}^r$ as $5r \log r$ (Hoop et al., 2022). We approximate the cost of the two-dimensional real FFT, used in implementation, by $(15/2) \cdot N \log(\sqrt{N})$ where N is the total number of grid points. Assuming that the grid is of size $s \times s = N$ for s an even integer, 1d FFTs are first computed for each row. The real FFT requires $s/2$ column 1d FFTs, hence the $(15/2) \cdot N \log \sqrt{N}$ complexity.

Architecture	Evaluation Cost
FNO	$2N(d_u + d + 2d_{\text{model}})d_{\text{FNO}} + 2Nd_{\text{FNO}}d_z$ $+ 4\big(15d_{\text{model}}N\log(\sqrt{N}) + k_{\max}\cdot(2d_{\text{model}}^2 - d_{\text{model}})$ $+ 2Nd_{\text{model}}^2\big)$
Transformer NO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z + 6\big(8d_{\text{model}}^2N$ $+ 2N^2d_{\text{model}} + 4d_{\text{model}}^2N\big)$
ViT NO	$(d_u + d + d_{\text{model}})\cdot 15N\log(\sqrt{N/P})/2$ $+ Pk_{\max}d_{\text{model}}\big(2(d_u + d) - 1\big) + 2Nd_{\text{model}}d_z$ $+ 6\Big(8d_{\text{model}}^2N + 2d_{\text{model}}NP + 4d_{\text{model}}^2N\Big)$
FANO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z$ $+ 6\Big(45d_{\text{model}}N\log(\sqrt{N/P}) + 3k_{\max}P\big(2d_{\text{model}}^2 - d_{\text{model}}\big)$ $+ 2Nd_{\text{model}}^2 + 2d_{\text{model}}NP + 4Nd_{\text{model}}^2\Big)$
AFNO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z$ $+ 4\big(16d_{\text{model}}^2 + 15Nd_{\text{model}}\log(\sqrt{N}) + 6N(2d_{\text{model}}^2/b - d_{\text{model}})\big)$

Table 4.2: Evaluation cost (in FLOPS) for the proposed transformer neural operators, as implemented in practice, compared to FNO (Zongyi Li et al., 2021). As done in Kaplan et al. (2020) and for ease of presentation, the additional evaluation cost arising from nonlinear activations, the softmax operator and normalizations are omitted. Furthermore, we omit the cost arising from skip connections. Indeed these do not affect the scaling. We further note that for our implementations, d_{FNO}, which is the latent dimension of the two-layer lifting and projection MLP in FNO, is fixed to be 128. In this table we present the expression for the evaluation cost of AFNO without consideration of possible patching, which we do not employ for our comparisons: this in combination with the fact that we do not employ mode truncation is the reason for the Nd_{model}^2 term. In our context, the parameter b in AFNO is set to $b = 8$.

Chapter 5

SMOOTHING AND FORECASTING WITH NEURAL OPERATORS

5.1 Introduction

Machine learning has opened new frontiers in data assimilation and forecasting in dynamical systems, ranging from methods which improve on traditional model-driven algorithms, through to new purely data-driven methods. The goal of this chapter is to establish a mathematical approach to the study of the purely data-driven methods. To this end, we consider the general dynamical system given by

$$\dot{p} = f(p, q), \qquad p(0) = p_0 \in \mathbb{R}^{d_p}, \tag{5.1a}$$

$$\dot{q} = g(p, q), \qquad q(0) = q_0 \in \mathbb{R}^{d_q}. \tag{5.1b}$$

Our focus is on two problems: (i) existence, universal approximation and approximation from data, of the map from observed $\{p(t)\}_{t\in[0,T]}$ to unobserved $\{q(t)\}_{t\in[0,T]}$; and (ii) existence, universal approximation, and approximation from data of the map from observed $\{p(t)\}_{t\in[0,T]}$ to unobserved $\{p(t)\}_{t\in[T,T+\tau]}$. In particular, in the training of data-driven methods for (i) and (ii) we assume no knowledge of the vector fields (f, g); and in (ii) we do not even assume we know the dimension of q, nor do we have any data in the q variables. Once the data-driven methods are trained they simply take an instance of $\{p(t)\}_{t\in[0,T]}$ as input.

5.1.1 Context and Literature Review

In the context of (5.1) there are three problem classes which naturally arise: (I) *smoothing* concerns the offline denoising and estimation of the observed and unobserved states over some time interval, given observational data in that same time interval; an instance of this, which we study in this chapter, is the estimation of unobserved $\{q(t)\}_{t\in[0,T]}$ from observed $\{p(t)\}_{t\in[0,T]}$; (II) *forecasting* concerns estimation of a state over some future time interval, given observational data over a past time interval; an instance of this, which we also study in this chapter, is the estimation of $\{p(t)\}_{t\in[T,T+\tau]}$ from observed $\{p(t)\}_{t\in[0,T]}$. Also of importance is (III) *filtering*, which concerns the online estimation of an underlying state, sequentially as observations are received; this, however, is not a focus of the current work.

The field of data assimilation (DA) is concerned with the use of observational data to perform state estimation in dynamical systems (including the estimation of quantities that are not observed). Both smoothing and filtering are used in practice, often in conjunction with forecasting, as in the field of weather prediction. Traditional model-driven techniques combine the observations with knowledge of the underlying dynamical system to address the three tasks (I–III) (Kalnay, 2002; Asch, Marc Bocquet, and Nodet, 2016; G. Evensen, F.C. Vossepoel, and van Leeuwen, 2022). Although primarily developed for geophysical applications in the ocean-atmosphere sciences, such model-driven methods to address tasks (I–III) have been systematized and now constitute general-purpose methodologies (Reich and Colin Cotter, 2015; Pandya, Vachharajani, and Srivastava, 2022; Sanz-Alonso, A. Stuart, and Taeb, 2023b). In the last few years new purely data-driven algorithms, which do not require knowledge of the dynamical system when deployed, have started to emerge (Kurth et al., 2023; Bi et al., 2023; Price et al., 2025b; Allen et al., 2025; Alexe et al., 2024; Kossaifi et al., 2026). These methods have also been developed primarily in weather forecasting, but present opportunities for deployment in many other domains. Developing mathematical theory pertinent to these methods can play an important role in the process of making the methods more widely applicable and is the focus of this chapter.

Over the last half century, model-driven filtering, smoothing and forecasting methods have dominated in most application domains, starting from the seminal work of Kalman and Bucy for linear Gaussian systems (Kalman, 1960; Kalman and Bucy, 1961); the field evolved with introduction of the bootstrap, extended (ExKF) and ensemble Kalman filters (EnKF) (Doucet, Godsill, and Andrieu, 2000; Andrew H Jazwinski, 2007; Evensen, 1994) which, respectively, typically work well for small-, medium- and large-scale dynamical systems: bootstrap particle filter weights collapse in high dimensions and so they are best in low dimensions; EnKF has equal weights and has performed well for weather forecasting with state spaces in the billions of variables; ExKF falls between these cases as it makes a Gaussian approximation (mitigating weight collapse), but the covariances cannot be propagated in high dimension as they are too large. All model-driven approaches require knowledge of the dynamics, requiring computationally expensive evaluations in the context of many large-scale applications of interest. Purely data-driven, model-free, methods for forecasting were proposed by Lorenz in 1969 (Edward N. Lorenz, 1969), and go by the name of analog forecasting; these methods were not widely adopted, however, primarily because of the lack of data to support them, and because they are

discontinuous as a function of input data. In the last decade kernel analog forecasting has been developed (Alexander and Giannakis, 2020), a smoothening of the original approach of Lorenz, supported by theory. Furthermore, methods such as dynamic mode decomposition (Schmid, 2010), which are also purely data-driven, have been widely adopted and come with a developing theory (Mezić, 2020). These methods have not yet, however, found success in the context of large-scale data assimilation and forecasting problems; their success is mainly in the identification of large-scale coherent features in high-dimensional systems.

Machine learning has recently emerged as a novel computational tool for improving data assimilation and forecasting in both the model- and data-driven settings. Learning has been introduced within model-driven DA in a variety of ways, primarily in the filtering context. For example, it has been used for model error correction (Farchi et al., 2021; Levine and A. Stuart, 2021), for constructing cheap surrogates of the dynamics (Sanz-Alonso and Waniorek, 2025; Adrian, Sanz-Alonso, and Willett, 2025) and more recently to approximate conditioning in the assimilation step (Bach, Baptista, Calvello, et al., 2026). A data-driven approach to smoothing has recently been employed in Allen et al. (2025) and Gupta et al. (2026) for estimating the initial conditions of weather dynamics. Machine learning has also introduced a new approach to forecasting which is fully data-driven; this involves the learning of mappings which output predictions for the forecasted components of the system from their past observations. This approach has garnered wide interest in domains like weather forecasting, where both model evaluations and assimilation steps are particularly costly (Kurth et al., 2023; Bi et al., 2023; Price et al., 2025b; Alexe et al., 2024; Kossaifi et al., 2026). In a similar context, deploying pretrained time-series foundation models for forecasting of partially observed chaotic dynamical systems has been numerically investigated in Zhang and Gilpin (2025). Underpinning these data-driven approaches with theoretical foundations is the goal of this chapter. In Table 5.1 we summarize the distinction between model-driven DA, that has dominated for the last half century, and recently emerging data-driven approaches including, but not limited to, the specific methods we study in this chapter.

5.1.2 Contributions and Outline

In this chapter we develop theory for, and insights into, the data-driven smoothing and forecasting problems previously defined. We work in the noise-free setting, and in continuous time, focusing on three essential ingredients: (a) the interplay between the theoretical properties of the dynamics and the existence of well-defined

Table 5.1: Comparison between model-driven and data-driven approaches.

	Model-driven	**Data-driven**
Algorithm	Uses dynamical physics model	Uses data from dynamical model and/or direct observations
Limitation	Possibly expensive to evaluate and may miss some dynamics due to unmodeled physics	Requires existence of direct mapping, a sufficiently dense training dataset and may not enforce explicit physical constraints
Advantage	Interpretable and output satisfies prescribed physical constraints	Possibly cheap to evaluate and agnostic to model dynamics

maps that can in principle be approximated; (b) the universal approximation properties of neural operators (N. Kovachki et al., 2023a), neural network architectures parametrizing mappings between spaces of functions, for these maps; and (c) the implementation and properties of approximations of these maps learned from data in a model-free setting. We show that it is possible to construct the desired operators, defined locally, under an observability-rank condition on the dynamics. We show that this condition is intimately connected to the setup of Hermann and Krener (1977), a foundational work in control theory which establishes that, under a more general observability-rank condition, distinct observations correspond to distinct dynamics. Takens' delay embedding theorem (Takens, 2006; Sauer, Yorke, and Casdagli, 1991) is also conceptually pertinent to our work, but the concrete control-theoretic perspective of Hermann and Krener (1977) provides a more natural basis for the theory and algorithms that we develop here. The existence of the desired operators allows approximation via neural operators, for which an emerging universal approximation theory has been developed (Lanthaler, Zongyi Li, and Andrew M. Stuart, 2025). In particular, we show how architectures such as the transformer neural operator (Calvello, Monmarché, et al., 2026) may be applied in the smoothing and forecasting contexts, and yield desirable universal approximation results. We then implement these neural operators, demonstrating data-driven approximation of the maps in practice. Our contributions are as follows:

(C1) We introduce an observability condition on the dynamics, as appearing in control theoretic literature, which guarantees existence of a continuous operator locally mapping an observed component of the system to an unobserved component. This condition is defined in Assumption P2.

(C2) We prove the existence of a neural operator approximating to arbitrary ac-

curacy the operator mapping an observed component of the system to an unobserved component. This is the first universal approximation theorem for a smoothing problem in DA. The result is stated in Theorem 5.3.4.

(C3) We prove the existence of a neural operator approximating to arbitrary accuracy the operator mapping a trajectory of the observed component of the system to the future of that trajectory. This is the first universal approximation theorem for a forecasting problem for partially observed dynamical systems. The result is stated in Theorem 5.3.7.

(C4) We showcase the universal approximation theory developed by applying transformer neural operators on smoothing and forecasting problems in the setting of the Lorenz '63, Lorenz '96, and Kuramoto-Sivashinsky dynamical systems.

We describe the general dynamical system setting in Subsection 5.1.3; here we also define the smoothing and forecasting problems which will be the object of investigation. In Section 5.2 we introduce the observability framework that underpins our analysis. Building on regularity and observability assumptions on the dynamical systems, we develop the universal approximation theory for smoothing and forecasting in Section 5.3, thus addressing Contributions (C1) to (C3). In Section 5.4 we demonstrate this theory numerically, thus addressing Contribution (C4). Finally in Section 5.5 we discuss the results of the chapter and conclude by outlining avenues for further work.

5.1.3 Dynamical System Setup

Consider the dynamical system (5.1). We make the following regularity assumption:

Assumption P2 (Regularity). *Assume that, for some* $k \in \mathbb{N}$*,*

- $f \in C^k(\mathbb{R}^{d_p+d_q}; \mathbb{R}^{d_p})$*;*
- $g \in C^k(\mathbb{R}^{d_p+d_q}; \mathbb{R}^{d_q})$.

Assume, furthermore, that solutions to (5.1) *exist for all* $t \in \mathbb{R}^+$.

Under Assumption P2 it follows that, for any $T > 0$, $p \in C^k([0,T]; \mathbb{R}^{d_p})$ and $q \in C^k([0,T]; \mathbb{R}^{d_q})$. We define by $\Phi : \mathbb{R}^+ \times \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_p+d_q}$ the semigroup of operators associated to the dynamical system, so that $\big(p(t), q(t)\big) = \Phi(t, p_0, q_0)$.

Let $I \subset \mathbb{R}^{d_p+d_q}$ denote a compact set of initial conditions to (5.1) and fix $T > 0$. Since by Assumption P2 solutions to (5.1) exist for all $t \in \mathbb{R}^+$, the result of Teschl, 2012, Theorem 2.10 implies that $\Phi(\bullet, \bullet) \in C^k([0,T] \times I; \mathbb{R}^{d_p+d_q})$; therefore, it may be deduced that the map $x \mapsto \Phi(\bullet, x) \in C^k([0,T]; \mathbb{R}^{d_p+d_q})$ is continuous. Since the image of a compact set under a continuous map is compact, we deduce that the set of orbits $S^I_{[0,T]} = \{\Phi(t,x) \text{ for } t \in [0,T] : x \in I\}$ is compact in $C^k([0,T]; \mathbb{R}^{d_p+d_q})$. Let $\pi_p \colon \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_p}$ be the projection map onto the first d_p coordinates and $\pi_q \colon \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_q}$ be the projection map onto the remaining d_q coordinates of the state space. We note that these are both continuous functions. Define the projected sets

$$S^I_{[0,T],p} = \{\pi_p(s) : s \in S^I_{[0,T]}\}, \quad S^I_{[0,T],q} = \{\pi_q(s) : s \in S^I_{[0,T]}\}; \tag{5.2}$$

these too are compact in $C^k([0,T]; \mathbb{R}^{d_p})$ and $C^k([0,T]; \mathbb{R}^{d_q})$ since $S^I_{[0,T]}$ is compact and projection is continuous. Fixing some $\tau > 0$, for the set of orbits defined on the future time interval $S^I_{[T,T+\tau]} = \{\Phi(t,x) \text{ for } t \in [T, T+\tau] : x \in I\}$ we similarly define the projected set

$$S^I_{[T,T+\tau],p} = \{\pi_p(s) : s \in S^I_{[T,T+\tau]}\}. \tag{5.3}$$

It is readily deduced that $S^I_{[T,T+\tau],p}$ is also compact in $C^k([T,T+\tau]; \mathbb{R}^{d_p})$. Given the dynamical systems (5.1) equipped with the regularity assumption in Assumption P2, we define the two data assimilation problems central to this chapter.

(P1) *Smoothing*: recover $q \in S^I_{[0,T],q} \subset C^k([0,T]; \mathbb{R}^{d_q})$ from observed $p \in S^I_{[0,T],p} \subset C^k([0,T]; \mathbb{R}^{d_p})$.

(P2) *Forecasting*: recover $p \in S^I_{[T,T+\tau],p}$ from $p \in S^I_{[0,T],p}$.

5.2 Observability

In this section we develop the observability framework that underpins the existence and universal approximation of smoothing and forecasting maps, the subject of Section 5.3. Consider dynamical system (5.1). Underlying the theory in Section 5.3 is the question of whether a unique map $\{p(t)\}_{t\in[0,T]} \mapsto \{q(t)\}_{t\in[0,T]}$ exists. A sufficient condition is that initial condition q_0 is determined by $\{p(t)\}_{t\in[0,T]}$. Then, equation (5.1b) can be integrated as a non-autonomous equation for $q(\cdot)$, driven by the observed $p(\cdot)$. This section is hence focused on the question of recovering q_0 from p and its derivatives at time $t = 0$, noting that these derivatives are determined by $\{p(t)\}_{t\in[0,T]}$, assuming sufficient differentiability. In fact we formulate the question

a bit more generally, seeking to recover $q(t)$ at some given time t, given p and its derivatives at that time. In Subsection 5.2.1 we introduce the notation required to discuss the observability rank condition which encapsulates solvability for $q(t)$ at some time t. We conclude in Subsection 5.2.3 with an example of observability in the Lorenz '63 equation.

5.2.1 Observability Setup

The results in Section 5.3 are obtained via an underlying observability assumption on the system (5.1). To state such an observability condition we introduce the following notation relating to the dynamical system. Let $\mathcal{L}^m$ denote the m^{th} Lie derivative of f along the vector field (f, g) and let $\widetilde{F}^{(n)} \colon \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{nd_p}$ be defined from the first n Lie derivatives of f along the vector field (f, g) as follows:

$$\widetilde{F}^{(n)}(\mathfrak{p}, \mathfrak{q}) = \left(\mathcal{L}^0 f(\mathfrak{p}, \mathfrak{q})^\top, \mathcal{L}^1 f(\mathfrak{p}, \mathfrak{q})^\top, \ldots, \mathcal{L}^{n-1} f(\mathfrak{p}, \mathfrak{q})^\top\right)^\top, \tag{5.4}$$

for any $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$. We also define $F^{(n)} \colon \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{(n+1)d_p}$ as

$$F^{(n)}(\mathfrak{p}, \mathfrak{q}) = \left(\mathfrak{p}^\top, \mathcal{L}^0 f(\mathfrak{p}, \mathfrak{q})^\top, \mathcal{L}^1 f(\mathfrak{p}, \mathfrak{q})^\top, \ldots, \mathcal{L}^{n-1} f(\mathfrak{p}, \mathfrak{q})^\top\right)^\top, \tag{5.5}$$

for any $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$. We note that the two differ, as $\widetilde{F}$ also includes $\mathfrak{p} \in \mathbb{R}^{d_p}$. By considering (5.1a), we may deduce that for any $(p, q) \in S^I_{[0,T]}$, it holds that

$$\begin{aligned}\mathcal{L} f\big(p(t), q(t)\big) &= \partial_p f\big(p(t), q(t)\big) \cdot f\big(p(t), q(t)\big) + \partial_q f\big(p(t), q(t)\big) \cdot g\big(p(t), q(t)\big) \\ &= \partial_t^2 p(t),\end{aligned}$$

for all $t \in [0, T]$. Therefore, by repeated application of Lie derivatives, it holds for any $(p, q) \in S^I_{[0,T]} \subset C^k([0, T]; \mathbb{R}^{d_p}) \times C^k([0, T]; \mathbb{R}^{d_q})$ and for all $n \leqslant k - 1$ that

$$\mathcal{L}^n f\big(p(t), q(t)\big) = \partial_t^{n+1} p(t), \tag{5.6}$$

for all $t \in [0, T]$. Define operator $\widetilde{P}^{(n)} \colon C^k([0, T]; \mathbb{R}^{d_p}) \to \bigotimes_{j=1}^n C^{k-j}([0, T]; \mathbb{R}^{d_p})$ via its action on $p \in C^k([0, T]; \mathbb{R}^{d_p})$ as follows:

$$\big(\widetilde{P}^{(n)}(p)\big)(t) = \left(\partial_t^1 p(t)^\top, \ldots, \partial_t^n p(t)^\top\right)^\top. \tag{5.7}$$

We may then deduce that any $(p, q) \in S^I_{[0,T]}$ satisfies the equation

$$\widetilde{F}^{(n)}\big(p(t), q(t)\big) = \big(\widetilde{P}^{(n)}(p)\big)(t), \tag{5.8}$$

for all $t \in [0, T]$. Similarly, define operator $P^{(n)} \colon C^k([0, T]; \mathbb{R}^{d_p}) \to \bigotimes_{j=0}^n C^{k-j}([0, T]; \mathbb{R}^{d_p})$ by

$$\big(P^{(n)}(p)\big)(t) = \left(\partial_t^0 p(t)^\top, \partial_t^1 p(t)^\top, \ldots, \partial_t^n p(t)^\top\right)^\top, \tag{5.9}$$

to deduce that any $(p, q) \in S^I_{[0,T]}$ satisfies the equation

$$F^{(n)}\big(p(t), q(t)\big) = \big(P^{(n)}(p)\big)(t), \tag{5.10}$$

for all $t \in [0, T]$. Notice that in the topology of (1.1), $P^{(n)}$ is continuous for all n since differentiation $(k - j)$-times is continuous in each $C^{k-j}([0, T]; \mathbb{R}^{d_p})$ component for $0 \leqslant j \leqslant n$.

5.2.2 Observability-Rank Condition

In the following, we introduce a condition under which, at some $t \in [0, T]$, it is possible to recover the unobserved $q(t)$ from the observed $p(t)$. Fixing some $t \in [0, T]$, and choosing $\widetilde{L} : \mathbb{R}^{nd_p} \to \mathbb{R}^{d_q}$, we consider the following equation for $\mathfrak{q}$:

$$\widetilde{L}\widetilde{F}^{(n)}\big(p(t), \mathfrak{q}\big) = \widetilde{L}\big(\widetilde{P}^{(n)}(p)(t)\big). \tag{5.11}$$

This constitutes a system of d_q equations for $\mathfrak{q} \in \mathbb{R}^{d_q}$. If there is a time t and linear operator $\widetilde{L}$ such that the operator $\widetilde{L}\widetilde{F}^{(n)}(p(t), \bullet) : \mathbb{R}^{d_q} \to \mathbb{R}^{d_q}$ is invertible, then there exists a unique solution $\mathfrak{q}$ to (5.11) and, by (5.8), $\mathfrak{q} = q(t)$. This fact provides motivation for imposing a local invertibility condition to allow existence of a map from observed $p \in C^k([0, T]; \mathbb{R}^{d_p})$ to unobserved $q \in C^k([0, T]; \mathbb{R}^{d_q})$ in the neighborhood of invertibility. However, in order to connect with formulations of related problems in the control theory literature, we formulate a slight modification of the foregoing.

Again fixing some $t \in [0, T]$, and now choosing $L : \mathbb{R}^{(n+1)d_p} \to \mathbb{R}^{d_p+d_q}$, we consider the equation

$$LF^{(n)}\big(\mathfrak{p}, \mathfrak{q}\big) = L\big(P^{(n)}(p)(t)\big). \tag{5.12}$$

This constitutes a system of $d_p + d_q$ equations for $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$. If there exists a time t and linear transformation L such that the operator $LF^{(n)}(\bullet, \bullet) : \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_p+d_q}$ is invertible, then there exists a unique solution $(\mathfrak{p}, \mathfrak{q})$, to (5.12) and, by (5.10), $(\mathfrak{p}, \mathfrak{q}) = \big(p(t), q(t)\big)$ and, in particular, we recover $q(t)$. Both approaches outlined lead to recovery of $q(t)$ from $p(t)$ and its derivatives, at some time $t \in [0, T]$. This motivates the following assumption.

Assumption P2 (Observability-Rank Condition). *We say that the system* (5.1) *satisfies the observability-rank condition at* $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$*, if there exists* $n \in \mathbb{N}$ *and a linear operator* $L\colon \mathbb{R}^{(n+1)d_p} \to \mathbb{R}^{d_p+d_q}$ *such that the rank of the matrix* $\mathsf{D}LF^{(n)}\big(\mathfrak{p}, \mathfrak{q}\big)$ *is* $d_p + d_q$.

Remark 5.2.1. *We recall that the inverse function theorem for Euclidean spaces states that for $F\colon \mathbb{R}^d \to \mathbb{R}^d$ a C^1 map, if $\mathsf{D}F$ is invertible at a point $x \in F$ then there exist open neighborhoods U, V of x and $F(x)$ respectively, such that there exists a continuous inverse $F^{-1}\colon V \to U$. The existence of open sets U and V along with continuous inverse is what we refer to as local invertibility. Under the observability-rank condition, $LF^{(n)}(\bullet, \bullet)$ is locally invertible at $(\mathfrak{p}, \mathfrak{q})$ by the inverse function theorem.*

Remark 5.2.2 (Observability in Control Theory). *We note that the observability-rank condition from Assumption P2 is an instance of the observability condition from the seminal paper Hermann and Krener (1977). In particular, considering the observation function $h(\mathfrak{p}, \mathfrak{q}) = \mathfrak{p}$ for any $\mathfrak{p} \in \mathbb{R}^{d_p}, \mathfrak{q} \in \mathbb{R}^{d_q}$, we may reformulate* (5.10) *in terms of this observation function h and hence Assumption P2 in the control theoretic setting of Hermann and Krener (1977). We elaborate on this connection in Section 5.A.*

5.2.3 Observability of Lorenz '63 Model

Consider the Lorenz '63 dynamical system in the form

$$\begin{cases} \dot{x}(t) = \sigma(y(t) - x(t)), \\ \dot{y}(t) = x(t)(\rho - z(t)) - y(t), \\ \dot{z}(t) = x(t)y(t) - \beta z(t), \end{cases} \tag{5.13}$$

initialized at $(x_0, y_0, z_0) \in \mathbb{R}^3$. We consider the setting where the component $p = x$ is observed and $q = (y, z)$ is unobserved. It is readily shown that the system satisfies the observability-rank condition, given this choice of observation. We note that by (5.10) it holds that

$$F^{(2)}(x_0, y_0, z_0) = \begin{pmatrix} x_0 \\ \sigma(y_0 - x_0) \\ \sigma\Big(x_0(\rho - z_0) - y_0 - \sigma(y_0 - x_0)\Big) \end{pmatrix}. \tag{5.14}$$

Now, for L the identity matrix, the associated differential is

$$\mathsf{D}\, LF^{(2)}(x_0, y_0, z_0) = \begin{pmatrix} 1 & 0 & 0 \\ -\sigma & \sigma & 0 \\ \sigma(\rho + \sigma - z_0) & -\sigma(\sigma + 1) & -\sigma x_0 \end{pmatrix}.$$

From the upper-triangular structure it follows readily that this is of full rank for $x_0 \neq 0$. Hence we can conclude that the Lorenz '63 dynamical system satisfies

the observability-rank condition at $t = 0$, whenever $x_0 \neq 0$. In particular, the map $\{x(t)\}_{t\in[0,T]} \mapsto \{\big(y(t), z(t)\big)\}_{t\in[0,T]}$ is well-defined whenever $x_0 \neq 0$.

Moreover, the Lorenz '63 model is *not* observable if the only data given is z. This is due to the fact that, given a solution (x, y, z), there is another solution given by reflection, $(-x, -y, z)$. Specifically, if (x, y, z) satisfies (5.13), then

$$-\dot{x} = \sigma((-y) - (-x)) \tag{5.15}$$

$$-\dot{y} = \rho(-x) - (-y) - (-x)z \tag{5.16}$$

$$\dot{z} = (-x)(-y) - \beta z, \tag{5.17}$$

i.e. $(-x, -y, z)$ is a solution to (5.13). Thus, there cannot exist a well-defined mapping $z \mapsto (x, y)$. We also demonstrate this experimentally in Section 5.4.

5.3 Universal Approximation of Smoothing and Forecasting Maps

In this section we state and prove universal approximation results for neural operators approximating the maps underlying the smoothing Problem (P1) and forecasting Problem (P2). In Subsection 5.3.1 we recall a specific form of universal approximation theorem for neural operators that we employ for both the smoothing and forecasting problems. Subsection 5.3.2 is devoted to the existence and universal approximation map $\{p(t)\}_{t\in[0,T]} \mapsto \{q(t)\}_{t\in[0,T]}$. The recovery of this map amounts to solving an instance of a smoothing Problem (P1). In Subsection 5.3.3 we apply the results from the subsection preceding it to deduce the existence and universal approximation of map $\{p(t)\}_{t\in[0,T]} \mapsto \{p(t)\}_{t\in[T,T+\tau]}$. The recovery of this map amounts to solving an instance of a forecasting Problem (P2).

5.3.1 Universal Approximation Theorem

We state a general universal approximation theorem for neural operators that gives conditions under which a continuous operator may be approximated by a transformer neural operator from Calvello, Nikola B. Kovachki, et al., 2025, Theorem 22; we note that other neural operators could also be used (Nikola B Kovachki, Lanthaler, and Andrew M Stuart, 2024). We use this theorem in the two subsequent subsections to study the approximation of smoothing and forecasting maps.

Theorem 5.3.1. *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary, and fix integers $s, s' \geqslant 0$. If $\Psi^\dagger \colon C^s(\overline{D}; \mathbb{R}^r) \to C^{s'}(\overline{D}; \mathbb{R}^{r'})$ is a continuous operator and $K \subset C^s(\overline{D}; \mathbb{R}^r)$ a compact set, then for any $\varepsilon > 0$ there exists a transformer neural*

operator $\Psi(\bullet;\theta)\colon K \subset C^s(\overline{D};\mathbb{R}^r) \to C^{s'}(\overline{D};\mathbb{R}^{r'})$ *so that*

$$\sup_{u\in K} \left\| \Psi^\dagger(u) - \Psi(u;\theta) \right\|_{C^{s'}} \leqslant \varepsilon. \tag{5.18}$$

5.3.2 Smoothing

In this subsection we show that if a dynamical system of the form (5.1) satisfies the regularity conditions in Assumption P2 and the observability-rank condition Assumption P2 at a point, it is possible to construct a continuous operator mapping the observed trajectory to the initial condition of the full trajectory; we then show how this leads to existence of a continuous operator mapping the observed trajectory over a time interval to the unobserved trajectory over the same interval. We note that the continuity of these operators is necessary to apply existing universal approximation results for neural operators.

Proposition 5.3.2 (Existence of map $W : p \mapsto (p(0), q(0))$**).** *Consider the dynamical system* (5.1) *satisfying the regularity Assumption P2, for integer* k*, and the observability Assumption P2 at some* $(\mathfrak{p},\mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$*, for integer* $n \leqslant k$*. Let* $U \ni (\mathfrak{p},\mathfrak{q})$ *and* $V \ni LF^{(n)}(\mathfrak{p},\mathfrak{q})$ *be the open sets given by the inverse function theorem. Then, for every compact* $I \subset U$*, there exists a continuous map* $W\colon S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to \mathbb{R}^{d_p+d_q}$ *such that, for every* $(p,q) \in S^I_{[0,T]}$*,*

$$W(p) = (p(0), q(0)). \tag{5.19}$$

Proof. Observe that from (5.12) for every $(p,q) \in S^I_{[0,T]}$ we have that

$$LF^{(n)}(p(0),q(0)) = L(P^{(n)}(p)(0)) \tag{5.20}$$

and $LF^{(n)}(\bullet,\bullet)$ is locally invertible by the inverse function theorem from Assumption P2. Furthermore, the evaluation functional at time $t = 0$, namely $\delta_0\colon \bigotimes_{j=0}^n C^{k-j}([0,T];\mathbb{R}^{d_p}) \to \mathbb{R}^{(n+1)d_p}$, is linear and bounded, hence continuous with respect to the topology of (1.1). Using these operators, we can construct $W\colon S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to \mathbb{R}^{d_p+d_q}$ so that

$$W(p) = \left(LF^{(n)}\right)^{-1} \circ L \circ \delta_0 \circ P^{(n)}(p). \tag{5.21}$$

From (5.20) we have that $L \circ \delta_0 \circ P^{(n)}(p) \in V$ for all p, so composition with $(LF^{(n)})^{-1}$ is well-defined. By construction $(LF^{(n)})^{-1}$ is continuous and W is continuous since it is a composition of continuous operators. For all $(p,q) \in S^I_{[0,T]}$ we have $W(p) = (p(0),q(0))$ since $(p(0),q(0))$ is a solution to (5.20) and $LF^{(n)}$ is a bijection from U to V. □

Following Proposition 5.3.2 we may now establish the existence of an operator mapping observed p trajectories to unobserved q trajectories.

Proposition 5.3.3 (Existence of map $\Psi_S^\dagger : p \mapsto q$). *Consider the dynamical system* (5.1) *satisfying regularity Assumption P2, for integer k, and the observability Assumption P2 at some $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$, for integer $n \leqslant k$. Let $U \ni (\mathfrak{p}, \mathfrak{q})$ and $V \ni LF^{(n)}(\mathfrak{p}, \mathfrak{q})$ be the open sets given by the inverse function theorem. Then for every compact $I \subset U$, there exists continuous operator $\Psi_S^\dagger \colon S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to S^I_{[0,T],q} \subset C^k([0,T];\mathbb{R}^{d_q})$ such that, for every $(p,q) \in S^I_{[0,T]}$ and $t \in [0,T]$,*

$$\Psi_S^\dagger(p)(t) = q(t). \tag{5.22}$$

Proof. We construct such a $\Psi^\dagger$ using W from Proposition 5.3.2 and composition with the solution operator Φ. Define $\Psi_S^\dagger : S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to S^I_{[0,T],q} \subset C([0,T];\mathbb{R}^{d_q})$ by

$$\Psi_S^\dagger(p)(t) = \pi_q \circ \Phi(t, W(p)), \quad 0 \leqslant t \leqslant T. \tag{5.23}$$

Recall that π_q is continuous. By construction $W(p) = (p(0), q(0))$, so $\pi_q \circ \Phi(t, W(p)) = q(t)$. Observe that this is a continuous operator since Φ is continuous with respect to initial conditions, W is continuous with respect to p and composition of continuous operators is continuous. □

Leveraging the approximation properties of neural operators, we may now establish a universal approximation theorem for the solution of Problem (P1), an instance of a smoothing problem.

Theorem 5.3.4 (Universal Approximation for Smoothing). *Consider the dynamical system* (5.1) *satisfying regularity Assumption P2, for integer k, and the observability Assumption P2 at some $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$, for integer $n \leqslant k$. For any $\varepsilon > 0$ there exists a neural operator $\Psi(\bullet;\theta) : S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to S^I_{[0,T],q} \subset C^k([0,T];\mathbb{R}^{d_q})$ satisfying*

$$\sup_{p \in S^I_{[0,T],p}} \|\Psi_S^\dagger(p) - \Psi(p;\theta)\|_{C^k} \leqslant \varepsilon.$$

Proof. We know that $\Psi_S^\dagger$ exists and continuous since the assumptions of Proposition 5.3.3 are satisfied. We recall that $S^I_{[0,T],p}$ is compact. The conclusion follows from a direct application of Theorem 5.3.1 with $D = [0,T]$, $r = d_p$, $r' = d_q$, $\Psi^\dagger = \Psi_S^\dagger$, and $K = S_{[0,T],p}$. □

Remark 5.3.5. *Since our observability-rank condition Assumption P2 is local, all operators constructed in the previous theorems have a domain of input trajectories which are near the point of invertibility* $(\mathfrak{p}, \mathfrak{q})$. *The set of these trajectories is determined, non-constructively, by the inverse function theorem. It is of general interest to determine the largest possible domain of input trajectories for which the smoothing map* $\Psi_S^\dagger$ *(or the subsequently defined forecasting map* $\Psi_F^\dagger$*) exists and is continuous. A potential approach to this problem is to determine all points of invertibility and stitch together a global inverse from the open sets given by the inverse function theorem. Such analysis, however, will likely need to be carried out on a case-by-case basis and is not the focus of the current work.*

In the case of the Lorenz '63 system, studied in subsection 5.2.3, it is easy to see that $U = \mathbb{R} \setminus \{0\} \cup \mathbb{R}^2$. *Therefore the domain* $S^I_{[0,T],p}$ *of* $\Psi_S^\dagger$ *contains all trajectories with initial condition* x_0 *uniformly bounded away from* 0.

5.3.3 Forecasting

Using Proposition 5.3.2 we may also establish the existence of an operator mapping the observed portion of p trajectories, $p \restriction_{[0,T]}$, to the future portion $p \restriction_{[T,T+\tau]}$.

Proposition 5.3.6 (Existence of map $\Psi_F^\dagger : p \restriction_{[0,T]} \mapsto p \restriction_{[T,T+\tau]}$**).** *Consider the dynamical system* (5.1) *satisfying regularity Assumption P2, for integer* k*, and the observability Assumption P2 at some* $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$*, for integer* $n \leqslant k$*. Let* $U \ni (\mathfrak{p}, \mathfrak{q})$ *and* $V \ni LF^{(n)}(\mathfrak{p}, \mathfrak{q})$ *be the open sets given by the inverse function theorem. Then for every compact* $I \subset U$*, there exists a continuous operator* $\Psi_F^\dagger \colon S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to S^I_{[T,T+\tau],p} \subset C^k([T,T+\tau];\mathbb{R}^{d_p})$ *such that, for every* $(p,q) \in S^I_{[0,T]}$*,*

$$\Psi_F^\dagger(p)(t) = p(t), \qquad T \leqslant t \leqslant T+\tau. \tag{5.24}$$

Proof. We construct such a $\Psi_F^\dagger$ using W from Proposition 5.3.2 and composition with the solution operator Φ. Define $\Psi_F^\dagger \colon S^I_{[0,T],p} \subset C^k([0,T];\mathbb{R}^{d_p}) \to S^I_{[T,T+\tau],p} \subset C^k([T,T+\tau];\mathbb{R}^{d_p})$ by

$$\Psi_F^\dagger(p)(t) = \pi_p \circ \Phi(t, W(p)), \quad T \leqslant t \leqslant T+\tau. \tag{5.25}$$

We first recall that π_p is continuous. By construction it holds that $W(p) = (p(0), q(0))$, hence $\pi_p \circ \Phi(t, W(p)) = p(t)$ for $T \leqslant t \leqslant T+\tau$. We observe that this is a continuous operator since Φ is continuous with respect to initial conditions, W is continuous with respect to p, and composition of continuous operators is continuous. □

Leveraging the approximation properties of neural operators, we may now establish a universal approximation theorem for the solution of Problem (P2), an instance of a forecasting problem.

Theorem 5.3.7 (Universal Approximation for Forecasting). *Consider the dynamical system* (5.1) *satisfying regularity Assumption P2, for integer k, and the observability Assumption P2 at some $(\mathfrak{p}, \mathfrak{q}) \in \mathbb{R}^{d_p+d_q}$, for integer $n \leqslant k$. Then for any $\varepsilon > 0$ there exists a neural operator $\Psi(\bullet; \theta) \colon S^I_{[0,T],p} \subset C^k([0,T]; \mathbb{R}^{d_p}) \to S^I_{[T,T+\tau],p} \subset C^k([T, T+\tau]; \mathbb{R}^{d_p})$ satisfying*

$$\sup_{p \in S^I_{[0,T],p}} \|\Psi^\dagger_F(p) - \Psi(p; \theta)\|_{C^k([T,T+\tau];\mathbb{R}^{d_p})} \leqslant \varepsilon.$$

Proof. We know that $\Psi^\dagger_F$ exists and is continuous since the assumptions of Proposition 5.3.6 are satisfied. Recall that $S^I_{[0,T],p}$ is compact. We apply the result of Theorem 5.3.1 to an operator defined via linear homeomorphisms of Ψ_F, which we will then use to establish the approximation result in its final form. Indeed, we define the translation map $\Lambda_T : C^k([T, T+\tau]; \mathbb{R}^{d_p}) \to C^k([0, \tau]; \mathbb{R}^{d_p})$ so that

$$(\Lambda_T \varphi)(s) = \varphi(T + s), \qquad s \in [0, \tau],$$

for any $\varphi \in C^k([T, T+\tau]; \mathbb{R}^{d_p})$. We note that

$$\|\varphi\|_{C^k([T,T+\tau];\mathbb{R}^{d_p})} = \|\Lambda_T \varphi\|_{C^k([0,\tau];\mathbb{R}^{d_p})}.$$

It is also possible to define the rescaling operator $R_\tau : C^k([0, \tau]; \mathbb{R}^{d_p}) \to C^k([0, 1]; \mathbb{R}^{d_p})$ so that

$$(R_\tau \varphi)(s) = \varphi(\tau s), \qquad s \in [0, 1],$$

for any $\varphi \in C^k([0, \tau]; \mathbb{R}^{d_p})$. We similarly define the rescaling operator R_T. We note that these rescaling operators are linear homeomorphisms, preserving equivalence of norms up to multiplicative constants. We therefore define the translated and rescaled forecasting operator $\widetilde{\Psi}^\dagger_F : R_T(S^I_{[0,T],p}) \subset C^k([0, 1]; \mathbb{R}^{d_p}) \to C^k([0, 1]; \mathbb{R}^{d_p})$ defined via the composition

$$\widetilde{\Psi}^\dagger_F = R_\tau \circ \Lambda_T \circ \Psi^\dagger_F \circ R_T^{-1},$$

which is continuous by continuity of all the operators in the composition. We note that $R_T(S^I_{[0,T],p})$ is compact since R_T is a homeomorphism and therefore preserves compactness. Hence, by Theorem 5.3.1, it is possible to deduce that for any

$\delta > 0$ there exists a neural operator $\widetilde{\Psi}(\bullet, \theta) : R_T(S^I_{[0,T],p}) \subset C^k([0,1]; \mathbb{R}^{d_p}) \to C^k([0,1]; \mathbb{R}^{d_p})$ satisfying

$$\sup_{p \in R_T(S^I_{[0,T],p})} \|\widetilde{\Psi}^\dagger_F(p) - \widetilde{\Psi}(p;\theta)\|_{C^k([0,1];\mathbb{R}^{d_p})} \leqslant \delta. \tag{5.26}$$

This indeed follows from a direct application of Theorem 5.3.1 with $D = [0,1]$, $r = d_p$, $r' = d_p$, $\Psi^\dagger = \widetilde{\Psi}^\dagger_F$, and $K = R_T(S^I_{[0,T],p})$. Defining $\Psi(\bullet, \theta) : C^k([0,T]; \mathbb{R}^{d_p}) \to C^k([T, T+\tau]; \mathbb{R}^{d_p})$ via the composition

$$\Psi(\bullet, \theta) = \Lambda_T^{-1} \circ R_\tau^{-1} \circ \widetilde{\Psi}(\bullet; \theta) \circ R_T,$$

it is readily deduced that for any $p \in S^I_{[0,T],p}$ it holds that

$$\|\Psi^\dagger_F(p) - \Psi(p;\theta)\|_{C^k([T,T+\tau];\mathbb{R}^{d_p})} = C\|\widetilde{\Psi}^\dagger_F(R_T p) - \widetilde{\Psi}(R_T p;\theta)\|_{C^k([0,1];\mathbb{R}^{d_p})},$$

for a constant C depending on T and τ. Therefore, choosing $\varepsilon = \delta/C$ and applying the result established in (5.26) yields the conclusion. □

5.4 Data-Driven Approximation of Smoothing and Forecasting Maps

The theoretical developments of Section 5.3 prove the existence of, and universal approximation theorems for, maps that underpin data-driven smoothing and forecasting in data assimilation. In this section we demonstrate that it is possible to learn these maps, purely from data, in practice. In Subsection 5.4.1 we describe the experimental set-up in general, followed in Subsection 5.4.2 by a summary of the numerical results. We describe details of the experiment with the Lorenz '63 (Edward N. Lorenz, 1963b) and Lorenz '96 (Edward N Lorenz, 1996b) dynamical systems in Subsection 5.4.3 and Subsection 5.4.4, respectively; experiments with the Kuramoto-Sivashinsky equation (Kuramoto, 1978; Michelson and Sivashinsky, 1977) are contained in Subsection 5.4.4.

5.4.1 Experimental Set-Up

To train the neural operators solving the data assimilation problems we consider the following data scenario. For each dynamical system, the set $\{p^{(j)}, q^{(j)}\}_{j=1}^J$ of input-output pairs are generated using the dynamics in (5.1), given i.i.d. initial conditions $\big(p^{(j)}(0), q^{(j)}(0)\big) \sim \pi$. Measure π is computed as the pushforward, over some burn-in time, of the simpler measure π_0. For each dynamical system we consider the existence of operators $\Psi^\dagger_S : C([0,T]) \to C([0,T])$ and $\Psi^\dagger_F : C([0,T]) \to$

$C([T, T+\tau])$ defined as the mappings[1]

$$\Psi_S^\dagger : \{p(t)\}_{t\in[0,T]} \mapsto \{q(t)\}_{t\in[0,T]} \quad \Psi_F^\dagger : \{p(t)\}_{t\in[0,T]} \mapsto \{p(t)\}_{t\in[T,T+\tau]}. \quad (5.27)$$

We train transformer neural operators (Calvello, Nikola B. Kovachki, et al., 2025) to construct approximations of $\Psi_S^\dagger$ and $\Psi_F^\dagger$ picked from the parametric class

$$\Psi_S : C([0,T]) \times \Theta \to C([0,T]), \quad \Psi_F : C([0,T]) \times \Theta \to C([T,T+\tau])$$

respectively. We use the transformer neural operator based on self-attention from Calvello, Nikola B. Kovachki, et al. (2025) to approximate the $\Psi_S^\dagger$, hence solving smoothing, and to approximate $\Psi_F^\dagger$ in the context of Lorenz '63 dynamics, hence solving forecasting in this setting. We use a variant of this architecture which uses cross-attention to approximate $\Psi_F^\dagger$ in the other contexts, hence solving forecasting in those settings. Architectural details as well as the universal approximation properties for these neural operators are discussed in more depth in Section 5.B. The set Θ is assumed to be a finite dimensional parameter space from which a $\theta^* \in \Theta$ is selected so that $\Psi_\bullet(\bullet, \theta^*) \approx \Psi_\bullet^\dagger$. To define θ^* we employ a cost functional c and solve the following optimization problem:

$$\theta^* = \arg\min_{\theta\in\Theta} \mathbb{E}_\bullet \Big[c\big(\Psi_\bullet(\bullet, \theta), \Psi_\bullet^\dagger(\bullet)\big) \Big]. \quad (5.28)$$

In practice, we will only have access to the values of each input and output trajectory at a set of discretization points of the domain, which we denote as $\{t_i\}_{i=1}^N$. The self-attention based transformer neural operator that we deploy for smoothing uses the same number of grid-points for the input and output; employing a cross-attention based transformer neural operator is used to bypass this constraint on matching input and output grids for forecasting in some settings, where the input is on the time-interval $[0, T]$ and the output on $[T, T+\tau]$. We note that due to its discretization invariant properties, this architecture can be used to obtain a prediction for the output at any arbitrary discretization of $[T, T+\tau]$. For the smoothing experiments we use 6 self-attention transformer neural operator layers with 128 latent channels, for each self-attention operator, we use 8 attention heads. We also use this set-up for forecasting in the Lorenz '63 context. For the other forecasting experiments we use

[1] We note that the existence of the operators in (5.27) is not directly guaranteed by Proposition 5.3.3 and Proposition 5.3.6, as the former are defined on the whole of $C([0,T])$, rather than a local domain of existence. We note that establishing the existence of the operators in (5.27) constitutes an important avenue for future work, which we present in Section 5.5. We assume their existence for our numerical investigations.

an architecture composed of an encoder and a decoder: the encoder comprises 6 self-attention transformer neural operator layers with 1024 latent channels; the decoder on the other hand is comprised of a single layer consisting of a self-attention operator and a cross-attention operator both with 1024 latent channels.

5.4.2 Summary of Results from Numerical Experiments

We provide here a summary of the experimental results. In Table 5.1 we collect the results of the smoothing experiments for the various systems, reporting relative L^2 errors between the true unobserved trajectory and the prediction generated by the neural operator. All errors are reported on test data not used for training.

Table 5.1: Smoothing performance on Lorenz '63 (L63), Lorenz '96 (L96), and the Kuramoto–Sivashinsky equation (KSE). Relative L_2 errors are averaged over test trajectories.

System	**Avg. Rel.** L_2	**Med. Rel.** L_2	**Std. Rel.** L_2	**Min. Rel.** L_2	**Max. Rel.** L_2
L63	0.012426	0.012402	0.001612	0.008761	0.017275
L96	0.047156	0.044635	0.014569	0.025633	0.258743
KSE	0.009433	0.009322	0.000755	0.007678	0.013157

In Table 5.2 we report the results of the forecasting experiments. For each system, we report the mean, median and standard deviation of the relative L^2 errors obtained on the sample trajectories in the test set. We also report the relative improvement achieved by the neural operator prediction when compared to the constant prediction taken using the final value of the input trajectory. We note that the chaotic nature of the problems considered means that long-term trajectory forecasting is difficult. However we can compose forecasts and look at the resulting statistics of the time-series — in particular the invariant measure of the resulting stationary process; these are well-approximated. Thus, for each of the systems, we report in the respective subsections the statistics of the long-time forecasts obtained via composition of the trained neural operators. The results in Table 5.2 along with the accurate prediction of trajectory statistics demonstrate the success of using a neural operator forecasting approach.

In the following subsections, we describe the experimental set-up for the various systems and provide analysis of the numerical results.

Table 5.2: Forecasting performance on Lorenz '63 (L63), Lorenz '96 (L96), and the Kuramoto–Sivashinsky equation (KSE). Relative L_2 errors are averaged over test trajectories. Baseline corresponds to a constant forecast equal to the last input state; we report the relative improvement achieved with the transformer neural operator compared to the baseline.

System	Avg. Rel. L_2	Med. Rel. L_2	Std. Rel. L_2	Rel. Improvement (%)
L63	0.050	0.033	0.100	95.53
L96	0.033	0.032	0.004	94.38
KSE	0.046	0.045	0.007	82.68

5.4.3 Lorenz '63

We study the Lorenz '63 dynamical system (Edward N. Lorenz, 1963b) in the form

$$\dot{x} = \sigma(y - x), \tag{5.29a}$$

$$\dot{y} = x(\rho - z) - y, \tag{5.29b}$$

$$\dot{z} = xy - \beta z. \tag{5.29c}$$

For the purposes of our experiments, the parameters are set to the classical values $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$ at which the system has provable stable chaotic and statistical properties (Tucker, 1999; Holland and Melbourne, 2007). The initial conditions x_0, y_0, z_0 for training and test sets are sampled i.i.d from the uniform distributions $U([-15, 15])$, $U([-15, 15])$, and $U([0, 40])$, respectively. We use a burn-in time of 20 to ensure the trajectories are close to the attractor at time $t = 0$. For the Problem (P1) smoothing experiment we use the x trajectory on a time interval of $[0, 5]$ as input data and predict the coupled y, z trajectories over the same time interval. For practical implementation, we use points on the trajectories sampled at uniform intervals $\Delta t = 0.02$. For the Problem (P2) forecasting experiment we use the x trajectory on a time interval of $[0, 2]$ as input data and predict the coupled x trajectory over the future time $[2, 4]$. For practical implementation, we use points on the input and output trajectories sampled at uniform intervals $\Delta t = 0.01$. We train the neural operator solving the smoothing problem using 10000 training trajectories and the one solving the forecasting problem using 200000 trajectories. For both experiments we use a test set of 2000 trajectories. We note the higher data complexity required to solve the forecasting problem.

We summarize the parameter and experimental settings for the Lorenz '63 dynamical system in Table 5.3. In Figure 5.1 we demonstrate the qualitative results of the smoothing experiment. Indeed, we display the predicted trajectories corresponding

Table 5.3: Lorenz ‘63 system settings

Category	**Smoothing**	**Forecasting**
Parameters	$\sigma = 10,\ \rho = 28,\ \beta = \frac{8}{3}$	
Observed	$x \in C([0,5];\mathbb{R})$	$x \in C([0,2];\mathbb{R})$
Unobserved	$(y,z) \in C([0,5];\mathbb{R}^2)$	$(y,z) \in C([2,4];\mathbb{R}^2)$
Observation time step	$\Delta t = 0.02$	$\Delta t = 0.01$

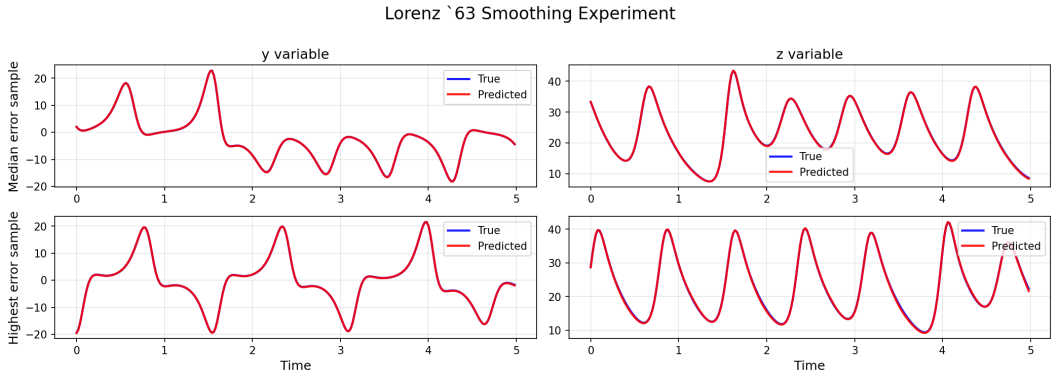


Figure 5.1: Median and worst-case relative L^2 error samples for smoothing experiment involving prediction of (y,z) from x.

to the median and highest relative L^2 errors in the test set compared to true trajectories. The panels showcase predictions that are qualitatively indistinguishable from the truth, indicating the success of the purely data-driven smoothing approach. To further demonstrate the importance of observability, we experiment with the smoothing setting of mapping the z trajectory over the time interval $[0,5]$ to the x,y trajectories over the same time interval. We use the same model and training setup as before. Figure 5.2 displays the predicted trajectories corresponding to the median and highest relative L^2 errors in the test set compared to true trajectories. The panels showcase the failure in the predictions, which achieve a median relative L^2 error of 1; it is also interesting to note that the model seems to predict the 0 trajectory, possibly due to the reflection argument discussed in Subsection 5.2.3. In Figure 5.3 we demonstrate the qualitative results of the forecasting experiment. Indeed, in Figure 5.3a we display the predictions for the trajectories corresponding to the median and highest relative L^2 errors in the test set compared to ground truths. We note that the prediction for the test sample presenting the largest error still resembles a possible trajectory of the Lorenz ‘63 system. Indeed, the prediction diverges to another lobe of the attractor; this behavior is to be expected in prediction problems in the context of chaotic systems. With the goal of obtaining forecasts on

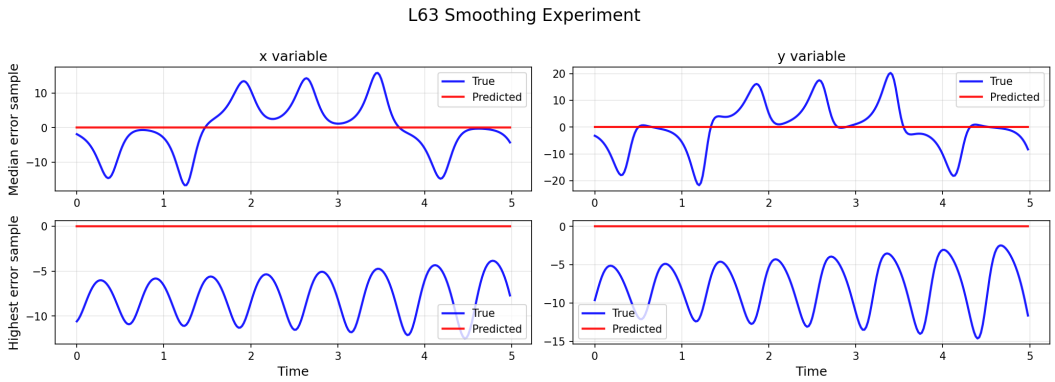
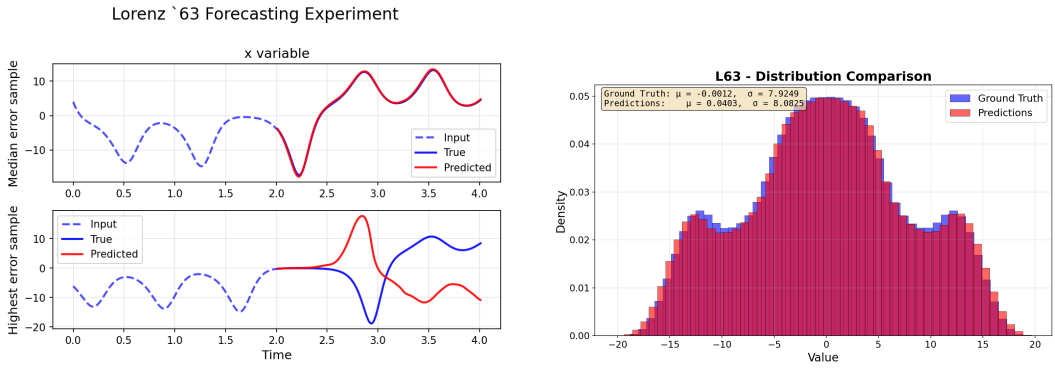


Figure 5.2: Median and worst-case relative L^2 error samples for smoothing experiment involving prediction of (x, y) from z.

longer time horizons, we experiment with composition of the learned forecasting map $\Psi_F : C([0,2]) \to C([2,4])$ to obtain $\Psi_F^n : C([0,2]) \to C([2,2+2n])$. In Figure 5.3b, for $n = 500$ we compare the distribution (histogram) of points in the predicted future trajectories $\widehat{p}$ to points in the ground truth trajectories p. The matching of distributions, and hence statistics, indicates the success in predicting valid trajectories from the attractor of the Lorenz system, despite the chaotic nature of the equations and hence per-trajectory divergence. As further validation of the quantitative success of the purely data-driven forecasting approach, we recall the numerical comparison with the prediction given by the constant value of $x(T)$ for $T = 2$, that is taking the predicted trajectory to be $\widehat{p}(t) = x(2)$ for $t \in [2,4]$. Table 5.2 shows the improvement in accuracy yielded by the cross-attention based transformer neural operator, in comparison to this naive approach.



(a) (b)

Figure 5.3: Median and worst-case relative L^2 error samples in forecasting (a). Distribution of trajectory predictions under composition of the learned forecasting map (b).

5.4.4 Lorenz '96

The Lorenz '96 model (Edward N Lorenz, 1996b) consists of a linearly damped and externally forced system equipped with an energy-conserving quadratic nonlinearity, which generates cyclic interactions among the state variables. These properties make the model a standard testbed for investigating atmospheric predictability. The system dynamics are governed by the following equations:

$$\frac{du_i}{dt} = (u_{i+1} - u_{i-2})u_{i-1} - u_i + F, \quad i = 1, 2, \ldots, d. \tag{5.30}$$

For the purposes of our experiments $d = 40$ in the smoothing setting, while $d = 8$ in the forecasting setting; furthermore, in both setups the parameter is set to $F = 8$. For all components u_i the initial conditions for training and test sets are found by first selecting initial conditions i.i.d. from $F + U([-1, 1])$, in both index i and random choice of initialization at time $t = -200$, and then simulating the system using a burn-in time of 200 to ensure the initialization of the trajectories are sampled i.i.d. from the attractor at time $t = 0$. For the smoothing experiment we take trajectories over a time interval of $[0, 5]$ as input data and predict the unobserved trajectories over the same time interval. For practical implementation, we use points on the trajectories sampled at uniform intervals $\Delta t = 0.02$. For the forecasting experiment we also use trajectories on a time interval of $[0, 5]$ as input data and predict observed trajectories over the future time $[5, 5.2]$. For practical implementation, we use points on the input and output trajectories sampled at uniform intervals $\Delta t = 0.02$. We train the neural operator solving the smoothing problem using 10000 training trajectories and the one solving the forecasting problem using 160000 trajectories. For both experiments we use a test set of 2000 trajectories. We again note the higher data complexity required to solve the forecasting problem.

We summarize the parameter and experimental settings for the Lorenz '96 dynamical system in Table 5.4. In Figure 5.4 we demonstrate the qualitative results of the

Table 5.4: Lorenz '96 system settings

Category	**Smoothing**	**Forecasting**
Parameters	$F = 8$	
Observed	$(u_1, \ldots, u_{21}, u_{23}, u_{25}, \ldots, u_{39}) \in C([0, 5]; \mathbb{R}^{30})$	$(u_1, u_2, u_3, u_5, \ldots, u_{39}) \in C([0, 5]; \mathbb{R}^{30})$
Unobserved	$(u_{22}, u_{24}, \ldots, u_{40}) \in C([0, 5]; \mathbb{R}^{10})$	$(u_1, u_2, u_3, u_5, \ldots, u_{39}) \in C([5, 5.2]; \mathbb{R}^{30})$
Obs. time	$\Delta t = 0.02$	

smoothing experiment. Indeed, we display the predicted trajectories corresponding to the median and highest relative L^2 errors in the test set compared to true

trajectories. The median error panel demonstrates near-perfect overlap between prediction and truth, while the highest error panel exhibits overlap only after $t = 1$, indicating a possible effect of synchronization (Pecora and Carroll, 1990). Together, these results demonstrate the qualitative success of the purely data-driven smoothing approach. In Figure 5.5 we display results for the forecasting experiment. Here

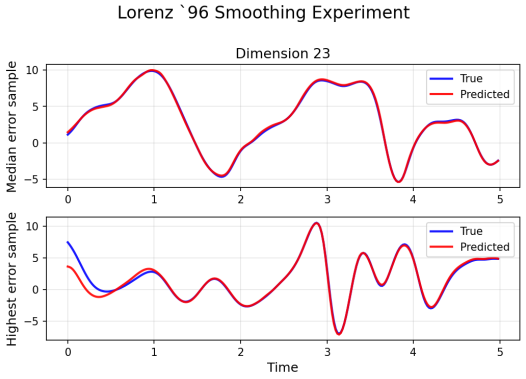


Figure 5.4: Median and worst-case performance on the Lorenz '96 system for smoothing.

we show the pointwise errors relative to the root mean square of the ground truth (computed in space) for the sample in the test set achieving the median relative L^2 error. With the goal of obtaining forecasts on longer time horizons, we experiment with composition of the learned forecasting map $\Psi_F : C([0,5]) \to C([5,5.2])$ to obtain $\Psi_F^n : C([0,5]) \to C([5,5+0.2n])$. In Figure 5.6a, we show an example of predicted trajectories obtained by composing the map with $n = 50$; the plot shows trajectory divergence after a handful of compositions, due to chaos and error compounding. However, in Figure 5.6b, for $n = 500$ we compare the distribution of points in the predicted future trajectories $\widehat{p}$ to points in the ground truth trajectories p. The matching of distributions, and hence statistics, indicates the success in predicting the invariant measure supported on the attractor of the Lorenz '96 system, despite the chaotic nature of the equations that results in trajectory divergence. As further validation of the quantitative success of the purely data-driven forecasting approach, we recall the numerical comparison with the prediction given by the constant value of $x(T)$ for $T = 5$, that is taking the predicted trajectory to be $\widehat{p}(t) = p(5)$ for $t \in [5, 5.2]$. Table 5.2 shows the improvement in accuracy yielded by the cross-attention based transformer neural operator, in comparison to this naive approach.

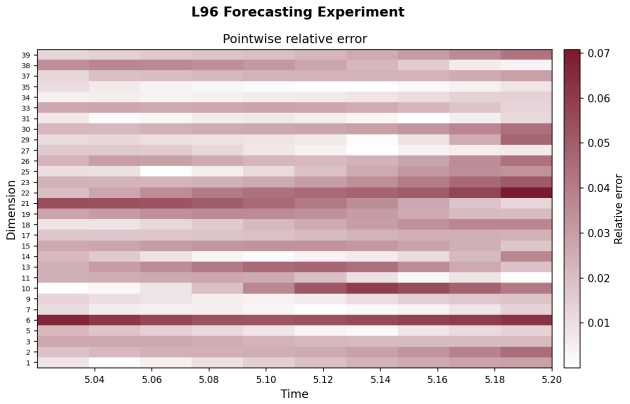
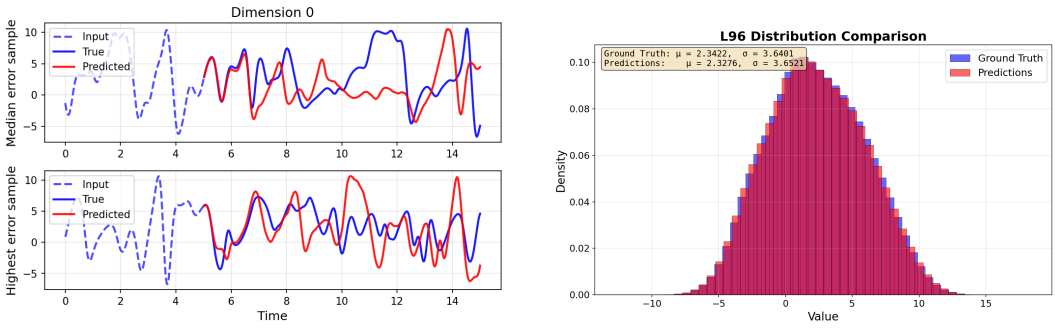


Figure 5.5: Pointwise errors relative to the root mean square of the ground truth (computed in space) for the sample in the test set achieving the median relative L^2 error.



(a) (b)

Figure 5.6: Forecast obtained by composing the learned map on one-step median and worst-case relative L^2 error samples (a). Distribution of trajectory predictions under composition of the learned forecasting map (b).

5.4.5 Kuramoto-Sivashinsky

The Kuramoto–Sivashinsky (KS) equation (Kuramoto, 1978) is a well-studied nonlinear partial differential equation that models the development of instabilities in spatially extended systems. It exhibits rich spatiotemporal chaotic behavior and arises in a variety of physical contexts, including flame front propagation, thin film flows, and reaction–diffusion phenomena. With the periodicity in space imposed to identity $x = L$ and $x = 0$, the one-dimensional KS equation for $u : [0, L] \times \mathbb{R}^+ \to \mathbb{R}$ takes the form

$$\frac{\partial u}{\partial t} + \frac{\partial^4 u}{\partial x^4} + \frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} = 0, \quad (x, t) \in (0, L) \times \mathbb{R}^+, \tag{5.31a}$$

$$u(x, 0) = u_0, \quad \forall x \in [0, L]. \tag{5.31b}$$

For the purposes of our experiments, the parameter is set to $L = 32\pi$. Numerically, the KS equation (5.31) is discretized in space using a Fourier pseudospectral method, which naturally enforces periodic boundary conditions. Time integration is carried out using the exponential time-differencing fourth-order Runge–Kutta (ETDRK4) scheme (Kassam and Trefethen, 2005). This method treats the stiff linear operator analytically, thereby avoiding the explicit linear Courant–Friedrichs–Lewy (CFL) constraint associated with fully explicit Runge–Kutta schemes. As a result, the time step is selected according to accuracy requirements and nonlinear resolution rather than linear stability limitations. We simulate the dynamics using a burn-in time of 200 to ensure the trajectories are independent and sampled from the attractor. The observed trajectories are taken to be filtered solutions of (5.31), obtained by zeroing out the high Fourier modes of u. We denote this filtered solution by $\widetilde{u}$. In the smoothing setting we experiment with, the goal is to recover the full solution u from $\widetilde{u}$, where $\widetilde{u}$ is obtained by retaining the first 64 modes of u and zeroing out the rest. In the forecasting setting, $\widetilde{u}$ is obtained by retaining the first 32 modes of u and zeroing out the rest. For the smoothing experiment we use the observed trajectories on a time interval of $[0, 100]$ as input data and predict the unobserved trajectories over the same time interval. For practical implementation, we use points on the trajectories sampled at uniform intervals $\Delta t = 0.25$. For the forecasting experiment we use the observed trajectories on a time interval of $[0, 100]$ as input data and predict the the trajectories over the future time $[100, 102]$. For practical implementation, we use points on the input and output trajectories sampled at uniform intervals $\Delta t = 0.25$. We train the neural operator solving the smoothing problem using 10000 training trajectories and the one solving the forecasting problem using 80000 trajectories. For both experiments we use a test set of 2000 trajectories. We note the higher data complexity required to solve the forecasting problem. We summarize the parameter and experimental settings for the Kuramoto-Sivashinsky equation in Table 5.5. In Figure 5.7 we demonstrate the qualitative results of the smoothing

Table 5.5: Kuramoto–Sivashinsky (KS) system settings

Category	**Smoothing**	**Forecasting**
Parameters	$L = 32\pi$	
Spatial Grid	$x_j = jL/128, \quad j = 1, \ldots, 128$	
Observed	$\big(\widetilde{u}(x_1), \ldots, \widetilde{u}(x_{128})\big) \in C([0, 100]; \mathbb{R}^{128})$	$\big(\widetilde{u}(x_1), \ldots, \widetilde{u}(x_{128})\big) \in C([0, 100]; \mathbb{R}^{128})$
Unobserved	$\big(u(x_1), \ldots, u(x_{128})\big) \in C([0, 100]; \mathbb{R}^{128})$	$\big(\widetilde{u}(x_1), \ldots, \widetilde{u}(x_{128})\big) \in C([100, 102]; \mathbb{R}^{128})$
Obs. time	$\Delta t = 0.25$	

experiment. Indeed, we display the predicted trajectories corresponding to the median and highest relative L^2 errors in the test set compared to true trajectories. The panels showcase predictions that are qualitatively indistinguishable from the truth, indicating the success of the purely data-driven smoothing approach. In

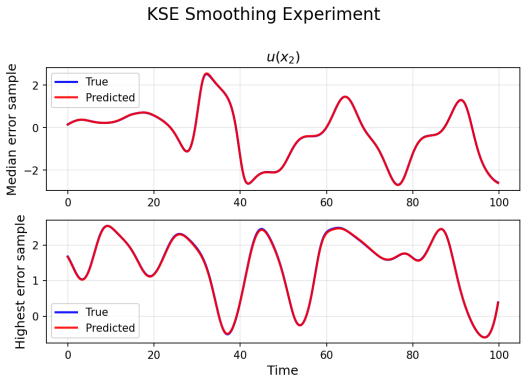


Figure 5.7: Median and worst-case performance on the KS system for smoothing.

Figure 5.8 we display results for the forecasting experiment. Here we show the pointwise errors relative to the root mean square of the ground truth (computed in space) for the sample in the test set achieving the median relative L^2 error. With the goal of obtaining forecasts on longer time horizons, we experiment with composition of the learned forecasting map $\Psi_F : C([0, 100]) \to C([100, 102])$ to obtain $\Psi_F^n : C([0, 100]) \to C([100, 100 + 2n])$. In Figure 5.9a, we show an example of predicted trajectories obtained by composing the map with $n = 100$; the plot shows trajectory divergence after a handful of compositions, due to chaos and error compounding. However, in Figure 5.9b, for $n = 1000$ we compare the distribution of points in the predicted future trajectories $\widehat{p}$ to points in the ground truth trajectories p. The matching of distributions, and hence statistics, indicates the success in predicting valid trajectories from the attractor of the KS system, despite the chaotic nature of the equations and hence per-trajectory divergence. As further validation of the qualitative success of the purely data-driven forecasting approach, we recall the numerical comparison with the prediction given by the constant value of $p(T)$ for $T = 100$, that is taking the predicted trajectory to be $\widehat{p}(t) = p(100)$ for $t \in [100, 102]$. Table 5.2 shows the improvement in accuracy yielded by the transformer neural operator, in comparison to this naive approach.

5.5 Conclusions and Future Directions

In this work we have formulated and analyzed a model for fully data-driven data assimilation, and the smoothing and forecasting problems in particular; we have also

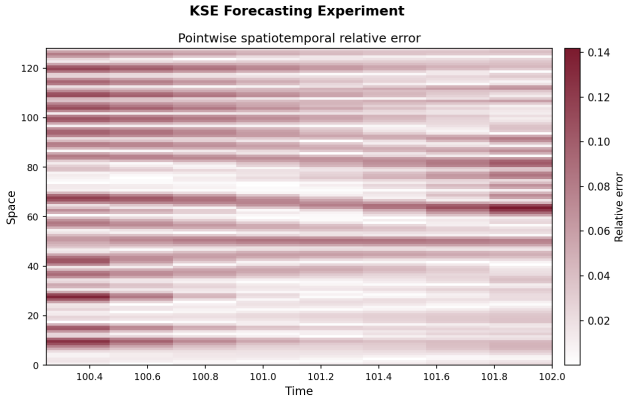
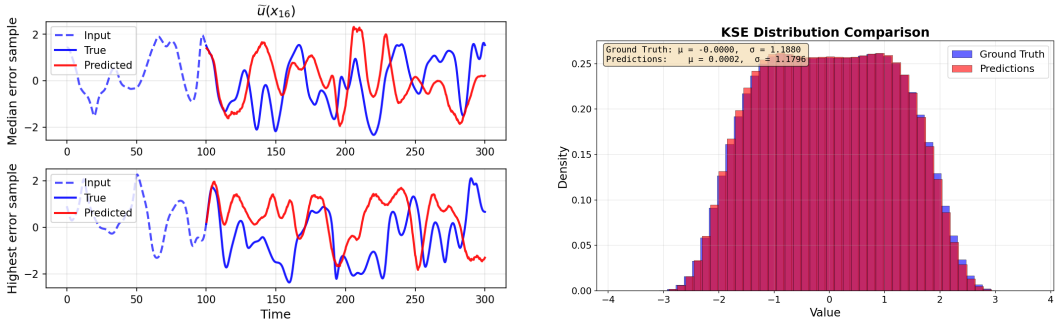


Figure 5.8: Pointwise errors relative to the root mean square of the ground truth (computed in space) for the sample in the test set achieving the median relative L^2 error.



(a) (b)

Figure 5.9: Forecast obtained by composing the learned map on one-step median and worst-case relative L^2 error samples (a). Distribution of trajectory predictions under composition of the learned forecasting map (b).

defined neural operator algorithms to tackle the problem in practice. We show that in the context of dynamical systems, under an observability-rank condition, there exists a continuous operator mapping observations to unobserved or predicted quantities. We formulate universal approximation theorems establishing the existence of neural operator parametrizations solving the smoothing and forecasting problems up to arbitrary accuracy. We have further demonstrated these capabilities in practice by deploying transformer neural operators for these problems in the context of the Lorenz '63, Lorenz '96, and Kuramoto-Sivashinsky dynamical systems.

The theory developed in this chapter represents a significant step towards accelerated, nonlinear data assimilation that is model agnostic, avoiding the need for specified dynamics and possibly expensive model evaluations. This provides, for example, a first mathematical underpinning for the development of the direct-observation

state estimators and direct-observation forecasts, which constitute the frontier of the AI-weather forecasting domain; see Allen et al. (2025) and Alexe et al. (2024), respectively. Nonetheless, many open challenges remain, and we highlight several interesting avenues for future work.

1. In Subsection 5.2.3 we show that the observability-rank condition is satisfied by the Lorenz '63 dynamical system whenever $x_0 \neq 0$. Characterizing for which points the Lorenz '96 dynamical system and Kuramoto-Sivashinsky equation satisfy this condition constitutes an avenue for further work. A similar analysis for the Navier-Stokes equation could lead to theory directly applicable to computational fluid dynamics and the atmospheric sciences.

2. Several chaotic dynamical systems of interest in the data assimilation context exhibit the property of synchronization (Pecora and Carroll, 1990). Synchronization of trajectories of coupled chaotic dynamical systems occurs, for example, in the setting of (5.1) supplemented with additive linear terms $A_p p$ and $A_q q$ for negative definite A_q, and with Lipschitz conditions on the functions f, g. An investigation of the interplay between the universal approximation properties of neural operators and the synchronization properties of the dynamics constitutes an interesting avenue for future work.

3. It is of interest to extend the local existence results in this chapter to show under a different set of assumptions the existence of an operator mapping observations to unobserved variables, defined on the whole domain of the dynamics. Applying the theory of fractal delay embeddings from Sauer, Yorke, and Casdagli (1991) constitutes a promising avenue for future work.

4. It is of interest to perform a detailed numerical study comparing the performance of different neural operator architectures when applied to the data assimilation problems presented in this chapter. Furthermore, an accuracy-complexity tradeoff analysis between neural operators and non learning-based smoothing algorithms would be of interest.

5. It is of interest to study discrete time dynamical systems from the perspective developed in this chapter, and to use this setting to make concrete connections to Takens' embedding theorem (Takens, 2006; Sauer, Yorke, and Casdagli, 1991).

5.A Observability-Rank Condition in Control Theory

In the following discussion we explain the connection between the observability-rank condition, employed in this chapter, and the observability-rank condition from control theory, introduced in the seminal paper Hermann and Krener (1977). In the general control-theoretic context, consider the partially observed dynamical system

$$\begin{aligned} \dot{x} &= \phi(x, u), \\ y &= h(x), \end{aligned} \tag{5.32}$$

where $u(t) \in \Omega \subset \mathbb{R}^l$ for each $t \geqslant 0$, $x \in M$, a C^∞ connected manifold of dimension m, $y \in \mathbb{R}^n$, and ϕ, h are C^∞ functions. For simplicity we may consider the setting where $M = \mathbb{R}^{d_p+d_q}$ and $h : \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_q}$ as follows.

Example 5.A.1. *The partially observed dynamical system* (5.1) *we consider in this chapter may be written in the form*

$$\begin{aligned} \dot{p} &= f(p, q), \\ \dot{q} &= g(p, q), \\ p &= \pi_p(p, q). \end{aligned} \tag{5.33}$$

This is a specific example of the control theoretic setting with $M = \mathbb{R}^{d_p+d_q}$, the external control u absent, $\phi = (f, g)$ and $h = \pi_p$.

We now introduce the technical notions required to define the observability-rank condition from Hermann and Krener (1977). These definitions may be found in Hermann and Krener (1977) and are developed for general C^∞ manifolds M and general C^∞ observation operators $h : \mathbb{R}^m \to \mathbb{R}^n$. Here we work in the setting of $M = \mathbb{R}^{d_p+d_q}$ and $h : \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_p}$, aligning with the Example 5.A.1, and in particular with the contents of our work.

Every point $\mathsf{u} \in \Omega$ defines a vector field $x \mapsto \phi(x, \mathsf{u}) \in C^\infty(\mathbb{R}^{d_p+d_q}; \mathbb{R}^{d_p+d_q})$; we let $\mathcal{F}^0 \subset C^\infty(\mathbb{R}^{d_p+d_q}; \mathbb{R}^{d_p+d_q})$ denote the collection of all of these vector fields, one for each $\mathsf{u} \in \Omega$. We let $\mathcal{H}^0$ denote the subset of $C^\infty(\mathbb{R}^{d_p+d_q})$ consisting of the d_p component functions $h_1, h_2, \ldots, h_{d_p} : \mathbb{R}^{d_p+d_q} \to \mathbb{R}$ of the observation map $h : \mathbb{R}^{d_p+d_q} \to \mathbb{R}^{d_p}$, viewed as scalar-valued smooth functions on $\mathbb{R}^{d_p+d_q}$. We define $\mathcal{H}$ to be the smallest linear subspace of $C^\infty(\mathbb{R}^{d_p+d_q})$ containing $\mathcal{H}^0$ and which is closed with respect to Lie differentiation under elements of $\mathcal{F}^0$. For each $\varphi \in \mathcal{H}$, the differential $\mathsf{D}\varphi(x) : \mathbb{R}^{d_p+d_q} \to \mathbb{R}$ is a linear functional on $\mathbb{R}^{d_p+d_q}$, i.e. an element of $(\mathbb{R}^{d_p+d_q})^*$. We define $\mathrm{d}\mathcal{H}^0 = \{\mathsf{D}\varphi : \varphi \in \mathcal{H}^0\}$ and $\mathrm{d}\mathcal{H} = \{\mathsf{D}\varphi : \varphi \in \mathcal{H}\}$ as

collections of maps $\mathbb{R}^{d_p+d_q} \to (\mathbb{R}^{d_p+d_q})^*$, and $\mathrm{d}\mathcal{H}(x) \subseteq (\mathbb{R}^{d_p+d_q})^*$ as the subspace obtained by evaluating elements of $\mathrm{d}\mathcal{H}$ at x.

Definition 5.A.2 (Observability-rank condition in Hermann and Krener (1977)). *The dynamical system* (5.32) *satisfies the observability-rank condition at* $\mathfrak{x}$ *if* $\dim(\mathrm{d}\mathcal{H}(\mathfrak{x})) = d_p + d_q$.

With the translation between our work and Hermann and Krener (1977), as described in Example 5.A.1, we may formulate the following proposition.

Proposition 5.A.3. *The dynamical system* (5.1) *satisfying Assumption P2 with* $f, g \in C^\infty$ *and satisfying Assumption P2 at the point* $(\mathfrak{p}, \mathfrak{q})$ *is an instance of a dynamical system* (5.32) *with a fixed control, satisfying the observability-rank condition from Hermann and Krener (1977) at the point* $\mathfrak{x} = (\mathfrak{p}, \mathfrak{q})$.

Proof of Proposition 5.A.3. We are in the specific setting of Example 5.A.1 and hence, because no control is present, the set $\mathcal{F}^0$ simply comprises the vector field $(f, g) \in C^\infty(\mathbb{R}^{d_p+d_q}; \mathbb{R}^{d_p+d_q})$. Letting $\mathfrak{x} = (\mathfrak{p}, \mathfrak{q})$ and recalling the notation for Lie derivatives defined in Section 1.2, we have that

$$\mathcal{L}_{(f,g)}(\pi_p)(\mathfrak{x}) = \frac{\partial \pi_p}{\partial x}(\mathfrak{x}) \begin{pmatrix} f(\mathfrak{x}) \\ g(\mathfrak{x}) \end{pmatrix} \tag{5.34a}$$

$$= \begin{pmatrix} I_{d_p \times d_p} & \mathbf{0}_{d_p \times d_q} \end{pmatrix} \begin{pmatrix} f(\mathfrak{p}, \mathfrak{q}) \\ g(\mathfrak{p}, \mathfrak{q}) \end{pmatrix} = f(\mathfrak{p}, \mathfrak{q}). \tag{5.34b}$$

Therefore, higher order Lie derivatives of π_p under the vector field (f, g) yield $\mathcal{L}^i_{(f,g)} f$. Let $\mathcal{G}$ be the span of the component functions $\{\pi_{p,1}, \ldots, \pi_{p,d_p}\} \cup \{\mathcal{L}^i_{(f,g)} f_j : 1 \leqslant j \leqslant d_p, 0 \leqslant i \leqslant n-1\}$, where $\pi_{p,j} : \mathbb{R}^{d_p+d_q} \to \mathbb{R}$ denotes the jth component of π_p and similarly f_j denotes the jth component of f, so that the evaluations $\{\pi_{p,j}(\mathfrak{p}, \mathfrak{q})\}_{j=1}^{d_p}$ and $\{\mathcal{L}^i_{(f,g)} f_j(\mathfrak{p}, \mathfrak{q})\}_{j=1}^{d_p}$ correspond to the $(n+1)d_p$ entries of $F^{(n)}(\mathfrak{p}, \mathfrak{q})$ in (5.5).

From (5.34) and closure of $\mathcal{H}$ under Lie differentiation under (f, g), we have that $\mathcal{G} \subseteq \mathcal{H}$; consequently, $\mathrm{d}\mathcal{H}$ contains differentials of elements of $\mathcal{G}$. The linear transformation $L : \mathbb{R}^{(n+1)d_p} \to \mathbb{R}^{d_p+d_q}$ from (5.12) defines a map

$$LF^{(n)} \colon (\mathfrak{p}, \mathfrak{q}) \mapsto [s_1(\mathfrak{p}, \mathfrak{q}), s_2(\mathfrak{p}, \mathfrak{q}), \ldots, s_{d_p+d_q}(\mathfrak{p}, \mathfrak{q})]^\top, \tag{5.35}$$

for $s_i \in \mathcal{G}$. Since $\mathrm{d}\mathcal{H}(\mathfrak{p}, \mathfrak{q}) \subseteq (\mathbb{R}^{d_p+d_q})^*$, we have that $\dim(\mathrm{d}\mathcal{H}(\mathfrak{p}, \mathfrak{q})) \leqslant d_p + d_q$. Since $s_1, \ldots, s_{d_p+d_q} \in \mathcal{G} \subseteq \mathcal{H}$, their differentials $\mathsf{D}s_1(\mathfrak{p}, \mathfrak{q}), \ldots, \mathsf{D}s_{d_p+d_q}(\mathfrak{p}, \mathfrak{q})$ are

elements of $\mathrm{d}\mathcal{H}(\mathfrak{p},\mathfrak{q})$. Moreover, these differentials are the rows of the Jacobian $\mathsf{D}(LF^{(n)})(\mathfrak{p},\mathfrak{q})$, which has rank $d_p + d_q$ by Assumption P2, and hence are linearly independent. Therefore $\dim(\mathrm{d}\mathcal{H}(\mathfrak{p},\mathfrak{q})) \geqslant d_p + d_q$, and combined with the upper bound $\dim(\mathrm{d}\mathcal{H}(\mathfrak{p},\mathfrak{q})) \leqslant d_p + d_q$, we conclude that $\dim(\mathrm{d}\mathcal{H}(\mathfrak{p},\mathfrak{q})) = d_p + d_q$. □

5.B Architectural Details

In this section we describe the transformer neural operator architectures employed in Section 5.4, with particular focus on cross-attention. The goal of this section is two-fold: to present the architectural details, but also to discuss how the architecture may be cast in the universal approximation context of Theorem 5.3.1, which is the object of this chapter. The first transformer neural operator uses self-attention. The second uses a combination of self-attention and cross-attention. The use of cross-attention enables the output of the architecture to be defined on a different number of grid points than the input. The self-attention based transformer neural operator we use is described in detail in Calvello, Nikola B. Kovachki, et al., 2025, Section 4.2. A slight modification to the architecture implemented in practice leads to a universal approximation theorem of the form we apply in this chapter (as in Theorem 5.3.1), the statement of which may be found in Calvello, Nikola B. Kovachki, et al., 2025, Theorem 22. We focus the following discussion on the cross-attention based transformer neural operator. To this end, we recall the definition of cross-attention in function space from Calvello, Nikola B. Kovachki, et al. (2025).

Let $D \subseteq \mathbb{R}^d$ and $E \subseteq \mathbb{R}^e$ be open sets. Let $u : D \to \mathbb{R}^{d_u}$ be a function, $v : E \to \mathbb{R}^{d_v}$ another function, and $x \in E$, $y \in D$ points. Then, for learnable parameters $Q \in \mathbb{R}^{d_K \times d_v}, K \in \mathbb{R}^{d_K \times d_u}, V \in \mathbb{R}^{d_V \times d_u}$, cross-attention is defined as the operator acting on functions v, u such that

$$\mathsf{C}(v,u)(x) = \mathbb{E}_{y\sim\pi(y;v,u,x)}[Vu(y)], \tag{5.36}$$

for any $x \in E$, where the probability density function is defined as $\pi(\cdot; v, u, x) : D \to \mathbb{R}^+$ so that

$$\pi(y;v,u,x) = \frac{\exp\Big(\big\langle Qv(x), Ku(y)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\int_D \exp\Big(\big\langle Qv(x), Ku(s)\big\rangle_{\mathbb{R}^{d_K}}\Big)\,\mathrm{d}s}, \tag{5.37}$$

for any $y \in D$. We note that self-attention is a special case of cross-attention where $v = u$. In the following, we denote by $\mathcal{V}, \mathcal{U}, \mathcal{W}, \mathcal{Z}$ as arbitrary function spaces; furthermore, while cross-attention allows a more general definition, here we write the architecture as a mapping between functions over the domain D. The

cross-attention based transformer neural operator we employ may be written as the map $\Psi_{\mathsf{C}}(\bullet, \bullet; \theta) : \mathcal{V}(D; \mathbb{R}^{\ell}) \times \mathcal{U}(D; \mathbb{R}^{r}) \to \mathcal{Z}(D; \mathbb{R}^{r'})$ acting on functions $v \in \mathcal{V}(D; \mathbb{R}^{\ell})$ and $u \in \mathcal{U}(D; \mathbb{R}^{r})$ such that

$$\Psi_{\mathsf{C}}(v, u; \theta) \coloneqq \Big(\mathsf{T}_{\text{out}} \circ \mathsf{D}(v, \bullet) \circ \mathsf{E}_{L} \circ \mathsf{T}_{\text{in}}\Big)(u, \theta). \tag{5.38}$$

In practice, we define $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{r}) \to \mathcal{W}(D; \mathbb{R}^{c})$ via concatenation with the grid (positional encoding) and application of a learnable linear transformation that lifts to a latent channel dimension c. The operator $\mathsf{E}_{L} : \mathcal{W}(D; \mathbb{R}^{c}) \to \mathcal{W}(D; \mathbb{R}^{c})$ is defined via a composition of L self-attention layers. Its construction is outlined in detail in Calvello, Nikola B. Kovachki, et al., 2025, Section 4. The transformer decoder $\mathsf{D} : \mathcal{V}(D; \mathbb{R}^{\ell}) \times \mathcal{W}(D; \mathbb{R}^{c}) \to \mathcal{W}(D; \mathbb{R}^{c})$ is defined for $(v, u) \in \mathcal{V}(D; \mathbb{R}^{\ell}) \times \mathcal{W}(D; \mathbb{R}^{c})$ via the iteration

$$v \hookleftarrow S_{\text{in}} v, \tag{5.39a}$$

$$v \hookleftarrow W_{1}^{\mathsf{D}} v + \mathsf{C}(v, v), \tag{5.39b}$$

$$v \hookleftarrow \mathsf{F}_{\text{LayerNorm}}(v), \tag{5.39c}$$

$$v \hookleftarrow W_{2}^{\mathsf{D}} v + \mathsf{C}(v, u), \tag{5.39d}$$

$$v \hookleftarrow \mathsf{F}_{\text{LayerNorm}}(v), \tag{5.39e}$$

$$v \hookleftarrow W_{3}^{\mathsf{D}} v + \mathsf{F}_{\text{NN}}(v), \tag{5.39f}$$

$$v \hookleftarrow \mathsf{F}_{\text{LayerNorm}}(v), \tag{5.39g}$$

where S_{in} is a learnable linear transformation lifting to the channel dimension, $\mathsf{F}_{\text{LayerNorm}}$ denotes application of layer normalization, F_{NN} a two layer multilayer perceptron and in practice, and multihead cross-attention is employed; we refer to Calvello, Nikola B. Kovachki, et al., 2025, Section 4 for a discussion on multihead attention. We also note that in our implementation, we fix the linear transformation $W_{1}^{\mathsf{D}}, W_{2}^{\mathsf{D}}, W_{3}^{\mathsf{D}}$ to be the identity. For implementation purposes, we define the operator $\mathsf{T}_{\text{out}} : \mathcal{W}(D; \mathbb{R}^{c}) \to \mathcal{Z}(D; \mathbb{R}^{r'})$ as a linear transformation projecting to the output space.

Due to the definition of cross-attention and step (5.39d) in the decoder, the discretization size of the output function corresponds to the discretization size of the input function v, making this architecture particularly attractive for forecasting problems, where the output time interval may be of length differing to the input. In practical application of this architecture, the v is fixed to be the rescaled identity function, sampled at the grid points where the output function is to be evaluated. Since the v is fixed, for the approximation theory developed in the following section we only

require consideration of the operator $\Psi(v, \bullet; \theta) : \mathcal{U}(D; \mathbb{R}^r) \to \mathcal{Z}(D; \mathbb{R}^{r'})$. Because the architecture is the discrete version of the continuum neural operator, the scheme is discretization invariant in both inputs, meaning that the parameters will be independent of any discretization of input and output functions; the resulting scheme is thus deployable at any discretization. In the next subsection we state and prove a universal approximation theorem for a slight variant of the scheme implemented in practice.

5.B.1 Universal Approximation for Cross-Attention

Consider activation functions $\sigma \in C^\infty(\mathbb{R})$ which are non-polynomial and Lipschitz continuous. We consider neural operators of the form (5.38) where $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^r) \to \mathcal{W}(D; \mathbb{R}^c)$ and $\mathsf{T}_{\text{out}} : \mathcal{W}\big(D; \mathbb{R}^c\big) \to \mathcal{Z}\big(D; \mathbb{R}^{d_z}\big)$ are defined by neural networks of the form

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(x) = R_2\sigma\Big(R_1\big(u(x), x\big) + b_R\Big) + b'_R, \tag{5.40}$$

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) = P_2\sigma\Big(P_1\big(v(x), x\big) + b_P\Big) + b'_P, \tag{5.41}$$

where $\big(u(x), x\big) \in \mathbb{R}^{r+d}$ and $\big(v(x), x\big) \in \mathbb{R}^{c+d}$, where R_1, R_2, P_1, P_2 are learned linear transformations of appropriate dimensions and b_R, b'_R, b_P, b'_P are learned vectors. We define the operator $\mathsf{E} : \mathcal{W}\big(D; \mathbb{R}^c\big) \to \mathcal{W}\big(D; \mathbb{R}^c\big)$ as a two-step map acting on its inputs $u \in \mathcal{W}(D; \mathbb{R}^c)$ as

$$u(x) \leftarrow\!\shortmid W_1^{\mathsf{E}} u(x) + \mathsf{C}\big(u, u; Q^{\mathsf{E}}, K^{\mathsf{E}}, V^{\mathsf{E}}\big)(x), \tag{5.42a}$$

$$u(x) \leftarrow\!\shortmid W_2^{\mathsf{E}} u(x) + W_3^{\mathsf{E}}\sigma\big(W_4^{\mathsf{E}} u(x) + b_1^{\mathsf{E}}\big) + b_2^{\mathsf{E}}, \tag{5.42b}$$

for any $x \in D$. Note that the layer thus defined does not include layer normalizations, and hence is a variant of the self-attention transformer layer employed in practice. We also define a no-layer normalization variant of the decoder in (5.39), which also includes a residual of the output of the encoder after application of cross-attention; indeed we consider $\mathsf{D} : \mathcal{V}(D; \mathbb{R}^\ell) \times \mathcal{W}(D; \mathbb{R}^c) \to \mathcal{W}(D; \mathbb{R}^c)$ defined for $(v, u) \in \mathcal{V}(D; \mathbb{R}^\ell) \times \mathcal{W}(D; \mathbb{R}^c)$ via the iteration

$$v(x) \leftarrow\!\shortmid S_{\text{in}} v(x) \tag{5.43a}$$

$$v(x) \leftarrow\!\shortmid W_1^{\mathsf{D}} v(x) + \mathsf{C}(v, v; Q_1^{\mathsf{D}}, K_1^{\mathsf{D}}, V_1^{\mathsf{D}})(x), \tag{5.43b}$$

$$v(x) \leftarrow\!\shortmid W_2^{\mathsf{D}}\big(v(x), u(x)\big) + \mathsf{C}(v, u; Q_2^{\mathsf{D}}, K_2^{\mathsf{D}}, V_2^{\mathsf{D}})(x), \tag{5.43c}$$

$$v(x) \leftarrow\!\shortmid W_3^{\mathsf{D}} v(x) + W_4^{\mathsf{D}}\sigma\big(W_5^{\mathsf{D}} v(x) + b_1^{\mathsf{D}}\big) + b_2^{\mathsf{D}}, \tag{5.43d}$$

for any $x \in D$. We may now apply the result of Lanthaler, Zongyi Li, and Andrew M. Stuart (2025) to show two universal approximation theorems for the resulting cross-attention based transformer neural operator.

Theorem 5.B.1. *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary, and fix integers $s, s' \geqslant 0$. If $\Psi^\dagger : C^s\big(\overline{D}; \mathbb{R}^r\big) \to C^{s'}\big(\overline{D}; \mathbb{R}^{r'}\big)$ is a continuous operator and $K \subset C^s\big(\overline{D}; \mathbb{R}^r\big)$ a compact set, then for any $\varepsilon > 0$, there exists a cross-attention based transformer neural operator $\Psi(v, \bullet; \theta) : K \subset C^s\big(\overline{D}; \mathbb{R}^r\big) \to C^{s'}\big(\overline{D}; \mathbb{R}^{r'}\big)$ so that*

$$\sup_{u \in K} \left\| \Psi^\dagger(u) - \Psi(v, u; \theta) \right\|_{C^{s'}} \leqslant \varepsilon. \tag{5.44}$$

Proof. We begin by noting that for $Q_2^{\mathsf{D}}, K_2^{\mathsf{D}} = 0$ and $V_2^{\mathsf{D}} = I$, the cross-attention mapping employed in the decoder reduces to

$$\mathsf{C}\big(v, u; Q_2^{\mathsf{D}}, K_2^{\mathsf{D}}, V_2^{\mathsf{D}}\big)(\bullet) = \mathbb{E}_{y \sim \pi(y; v, u, \bullet)}[V_2^{\mathsf{D}} u(y)] = \frac{1}{|D|} \int u(x)\, \mathrm{d}x. \tag{5.45}$$

For encoder weights $V^{\mathsf{D}} = 0$, $W_3^{\mathsf{E}} = 0$, $W_1^{\mathsf{E}} = W_2^{\mathsf{E}} = I$ and decoder weights $W_2^{\mathsf{D}} = \big(0, W\big)$, $W_3^{\mathsf{D}} = 0$, $W_4^{\mathsf{D}} = W_5^{\mathsf{D}} = I$ and $b_2^{\mathsf{D}} = 0$, the composition of encoder (5.42) and decoder (5.43) reduces to the mapping

$$u(\bullet) \mapsto \sigma\left(W u(\bullet) + b_1 + \frac{1}{|D|} \int u(x)\, \mathrm{d}x \right). \tag{5.46}$$

The existence of $W, R_1, R_2, P_1, P_2, b_1, b_R, b'_R, b_P, b'_P$ so that $\Psi(v, \bullet; \theta)$ satisfies (5.44) then follows from Lanthaler, Zongyi Li, and Andrew M. Stuart, 2025, Theorem 2.1, which also involves the application of the universality result for two-layer neural networks of Pinkus, 1999, Theorem 4.1. □

Chapter 6

TRANSFORMERS FOR NONLINEAR FILTERING

6.1 Introduction

Filtering, determining the conditional distribution of the state of a partially and noisily observed dynamical system given data collected sequentially in time, constitutes a longstanding computational challenge, particularly when the state is high-dimensional. Sequential data assimilation encompasses a broad range of methods aimed at finding the filtering distribution, or simply estimating the state, in an online fashion. The focus of this work is on the use of sequential data assimilation methods for filtering and state estimation, and in particular the methodologies developed primarily in the geophysical sciences, and weather prediction especially, which are relevant in high dimensions. The overarching goal is to introduce a new framework for the *machine learning* of data assimilation (DA) algorithms, and specifically ensemble filters, which facilitates the sharing of parameters between implementations with different ensemble sizes; we may, for example, train and deploy the same model with different ensemble sizes. To achieve this we work in the context of mean-field state-space formulations of filtering. This perspective enables us to consider the idea of learning algorithms which share a common set of parameters, regardless of ensemble size, by viewing large ensembles as approximating a common mean field limit. In practice, small ensemble sizes behave differently from large ensemble sizes, an issue that has heretofore been addressed by use of inflation and localization. Here we show that an efficient fine-tuning strategy can be used to transfer parameters trained at one ensemble size to another ensemble size, at which the algorithm can be deployed. The resulting algorithms are based on stochastic dynamical systems for a state variable whose evolution depends on its own law; particle approximation leads to implementable algorithms. We show how the attention mechanism, adapted from the transformer methodology at the heart of large language models, can be used to define probability measure-dependent transport maps, i.e. mappings that move probability mass from one distribution to another, that can be approximated by empirical measures using arbitrary ensemble sizes. We refer to the resulting class of neural network architectures as *measure neural mappings*.

In Subsection 6.1.1, we set our work in the context of pre-existing literature. Then,

we describe the specific contributions we make to achieve our overarching goal, and we overview the contents of the chapter in Subsection 6.1.2.

6.1.1 Literature Review

Data assimilation (DA) is a fundamental framework for integrating observational data with numerical models for the purpose of state estimation and forecasting. DA methods systematically combine observations with prior model forecasts, accounting for their respective uncertainties, to produce optimal estimates of the system state or to characterize a probability distribution over the state. Several textbooks provide comprehensive overviews of this field (Andrew H. Jazwinski, 1970; Kody Law, A. Stuart, and Kostas Zygalakis, 2015; Asch, Marc Bocquet, and Nodet, 2016; Reich and Colin Cotter, 2015; Evensen, Vossepoel, and Leeuwen, 2022; Bach, Baptista, Sanz-Alonso, et al., 2025).

The Kalman filter Kalman (1960) is a foundational sequential data assimilation algorithm for linear systems with Gaussian errors. The extended Kalman filter, using linearization, adapts the Kalman methodology to nonlinear systems, but its direct application to high-dimensional geophysical problems is computationally prohibitive. The ensemble Kalman filter (EnKF) (Evensen, 2003; Burgers, Van Leeuwen, and Evensen, 1998; Jeffrey L Anderson, 2001) addresses this limitation through a Monte Carlo approach, representing statistics via a limited ensemble of model states. Square-root variants, including the ensemble transform Kalman filter and the ensemble adjustment Kalman filter (Jeffrey L Anderson, 2001; Tippett et al., 2003), reduce both storage requirements and sampling errors through deterministic formulations. Stochastic EnKF variants update each ensemble member using the Kalman filter equation with random perturbations in the innovation; Van Leeuwen (2020a) provides a consistent derivation that favors perturbing the modeled observations. However, EnKF-based methods commonly encounter challenges in practical implementations and, in particular, suffer from sampling errors caused by finite ensemble sizes. To address these limitations, covariance inflation (Jeffrey L. Anderson and S. L. Anderson, 1999; Jeffrey L. Anderson, 2007) and localization (Brian R. Hunt, Kostelich, and Szunyogh, 2007) techniques have been developed. Approaches such as the iterative EnKF (Pavel Sakov, Dean S. Oliver, and Laurent Bertino, 2012) and the maximum likelihood ensemble filter (Zupanski, 2005) further improve performance in nonlinear regimes.

The integration of machine learning methodologies into DA has recently emerged

as a powerful strategy to enhance predictive accuracy and scalability beyond traditional approaches constrained by Gaussian assumptions. Such machine learning approaches are reviewed in Cheng et al. (2023b) and Bach, Baptista, Sanz-Alonso, et al. (2025). A key direction has been the direct learning of filters from data. Learning a fixed-gain filter was considered in Hoang, De Mey, and Talagrand (1994), Mallia-Parfitt and Bröcker (2016), Levine and A. Stuart (2022), and Bach, Baptista, Luk, et al. (2025). McCabe et al. (McCabe and Brown, 2021) developed an ensemble filter that uses the mean and covariance matrix as input to a recurrent neural network. Bocquet et al. (Marc Bocquet, Farchi, et al., 2024) further demonstrated the efficacy of neural network-based filtering schemes without relying on an ensemble, achieving comparable accuracy to ensemble methods by identifying key dynamical perturbations directly from forecast states.

Complementary to these direct learning strategies, a distinct class of approaches formulates ensemble filtering as a transport problem. These methods explicitly construct transformations that map forecast distributions onto analysis distributions. Representative techniques include Knothe–Rosenblatt rearrangements (Spantini, Baptista, and Youssef Marzouk, 2022b), normalizing flows (Chipilski, 2025), bridge matching (Shi et al., 2022), and optimal transport frameworks (Al-Jarrah et al., 2023). These methods offer rigorous probabilistic interpretations and enable accurate sampling from the filtering distribution at each assimilation step. Typically these approaches find transformations acting on each individual particle, rather than operating directly at the distributional level, and are not designed to directly transform or update the ensemble members collectively.

Beyond per-particle transformations, a complementary line of research updates the ensemble collectively via particle flows (Van Leeuwen, Hans R Künsch, et al., 2019a), with applications to atmospheric models (Hu, Van Leeuwen, and Jeffrey L Anderson, 2024). Early particle-flow methods cast the analysis as a continuous ordinary differential equation that drives particles toward the posterior (Daum and Huang, 2011), with related continuous-time ensemble formulations such as the EnKF–Bucy perspective (Bergemann and Reich, 2012). Variational or Stein-based mappings are considered to push forward the prior through learned maps, e.g., the variational mapping particle filter (Pulido and Leeuwen, 2018).

Probabilistic formulations provide another promising direction for learning-based filters. These approaches aim to directly approximate the Bayesian inference problem of obtaining the filtering distribution (Bröcker, Engster, and Parlitz, 2009;

Boudier et al., 2023; Bach, Baptista, Luk, et al., 2025; Bach, Baptista, Sanz-Alonso, et al., 2025). In particular, recent works have proposed frameworks to jointly learn the forecast and analysis steps (Boudier et al., 2023), or assume knowledge of the dynamics and learn parameterized analysis maps via variational inference (Bach, Baptista, Luk, et al., 2025).

Building on the use of deep learning architectures for ensemble filtering, recent efforts have incorporated permutation-invariant neural networks to respect the unordered nature of ensembles. Such architectures enable models to learn interactions among ensemble members without being sensitive to their ordering, which is essential for consistent filter behavior across runs. These designs have been applied to directly learn ensemble update rules (Zhou, Z.-R. Liu, and Xiao, 2024) and to refine ensemble forecasts in post-processing settings (Höhlein et al., 2024).

Transformers and the underlying attention mechanism (Vaswani et al., 2017) are now ubiquitous in machine learning. Their success at modeling global, long-range correlations has made this architecture an attractive methodology for operator learning; see for example Cao (2021), Zijie Li, Meidani, and Farimani (2023), Calvello, Nikola B. Kovachki, et al. (2025), and Wang et al. (2025b). In this chapter we are interested in the attention mechanism as a map between metric spaces of probability measures, thus going beyond the definition on Euclidean spaces from Vaswani et al. (2017) and the generalization to Banach spaces in Calvello, Nikola B. Kovachki, et al. (2025). Recent developments in the mathematical analysis of transformers have led to the formulation of the continuum limit of self-attention layers (Geshkovski, Letrouit, et al., 2024). In Geshkovski, Letrouit, et al. (2024) the authors describe the evolution under this continuum limit architecture as the solution of a continuity equation; see Castin et al. (2025) for an overview on this perspective. This formulation is also employed in Geshkovski, Rigollet, and Ruiz-Balet (2024) to analyze self-attention dynamics as a measure-to-measure map. However, this analysis is restricted to measures defined on the sphere and does not include cross-attention. In this chapter we present a general methodological framework for neural network architectures on the metric space of probability measures and a general definition of attention as a measure-to-measure map. We will leverage this formulation to build implementable methodologies effecting data assimilation.

6.1.2 Contributions and Overview

Our five primary contributions are as follows:

1. We present a framework for learning ensemble filtering algorithms, based on a mean-field state-space formulation of the evolution of the filtering distribution.

2. We introduce a novel generalization of neural operators to maps defined to act between metric spaces of probability measures. We call such maps *measure neural mappings* (MNM). We identify an architecture to implement such maps, based on the attention mechanism.

3. The MNM architecture is used to define approximations of the mean-field formulation of the filtering distribution. This leads to a novel learning-based framework for ensemble methods: the *measure neural mapping enhanced ensemble filter* (MNMEF).

4. Because of the mean-field underpinnings, the basic learned parameters are shared between filters with different ensemble sizes. As a consequence, parameterizations learned at one ensemble size can be deployed at different ensemble sizes, leading to efficient training at small ensemble sizes. To enhance this approach, we introduce a fine-tuning methodology to incorporate a small number of parameters, such as localization and inflation, which intrinsically depend on ensemble size, into the resulting filters.

5. We demonstrate that the resulting filtering algorithm outperforms an optimized local ensemble transform Kalman filter (LETKF, a leading ensemble filter) on Lorenz '96, Kuramoto-Sivashinsky, and Lorenz '63 models, for both small and larger ensemble sizes.

In Section 6.2 we describe filtering from the perspective of mean-field dynamics and define a learning framework to approximate the true filter, addressing Contribution 1. Section 6.3 is devoted to the introduction of *measure neural mappings*, a generalization of neural operators to maps taking probability measures as inputs; we show in this section that the attention mechanism can be used to build a transformer architecture defined on probability measures, thereby tackling Contribution 2. Contributions 3 and 4 are addressed in Section 6.4, where we detail the algorithm for finite particle sizes. The numerical experiments in Section 6.5 support Contribution 5. We conclude in Section 6.6. We also note that a reader focused on methodological contributions alone can skip the self-contained Section 6.3.

6.2 Learning Mean-field Filters

In this section, we formulate the filtering problem (Subsection 6.2.1), describing sequential DA from the perspective of nonlinear evolution of probability densities (Subsection 6.2.2) and from the mean-field state-space evolution perspective (Subsection 6.2.3). We then present the general idea of learning an analysis map which has a probability measure as an input (Subsection 6.2.4). Finally, we introduce our learning-based mean-field filter approach, which enhances traditional ensemble Kalman filtering by incorporating trainable correction terms in the mean-field equations (Subsection 6.2.5). This allows us to extend and, as will show in numerical results presented in the penultimate section, improve upon the standard EnKF methodology, which relies on a Gaussian ansatz; in so-doing we maintain the interpretable structure of Kalman filtering.

6.2.1 Filtering Set-Up

Consider the stochastic dynamics model given by

$$v_{j+1}^\dagger = \Psi(v_j^\dagger) + \xi_j^\dagger, \quad j \in \mathbb{Z}^+, \tag{6.1a}$$

$$v_0^\dagger \sim \mathsf{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathsf{N}(0, \Sigma) \text{ i.i.d.}, \tag{6.1b}$$

where $\Psi : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ is the forward dynamic model, and $v_j^\dagger, \xi_j^\dagger \in \mathbb{R}^{d_v}$ for $j \in \mathbb{Z}^+$. We assume that the sequence $\{\xi_j^\dagger\}_{j\in\mathbb{Z}^+}$ is independent of initial condition $v_0^\dagger$; this is often written as $\{\xi_j^\dagger\}_{j\in\mathbb{Z}^+} \perp v_0^\dagger$. The data model is given by

$$y_{j+1}^\dagger = h(v_{j+1}^\dagger) + \eta_{j+1}^\dagger, \quad j \in \mathbb{Z}^+, \tag{6.2a}$$

$$\eta_j^\dagger \sim \mathsf{N}(0, \Gamma) \text{ i.i.d.}, \tag{6.2b}$$

where $h : \mathbb{R}^{d_v} \to \mathbb{R}^{d_y}$ is the observation operator and $y_j^\dagger, \eta_j^\dagger \in \mathbb{R}^{d_y}$ for $j \in \mathbb{Z}^+$. We assume that the sequence $\{\eta_j^\dagger\}_{j\in\mathbb{N}} \perp v_0^\dagger$ and that the two sequences in the dynamics and data models are independent of one another: $\{\xi_j^\dagger\}_{j\in\mathbb{Z}^+} \perp \{\eta_j^\dagger\}_{j\in\mathbb{N}}$. The states $v_j^\dagger$ lie in $\mathbb{R}^{d_v}$, while the observations $y_j^\dagger$ lie in $\mathbb{R}^{d_y}$. Note that there is a hidden parameter Δt specifying the time interval between steps j and $j+1$. It is implicitly embedded in (6.1) through Ψ, which represents the discrete-time map obtained by integrating the continuous dynamics over an interval of length Δt.

Remark 6.2.1. *We use † for all variables defining the state and observation, and the noises that define them. This is to distinguish them from other similar variables, appearing without †, in the algorithms that follow. In this chapter, we adopt autonomous dynamics with additive Gaussian noises according to* (6.1) *and* (6.2).

There are several extended settings, briefly outlined in Remark 6.2.2. These extensions are important in applications but introduce substantial technical challenges. We do not consider those settings in this chapter for clarity of exposition.

Remark 6.2.2 (Possible extensions beyond the baseline model). *Beyond the autonomous, additive-Gaussian setting in* (6.1)–(6.2)*, one may consider Bain and Crisan (2009), Andrew M Stuart (2010b), and Reich and Colin Cotter (2015):*

- ***Time-dependent dynamics/observations.*** *Allow Ψ, Σ, h, and Γ to vary with j. This can destroy stationarity and leads to non-homogeneous transition/-likelihood kernels.*
- ***Multiplicative or non-Gaussian noises.*** *Replace additive Gaussian errors by state-dependent or heavy-tailed noises. This can induce heteroscedastic innovations and non-quadratic likelihoods.*
- ***Infinite-dimensional state spaces.*** *Model v_j in a function space (e.g., a separable Hilbert space) for PDE-governed systems. One must address well-posedness (e.g., trace-class noise/covariances), operator-valued updates, and consistency under spatial discretization/mesh refinement.*

There are two primary types of problems in data assimilation: the filtering problem and the smoothing problem. In this chapter, we focus on the filtering problem. To this end we define the accumulated data up to time j by $Y_j^\dagger := \{y_1^\dagger, \dots, y_j^\dagger\}$. Using this notation we state the filtering problem in Definition 6.2.3, and the state estimation problem that stems from it in Definition 6.2.4.

Definition 6.2.3 (The Filtering Problem). *The filtering problem refers to identifying and sequentially updating the probability densities $\pi_j(v_j^\dagger) := \mathbb{P}(v_j^\dagger | Y_j^\dagger)$ in $\mathcal{P}(\mathbb{R}^{d_v})$ for $j \in [J]$. These densities π_j are referred to as the filtering distributions at time j.*

Definition 6.2.4 (State Estimation). *The state estimation problem refers to estimating the state $v_j^\dagger$, based on the given observations $Y_j^\dagger$, and to updating the estimate sequentially over a series of time indices $j \in [J]$.*

State estimation can be viewed as a subproblem of the filtering problem. Given the filtering distribution π_j, a natural state estimate is the conditional expectation $\mathbb{E}[v_j^\dagger | Y_j^\dagger]$ as this delivers the minimum mean-squared error estimate of $v_j^\dagger$. In this

chapter we explicitly focus on *state estimation*: our objective is to recover the hidden state $v_j^\dagger$ at each time step rather than to approximate the full filtering distribution π_j. Accordingly, our loss functions are defined with respect to the true state. We note that, in high-dimensional settings, the conditional mean is generally inaccessible in practice—accurate computation would require particle filters with prohibitively large ensembles—so it is standard to evaluate and tune filters against the true state. As outlined in Section 6.6, our framework can be extended to the filtering problem by adopting distribution-matching losses to learn the filtering distribution (and hence its mean).

6.2.2 Filtering: Probability Evolution

We consider the filtering problem, which consists of identifying the filtering distribution given by

$$\pi_j(v_j^\dagger) := \mathbb{P}(v_j^\dagger | Y_j^\dagger). \tag{6.3}$$

We define the following two measures:

$$\widehat{\pi}_{j+1}(v_{j+1}^\dagger) = \mathbb{P}(v_{j+1}^\dagger | Y_j^\dagger), \tag{6.4a}$$

$$\rho_{j+1}(v_{j+1}^\dagger, y_{j+1}^\dagger) = \mathbb{P}(v_{j+1}^\dagger, y_{j+1}^\dagger | Y_j^\dagger). \tag{6.4b}$$

The update $\pi_j \to \pi_{j+1}$ can be written in the following three steps:

Predict (linear):	$\widehat{\pi}_{j+1} = \mathsf{P}\pi_j,$	(6.5a)
Extend to data/state space (linear):	$\rho_{j+1} = \mathsf{Q}\widehat{\pi}_{j+1},$	(6.5b)
Analysis (nonlinear):	$\pi_{j+1} = \mathsf{B}(\rho_{j+1}; y_{j+1}^\dagger).$	(6.5c)

P is a linear operator that transforms the filtering distribution at time j to the forecast distribution at time $j+1$, and is given for the stochastic dynamics model in (6.1) by

$$\mathsf{P}\pi(v) = \frac{1}{\sqrt{(2\pi)^{d_v} \det \Sigma}} \int \exp\left(-\frac{1}{2}\|v - \Psi(u)\|_\Sigma^2\right) \pi(u)\, du. \tag{6.6}$$

Q is also a linear operator, one that multiplies the forecast distribution by the observation likelihood, and is given by

$$\mathsf{Q}\pi(v, y) = \frac{1}{\sqrt{(2\pi)^{d_y} \det \Gamma}} \exp\left(-\frac{1}{2}\|y - h(v)\|_\Gamma^2\right) \pi(v). \tag{6.7}$$

Then, the nonlinear operator B can be written in the form

$$\mathsf{B}(\rho_{j+1}; y_{j+1}^\dagger)(v) = \frac{\rho_{j+1}(v, y_{j+1}^\dagger)}{\int_{\mathbb{R}^{d_v}} \rho_{j+1}(u, y_{j+1}^\dagger)\, du}. \tag{6.8}$$

6.2.3 Filtering: State-Space Evolution

Directly evolving the filtering distribution π_j is generally intractable since the nonlinear operator B (6.8) cannot be computed explicitly for most practical applications. Alternatively, we consider a random state evolution process governed by its own law, given by

$$\text{Predict:} \qquad \widehat{v}_{j+1} = \Psi(v_j) + \xi_j, \quad \xi_j \sim \mathsf{N}(0, \Sigma), \tag{6.9a}$$

$$\text{Extend to observation:} \qquad \widehat{y}_{j+1} = h(\widehat{v}_{j+1}) + \eta_{j+1}, \quad \eta_{j+1} \sim \mathsf{N}(0, \Gamma), \tag{6.9b}$$

$$\text{Analysis:} \qquad v_{j+1} = \mathfrak{B}(\widehat{v}_{j+1}, \widehat{y}_{j+1}; y^\dagger_{j+1}, \rho_{j+1}), \tag{6.9c}$$

where ρ_{j+1} is the joint measure defined by (6.4b). From the definitions of P (6.6) and Q (6.7), and from equations (6.9a) and (6.9b), it follows that $(\widehat{v}_{j+1}, \widehat{y}_{j+1})$ is distributed as ρ_{j+1}, i.e., $\rho_{j+1} = \mathrm{Law}\big((\widehat{v}_{j+1}, \widehat{y}_{j+1})\big)$. Now, using the theory of transport, we choose $\mathfrak{B}$ so that the pushforward of ρ_{j+1} under this map delivers the desired conditioning on the observed data $y^\dagger_{j+1}$:

$$\mathsf{B}(\rho_{j+1}; y^\dagger_{j+1}) = \mathfrak{B}(\,\bullet\,, \bullet\,, ; y^\dagger_{j+1}, \rho_{j+1})_\sharp(\rho_{j+1}).$$

It follows that if $\pi_j = \mathrm{Law}(v_j)$ then $\pi_{j+1} = \mathrm{Law}(v_{j+1})$.

The preceding construct is mathematically appealing but it leaves open the question of how to identify an appropriate choice of $\mathfrak{B}$. The mean-field ensemble Kalman filter (Calvello, Reich, and Andrew M. Stuart, 2025) leaves (6.9a) and (6.9b) intact but replaces the mapping $\mathfrak{B}$ in (6.9c) by the affine (in $(\widehat{v}_{j+1}, \widehat{y}_{j+1})$) approximation

$$v_{j+1} = \mathfrak{B}_{\mathrm{EnKF}}(\widehat{v}_{j+1}, \widehat{y}_{j+1}; y^\dagger_{j+1}, \rho_{j+1}), \tag{6.10}$$

defined by

$$v_{j+1} = \widehat{v}_{j+1} + K_{j+1}\left(y^\dagger_{j+1} - \widehat{y}_{j+1}\right), \tag{6.11a}$$

$$K_{j+1} = \widehat{C}^{vh}_{j+1}\left(\widehat{C}^{hh}_{j+1} + \Gamma\right)^{-1}. \tag{6.11b}$$

Here, K_{j+1} is called the Kalman gain, $\widehat{C}^{vh}$ is the covariance between $\widehat{v}_{j+1}$ and $h(\widehat{v}_{j+1})$, and $\widehat{C}^{hh}_{j+1}$ is the covariance for $h(\widehat{v}_{j+1})$. Both $\widehat{C}^{vh}$ and $\widehat{C}^{hh}_{j+1}$ are computed under $\widehat{\pi}_{j+1}$:

$$\widehat{C}^{vh}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}_{j+1}}\left[\left(\widehat{v} - \overline{v}_{j+1}\right) \otimes \left(h(\widehat{v}) - \overline{h}_{j+1}\right)\right], \tag{6.12a}$$

$$\widehat{C}^{hh}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}_{j+1}}\left[\left(h(\widehat{v}) - \overline{h}_{j+1}\right) \otimes \left(h(\widehat{v}) - \overline{h}_{j+1}\right)\right]. \tag{6.12b}$$

Here, $\overline{v}_{j+1}$ and $\overline{h}_{j+1}$ are the mean of states and observations respectively, under $\widehat{\pi}_{j+1}$:

$$\overline{v}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}_{j+1}}\widehat{v}, \quad \overline{h}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}_{j+1}}h(\widehat{v}). \tag{6.13}$$

We note that in computation of these covariances we have simply used expectations under $\widehat{\pi}_{j+1}$, the marginal of ρ_{j+1} on the state. It is possible to use the replacement

$$\widehat{C}^{yy}_{j+1} = \widehat{C}^{hh}_{j+1} + \Gamma,$$

where $\widehat{C}^{yy}_{j+1}$ is the covariance for $\widehat{y}_{j+1}$. Hence we write the map $\mathfrak{B}_{\mathrm{EnKF}}$ as dependent on ρ_{j+1}, which covers both cases.

What is sometimes termed the stochastic ensemble Kalman filter is implemented in practice by employing an interacting particle system approximation of the mean-field model. The methodology uses the empirical measure defined by the set of N particles $\{v_j^{(\ell)}\}_{\ell=1}^N$ to approximate the filtering distribution $\pi_j = \mathrm{Law}(v_j)$, and hence the empirical measure defined by $\left\{\widehat{v}_{j+1}^{(\ell)}\right\}_{\ell=1}^N$ and $\left\{\left(\widehat{v}_{j+1}^{(\ell)}, \widehat{y}_{j+1}^{(\ell)}\right)\right\}_{\ell=1}^N$, to approximate $\widehat{\pi}_{j+1}$ and ρ_{j+1}, respectively. We write the resulting approximations as

$$\widehat{\pi}^{(N)}_{j+1} := \frac{1}{N}\sum_{n=1}^N \delta_{\widehat{v}^{(n)}_{j+1}}, \quad \rho^{(N)}_{j+1} := \frac{1}{N}\sum_{n=1}^N \delta_{(\widehat{v}^{(n)}_{j+1}, \widehat{y}^{(n)}_{j+1})}. \tag{6.14}$$

Making this particle approximation in (6.9), and replacing $\mathfrak{B}$ by $\mathfrak{B}_{\mathrm{EnKF}}$ leads to the interacting particle system

$$\widehat{v}^{(n)}_{j+1} = \Psi(v^{(n)}_j) + \xi^{(n)}_j, \tag{6.15a}$$

$$\widehat{y}^{(n)}_{j+1} = h(\widehat{v}^{(n)}_{j+1}) + \eta^{(n)}_{j+1}, \tag{6.15b}$$

$$v^{(n)}_{j+1} = \mathfrak{B}_{\mathrm{EnKF}}\big(\widehat{v}^{(n)}_{j+1}, \widehat{y}^{(n)}_{j+1}; y^\dagger_{j+1}, \rho^{(N)}_{j+1}\big). \tag{6.15c}$$

This is a particular instance of the ensemble Kalman filter (EnKF), where use of the empirical approximation $\rho^{(N)}_{j+1}$ means that the covariances $\widehat{C}^{vh}_{j+1}$ and $\widehat{C}^{hh}_{j+1}$ given by (6.12) are now computed by use of empirical approximation $\widehat{\pi}^{(N)}_{j+1}$ of $\widehat{\pi}_{j+1}$:

$$\widehat{C}^{vh}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}^{(N)}_{j+1}}\left[(\widehat{v} - \overline{v}_{j+1}) \otimes \left(h(\widehat{v}) - \overline{h}_{j+1}\right)\right] = \frac{1}{N}\sum_{n=1}^N \left(\widehat{v}^{(n)}_{j+1} - \overline{v}_{j+1}\right) \otimes \left(h(\widehat{v}^{(n)}_{j+1}) - \overline{h}_{j+1}\right), \tag{6.16a}$$

$$\widehat{C}^{hh}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}^{(N)}_{j+1}}\left[\left(h(\widehat{v}) - \overline{h}_{j+1}\right) \otimes \left(h(\widehat{v}) - \overline{h}_{j+1}\right)\right] = \frac{1}{N}\sum_{n=1}^N \left(h(\widehat{v}^{(n)}_{j+1}) - \overline{h}_{j+1}\right) \otimes \left(h(\widehat{v}^{(n)}_{j+1}) - \overline{h}_{j+1}\right). \tag{6.16b}$$

Here $\overline{v}_{j+1}$ and $\overline{h}_{j+1}$, originally defined in (6.13), are also computed by use of an empirical approximation of $\widehat{\pi}_{j+1}$:

$$\overline{v}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}^{(N)}_{j+1}}\widehat{v} = \frac{1}{N}\sum_{n=1}^{N}\widehat{v}^{(n)}_{j+1}, \qquad \overline{h}_{j+1} = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}^{(N)}_{j+1}}h(\widehat{v}) = \frac{1}{N}\sum_{n=1}^{N}h(\widehat{v}^{(n)}_{j+1}). \tag{6.17a}$$

Remark 6.2.5. *The ensemble Kalman filter exhibits permutation invariance with respect to the ensemble members. This important property stems from the fact that both the sample means* $\overline{v}_{j+1}$, $\overline{h}_{j+1}$ *in* (6.17) *and the empirical covariance matrices* $\widehat{C}^{vh}_{j+1}$, $\widehat{C}^{hh}_{j+1}$ *in* (6.16) *are calculated as averages over all ensemble members, making them invariant to the ordering of particles in the ensemble. This permutation invariance is a crucial property that motivates the design of our methodology.*

6.2.4 Learning Probability Measure–Dependent Analysis Maps

Machine learning–based approaches can be applied to approximate the analysis step (6.5c). Indeed, one may seek an operator $\mathsf{B}_{\mathrm{MNM}}$ (*measure neural mappings, or MNMs, will be introduced in Section 6.3) acting on probability measures so that*

$$\mathsf{B}_{\mathrm{MNM}}(\rho_{j+1}; y^{\dagger}_{j+1}) \approx \mathsf{B}(\rho_{j+1}; y^{\dagger}_{j+1}). \tag{6.18}$$

This aim motivates a novel and significant extension of neural operators (N. Kovachki et al., 2023b; Nikola B Kovachki, Lanthaler, and Andrew M Stuart, 2024): designing operator architectures that act on the space of probability measures. We refer to these neural operators that act on probability measures as *measure neural mappings* (MNM). A natural way to construct such an approximation is by neural operator $\mathfrak{B}_{\mathrm{MNM}}$, acting on the joint space of state and data and parameterized by the observation and by the law of the state, so that

$$\mathfrak{B}_{\mathrm{MNM}}(\widehat{v}_{j+1}, \widehat{y}_{j+1}; y^{\dagger}_{j+1}, \rho_{j+1}) \approx \mathfrak{B}(\widehat{v}_{j+1}, \widehat{y}_{j+1}; y^{\dagger}_{j+1}, \rho_{j+1}), \tag{6.19a}$$

$$\mathsf{B}_{\mathrm{MNM}}(\rho_{j+1}; y^{\dagger}_{j+1}) = \mathfrak{B}_{\mathrm{MNM}}(\bullet, \bullet; y^{\dagger}_{j+1}, \rho_{j+1})\sharp\rho_{j+1} \approx \mathsf{B}(\rho_{j+1}; y^{\dagger}_{j+1}). \tag{6.19b}$$

The map $\mathfrak{B}_{\mathrm{MNM}}$ in (6.19a) acts on the joint state–observation space; in contrast the map $\mathsf{B}_{\mathrm{MNM}}$ in (6.19b), defined through pushforward, acts on the space of probability measures on the state space. The quantities after the semi-colon parameterize these maps. To use this neural operator approximation in the state-space evolution, we replace (6.9c) by the approximation

$$v_{j+1} = \mathfrak{B}_{\mathrm{MNM}}(\widehat{v}_{j+1}, \widehat{y}_{j+1}; y^{\dagger}_{j+1}, \rho_{j+1}). \tag{6.20}$$

6.2.5 EnKF-Inspired Probability Measure–Dependent Analysis Maps

We now propose a specific novel filtering methodology, which we coin as the *measure neural mapping enhanced ensemble filter* (MNMEF). In this section, we will discuss the mean-field formulation of this approach. We consider a specific form of $\mathfrak{B}_{\mathrm{MNM}}$ inspired by the success of the EnKF and describe it here in the mean-field limit. We generalize (6.11) to take the form

$$v_{j+1} = \widehat{v}_{j+1} + (K_\theta)_{j+1}\left(y^\dagger_{j+1} - \widehat{y}_{j+1}\right), \tag{6.21a}$$

$$(K_\theta)_{j+1} = K_\theta\left(\rho_{j+1}, y^\dagger_{j+1}\right), \tag{6.21b}$$

where $K_\theta\left(\rho_{j+1}, y^\dagger_{j+1}\right)$ is a parameterized $\mathbb{R}^{d_v \times d_y}$-valued function, and θ denotes the vector of trainable parameters. Our proposed choice for $\mathfrak{B}_{\mathrm{MNM}}$ (6.21a) introduces inductive bias by mimicking the affine EnKF update. The following proposed form for $K_\theta(\bullet, \bullet)$ is identical at each time step; for simplicity we omit the subscript $j+1$ in its definition and, in particular, use $\widehat{\pi}$ as the distribution of the predicted state. We assume the following form for K_θ, generalizing the Kalman gain (6.11b):

$$K_\theta = K^{(1)}_\theta \left(K^{(2)}_\theta + \Gamma\right)^{-1}. \tag{6.22}$$

Here $K^{(1)}_\theta$ and $K^{(2)}_\theta$ are modified versions of $\widehat{C}^{vh}$ and $\widehat{C}^{hh}$ from (6.12). Specifically we have

$$K^{(1)}_\theta = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}}\left[(\widehat{v} - \overline{v} + \widehat{w}_\theta) \otimes \left(h(\widehat{v}) - \overline{h} + \widehat{z}_\theta\right)\right], \tag{6.23a}$$

$$K^{(2)}_\theta = \mathbb{E}^{\widehat{v}\sim\widehat{\pi}}\left[\left(h(\widehat{v}) - \overline{h} + \widehat{z}_\theta\right) \otimes \left(h(\widehat{v}) - \overline{h} + \widehat{z}_\theta\right)\right], \tag{6.23b}$$

where $\widehat{w}_\theta$ and $\widehat{z}_\theta$ are trainable correction terms that depend on the state $\widehat{v}$, observation $h(\widehat{v})$, true observation $y^\dagger$, and the joint distribution ρ_h. We apply a neural operator $\mathfrak{F}(\cdot;\theta) : \mathbb{R}^{d_v} \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \times \mathcal{P}(\mathbb{R}^{d_b} \times \mathbb{R}^{d_y}) \to \mathbb{R}^{d_v} \times \mathbb{R}^{d_y}$ to define the correction terms as

$$\mathfrak{F}\left(\widehat{v}, h(\widehat{v}), y^\dagger, \rho_h; \theta\right) = (\widehat{w}_\theta, \widehat{z}_\theta), \tag{6.24}$$

where $\rho_h = \mathrm{Law}\big((\widehat{v}, h(\widehat{v}))\big)$ is the joint distribution of $(\widehat{v}, h(\widehat{v}))$. As well as being defined in the mean-field limit, this form of the model can both be learned and deployed through interacting particle system approximations, just as for the original ensemble Kalman filter. These interacting particle systems, as well as details of the architecture employed for $\mathfrak{F}$ in the particle approximation, are defined in Section 6.4.

Remark 6.2.6 (Factoring Our Γ). *We note that K_θ (6.21b) depends on the probability measure ρ while $\mathfrak{F}$ (6.24) depends on ρ_h. The relationship between ρ and ρ_h*

is given by

$$(\widehat{v}, \widehat{y}) \sim \rho \;\Leftrightarrow\; \begin{cases} (\widehat{v},\, h(\widehat{v})) \sim \rho_h, \\ \eta \sim \mathcal{N}(0, \Gamma), \quad \eta \perp \widehat{v}, \\ \widehat{y} = h(\widehat{v}) + \eta. \end{cases} \tag{6.25}$$

By factoring out the noise covariance in the term $(K_\theta^{(2)} + \Gamma)^{-1}$ *in* (6.22)*, we use the explicit knowledge of* Γ *in derivation of* K_θ*. Therefore,* $\mathfrak{F}$ *uses the input* ρ_h *which does not depend on* Γ*.*

Remark 6.2.7 (Beyond Gaussian Assumptions). *The mean-field map* $\mathfrak{B}_{MNM}$ (6.21a) *is constrained to be affine in both inputs* $\widehat{v}$ *and* $\widehat{y}$*. Indeed,*

$$\mathfrak{B}_{MNM}(\widehat{v}, \widehat{y}; y^\dagger, \rho) = \widehat{v} + K_\theta \left(y^\dagger - \widehat{y} \right).$$

We note that this is the same constraint satisfied by the ensemble Kalman filter; see Calvello, Reich, and Andrew M. Stuart (2025). However, as shown in Calvello, Reich, and Andrew M. Stuart (2025), the gain K in the ensemble Kalman filter is such that the resulting state estimate has law with first and second-order moments matching a Gaussian approximation of the true filter. Our approach is more general. Indeed, since K_θ *is trainable, unlike in the EnKF, the gain is not constrained to satisfy any moment-matching with a Gaussian approximation of the true filter. The ensemble Kalman filter possesses statistical guarantees for problems involving filtering distributions that are close to Gaussian (given, for example, by dynamics and observation operators that are close to linear) (Calvello, Monmarché, et al., 2026). Further work will involve investigating whether this more general methodology possesses theoretical guarantees for a broader class of filtering problems than the EnKF.*

The core of our proposed approach is to learn a neural operator $\mathfrak{F}$, as given in (6.24), which takes both probability measures, as well as vectors, as inputs. Section 6.3 is a self-contained description of a novel methodology for training neural operators that take probability measures as inputs; it leverages the attention mechanism (Vaswani et al., 2017). In Section 6.4 we will build on this novel neural operator framework to provide a concrete methodology for enhanced data assimilation when actioned using interacting particle approximations of the mean-field limit.

6.3 Learning Maps on the Space of Probability Measures

In this section we introduce a general framework to define neural network architectures acting on the space of probability measures. Let Ω_1, Ω_2 denote two sets,

equipped with sigma algebras so that we may assign probabilities, and consider maps of the form $\mathsf{F} : \mathfrak{P}(\Omega_1) \times \Theta \to \mathfrak{P}(\Omega_2)$; here $\Theta \subseteq \mathbb{R}^p$. Our goal is to define the parametric class of functions so that, by choice of $\theta^* \in \Theta$, we may approximate a given map $\mathsf{F}^\dagger : \mathfrak{P}(\Omega_1) \to \mathfrak{P}(\Omega_2)$ by $\mathsf{F}(\cdot; \theta^*) : \mathfrak{P}(\Omega_1) \to \mathfrak{P}(\Omega_2)$. As the desired size of the approximation error shrinks we will typically require Θ, and in particular the number of parameters p, to grow. Such methodology is now well-established for maps between Banach spaces; see the review Nikola B Kovachki, Lanthaler, and Andrew M Stuart (2024). Defining such maps between metric spaces in general, and probability spaces in particular, is an unexplored research question which we address here. We refer to neural networks on the space of probability measures as *measure neural mappings* (MNM).

In what follows, we show that the popular transformer architecture may be formulated as a mean-field mapping on the space of probability measures; this mapping is mean-field because it not only acts on but is also parametrized by probability measures. We call such a neural network architecture the *transformer measure neural mapping* (TMNM). This architecture will serve as a parametric family of operators of the form $\mathsf{F} : \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}(\mathbb{R}^{d_w}) \times \Theta \to \mathfrak{P}(\mathbb{R}^{d_v})$ defined via action on $\mu \in \mathfrak{P}(\mathbb{R}^{d_u})$ and $\pi \in \mathfrak{P}(\mathbb{R}^{d_w})$ as

$$\mathsf{F}(\mu, \pi; \theta) = \mathfrak{F}(\bullet\,; \pi, \theta)_\sharp \mu. \tag{6.26}$$

The transport map in (6.26) is a mapping of the form $\mathfrak{F} : \mathbb{R}^{d_u} \times \mathfrak{P}(\mathbb{R}^{d_w}) \times \Theta \to \mathbb{R}^{d_v}$. For ease of exposition we will henceforth omit the θ parameterizations of the operators. In Section 6.4 we will show how this transformer measure neural mapping, in particular the transport map that defines it, may be used to implement (6.24) in the context of learning an ensemble-based data assimilation algorithm. For this section, however, we work quite generally, without specific reference to data assimilation.

Remark 6.3.1 (Training a Transformer MNM). *In practice, implementing MNM architectures such as that in* (6.26) *involves working with samples from the measures* $\mu \in \mathfrak{P}(\mathbb{R}^{d_u})$ *and* $\pi \in \mathfrak{P}(\mathbb{R}^{d_w})$ *rather than the measures themselves. This finite sample approximation introduces the challenge of selecting an appropriate loss function that is readily computed from samples. For the purposes of this chapter, we will employ a loss function based on matching the mean of the output measure under the map. Future work will optimize based on the use of metrics on probability measures and scoring rules, not just the mean; see Section 6.4 for more details.*

In what follows we assume access to samples $u(i) \sim \mu$ for $i = 1, \dots, N$ and similarly $w(j) \sim \pi$ for $j = 1, \dots, M$. We note the following useful identity, which follows from (6.26) *(dropping explicit $\theta-$dependence):*

$$\mathsf{F}\Big(\frac{1}{N}\sum_{i=1}^{N}\delta_{u(i)}, \pi\Big) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathfrak{F}\left(u(i);\pi\right)}. \tag{6.27}$$

This characterization will be used to describe implementable algorithms.

In view of Remark 6.3.1 we proceed by describing the TMNM architecture in detail as a mapping between spaces of probability measures. This mean-field perspective is more general and justifies the deployment of MNM architectures trained with a finite collection of samples to ensembles of different sizes. Nonetheless, in the development that follows, we provide examples illustrating the finite sample empirical setting such as in Remark 6.3.1, which will be the focus of later sections.

We proceed in Subsection 6.3.1 by describing the TMNM architecture as a composition of specific maps on probability measures, which we call attention blocks and which involve the application of the attention map. In Subsection 6.3.2 we define attention as a map on measures and show how, via application of this map to empirical measures, it is possible to recover the formulation of attention on Euclidean spaces from Vaswani et al. (2017).

6.3.1 Transformer Measure Neural Mapping

The TMNM architecture as an operator of the form $\mathsf{F} : \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}(\mathbb{R}^{d_w}) \to \mathfrak{P}(\mathbb{R}^{d_u})$ consists of a composition of $N_e \in \mathbb{N}$ number of operators $\Sigma_w : \mathfrak{P}(\mathbb{R}^{d_w}) \to \mathfrak{P}(\mathbb{R}^{d_w})$, $N_d \in \mathbb{N}$ number of operators $\Sigma_u : \mathfrak{P}(\mathbb{R}^{d_u}) \to \mathfrak{P}(\mathbb{R}^{d_u})$, and $\mathsf{C} : \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}(\mathbb{R}^{d_w}) \to \mathfrak{P}(\mathbb{R}^{d_u})$, which we call the self-attention and cross-attention blocks, respectively. We define F via action on inputs $\mu \in \mathfrak{P}(\mathbb{R}^{d_u})$ and $\pi \in \mathfrak{P}(\mathbb{R}^{d_w})$ as

$$\mathsf{F}(\mu, \pi) = \Sigma_{u,N_d+N_e}(\bullet) \circ \cdots \circ \Sigma_{u,1+N_e}(\bullet) \circ \mathsf{C}(\mu, \bullet) \circ \Sigma_{w,N_e}(\bullet) \circ \cdots \circ \Sigma_{w,2}(\bullet) \circ \Sigma_{w,1}(\pi). \tag{6.28}$$

For ease of exposition, for operators Σ we will henceforth drop the notation $\Sigma_\bullet$, as the dimension of the input state space will be clear from context. We note that the cross-attention block is intimately related to the pooling multihead attention block from the set transformer architecture (J. Lee et al., 2019). In Remark 6.3.2 and in Section 6.4 we make explicit the connection between the TMNM and the set transformer. Both operators Σ and C will be defined via a pushforward operation

under a transport map $\mathfrak{C}$: we define

$$\Sigma(\mu) = \mathfrak{C}(\bullet; \mu)_\sharp \mu, \tag{6.29a}$$

$$\mathsf{C}(\mu, \pi) = \mathfrak{C}(\bullet; \pi)_\sharp \mu. \tag{6.29b}$$

We now outline the construction of the transport map appearing in (6.29a) and (6.29b). Indeed, we may write the map $\mathfrak{C} : \mathbb{R}^{d_u} \times \mathfrak{P}(\mathbb{R}^{d_w}) \to \mathbb{R}^{d_u}$ acting on the inputs $u \in \mathbb{R}^{d_u}$ and $\pi \in \mathfrak{P}(\mathbb{R}^{d_w})$ as the composition of the maps

$$u \hookleftarrow \mathfrak{F}^{\mathrm{LN}}\big(u + \mathfrak{A}(u; \pi)\big), \tag{6.30a}$$

$$u \hookleftarrow \mathfrak{F}^{\mathrm{LN}}\big(u + \mathfrak{F}^{\mathrm{NN}}(u)\big). \tag{6.30b}$$

We note that each of the maps appearing in the composition in (6.30) may be used to define a pushforward operator on measures.

Remark 6.3.2 (Set Transformer Architecture). *We note that the composition in* (6.28) *is a generalization of the set transformer architecture from J. Lee et al. (2019) to the continuum. Indeed, the set transformer acts on a collection of vectors* $\{u(1), \dots, u(N)\} \subset \mathbb{R}^{d_u}$ *which we may assume to be drawn from an underlying reference measure* μ*. The set transformer relies on learning a vector* $w \in \mathbb{R}^{d_w}$ *with which cross-attention is applied to the ensemble. The set transformer architecture may thus be viewed as the map* F *in* (6.28) *when acting on the empirical measure* $\mu^N = \frac{1}{N}\sum_{i=1}^N \delta_{u(i)}$ *and a learned dirac* δ_w*.*

We will now turn our focus to defining each of the maps in (6.30). The map $\mathfrak{F}^{\mathrm{LN}} : \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ defines layer normalization and is such that

$$\Big(\mathfrak{F}^{\mathrm{LN}}(u; \gamma, \beta)\Big)_k = \gamma_k \cdot \frac{u_k - m(u)}{\sqrt{\sigma^2(u) + \varepsilon}} + \beta_k, \tag{6.31}$$

for $k = 1, \dots, d_u$, any $u \in \mathbb{R}^{d_u}$, where the subscript notation $(\bullet)_k$ is used to denote the k'th entry of the vector. In equation (6.31), $\varepsilon \in \mathbb{R}^+$ is a fixed parameter, $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters and m, σ are defined as

$$m(u) = \frac{1}{d_u}\sum_{k=1}^{d_u} u_k, \qquad \sigma^2(u) = \frac{1}{d_u}\sum_{k=1}^{d_u}\big(u_k - m(u)\big)^2, \tag{6.32}$$

for any $u \in \mathbb{R}^{d_u}$. The operator $\mathfrak{F}^{\mathrm{NN}} : \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ is defined such that

$$\mathfrak{F}^{\mathrm{NN}}(u; W_1, W_2, b_1, b_2) = W_2 f\big(W_1 u + b_1\big) + b_2, \tag{6.33}$$

for any $u \in \mathbb{R}^{d_u}$, where $W_1, W_2 \in \mathbb{R}^{d_u \times d_u}$ and $b_1, b_2 \in \mathbb{R}^{d_u}$ are learnable parameters and where f is a nonlinear activation function.

The map $\mathfrak{A} : \mathbb{R}^{d_u} \times \mathfrak{P}(\mathbb{R}^{d_w}) \to \mathbb{R}^{d_u}$ is the attention map: Subsection 6.3.2 which follows is dedicated to its formulation as a transport defining the attention map on measures A.

6.3.2 Attention as a Map on Measures

We present the definition of the attention operator as a map between a product of the spaces of probability measures $\mathfrak{P}(\mathbb{R}^{d_u})$ and $\mathfrak{P}(\mathbb{R}^{d_w})$ into another space of probability measures $\mathfrak{P}(\mathbb{R}^{d_v})$ [1]. If considering the map as acting on measures defined on the unbounded domain $\mathbb{R}^{d_w}$ we must make assumptions on the input measures themselves, i.e., tail decay assumptions, so that the attention operator is well-defined. Hence, for ease of exposition we first present attention as an operator $\mathsf{A} : \mathfrak{P}(\Omega_u) \times \mathfrak{P}(\Omega_w) \to \mathfrak{P}(\mathbb{R}^{d_v})$, where Ω_u, Ω_w are bounded open subsets of $\mathbb{R}^{d_u}, \mathbb{R}^{d_w}$ respectively; we will then extend the definition to the general case $\mathsf{A} : \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}_\delta(\mathbb{R}^{d_w}) \to \mathfrak{P}(\mathbb{R}^{d_v})$, where the subset of measures $\mathfrak{P}_\delta(\mathbb{R}^{d_w}) \subset \mathfrak{P}(\mathbb{R}^{d_w})$ consisting of all probability measures whose tails decay strictly faster than exponential, will be appropriately defined.

Definition 6.3.3 (Attention on Measures: Bounded State Space). *We define the operator $\mathsf{A} : \mathfrak{P}(\Omega_u) \times \mathfrak{P}(\Omega_w) \to \mathfrak{P}(\mathbb{R}^{d_v})$ via its action on inputs $(\mu, \pi) \in \mathfrak{P}(\Omega_u) \times \mathfrak{P}(\Omega_w)$ as a pushforward of the first input measure $\mu \in \mathfrak{P}(\Omega_u)$ by a transport map $\mathfrak{A} : \Omega_u \times \mathfrak{P}(\Omega_w) \to \mathbb{R}^{d_v}$ which is parametrized by the second input measure $\pi \in \mathfrak{P}(\Omega_w)$, hence*

$$\mathsf{A}(\mu, \pi) = \mathfrak{A}(\cdot; \pi)_\sharp \mu. \tag{6.34}$$

The transport map $\mathfrak{A} : \Omega_u \times \mathfrak{P}(\Omega_w) \to \mathbb{R}^{d_v}$ takes as input a vector $u \in \Omega_u$ and a probability measure $\pi \in \mathfrak{P}(\Omega_w)$ and computes an expectation under a rescaling of the measure π, which is parametrized by u. Namely for $(u, \pi) \in \Omega_u \times \mathfrak{P}(\Omega_w)$ it holds that

$$\mathfrak{A}(u; \pi) = \mathbb{E}_{w \sim p(\cdot; u, \pi)} V w, \tag{6.35}$$

where

$$p(dw; u, \pi) = \frac{\exp\big(\langle Qu, Kw \rangle\big)\pi(dw)}{\int_{\Omega_w} \exp\big(\langle Qu, Kz \rangle\big)\pi(dz)}. \tag{6.36}$$

[1] In this self-contained subsection we use the notation d_v to denote the dimension associated with the output space of probability measures. This is not to be confused with the dimension of the state of the dynamical system (6.1) that is subject of the majority of the remainder of the chapter.

Dimensions of the learnable matrices Q, K, V are determined from the above, provided Q, K have the same output dimension, which is a free parameter to be specified.

Remark 6.3.4 ($p(\bullet; u)$ is a probability measure). *Clearly, the normalization constant in* (6.36) *is finite, as the integral is taken over Ω_w, a bounded domain. Furthermore p integrates to* 1 *over Ω_w, hence it defines a probability measure.*

In view of Remark 6.3.4, in order to extend the definition of the attention operator expressed in (6.34) to a map on probability measures defined on an unbounded domain, extra care is required to ensure the normalization constant in (6.36) is finite. Indeed, we must assume that the measure $\pi \in \mathfrak{P}(\mathbb{R}^{d_w})$ has sufficiently fast tail decay in order to compensate for the exponential growth in the integrand. To this end, we introduce the following subset of $\mathfrak{P}(\mathbb{R}^{d_w})$, on which the attention operator will be shown to be well-defined. Intuitively, the condition that defines the subset is satisfied by all measures whose tails decay faster than exponentially.

Definition 6.3.5 (Set of Measures with Exponential Tail Decay). *For $\delta > 0$ we define $\mathcal{P}_\delta(\mathbb{R}^{d_u}) \subset \mathcal{P}(\mathbb{R}^{d_u})$ as a set of measures satisfying the following exponential tail decay condition:*

$$\mathcal{P}_\delta(\mathbb{R}^{d_u}) = \left\{\mu \in \mathcal{P}(\mathbb{R}^{d_u}) : \ \mu\{\omega \in \Omega : |X(\omega)| \geqslant t\} \leqslant 2\exp\left(-t^{1+\delta}\right)\right\}.$$

Given Definition 6.3.5, we may now define the attention operator as a map on probability measures defined over unbounded state spaces.

Definition 6.3.6 (Attention on Measures: Unbounded State Space). *We define the operator $\mathsf{A} : \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}_\delta(\mathbb{R}^{d_w}) \to \mathfrak{P}(\mathbb{R}^{d_v})$ via its action on inputs $(\mu, \pi) \in \mathfrak{P}(\mathbb{R}^{d_u}) \times \mathfrak{P}_\delta(\mathbb{R}^{d_w})$ as a pushforward of the first input measure $\mu \in \mathfrak{P}(\mathbb{R}^{d_u})$ by a transport map $\mathfrak{A} : \mathbb{R}^{d_u} \times \mathfrak{P}_\delta(\mathbb{R}^{d_w}) \to \mathbb{R}^{d_v}$ which is parametrized by the second input measure $\pi \in \mathfrak{P}_\delta(\mathbb{R}^{d_w})$, hence*

$$\mathsf{A}(\mu, \pi) = \mathfrak{A}(\cdot; \pi)_\sharp \mu. \tag{6.37}$$

The transport map $\mathfrak{A} : \mathbb{R}^{d_u} \times \mathfrak{P}_\delta(\mathbb{R}^{d_w}) \to \mathbb{R}^{d_v}$ takes as input a vector $u \in \mathbb{R}^{d_u}$ and a probability measure $\pi \in \mathfrak{P}_\delta(\mathbb{R}^{d_w})$ and involves computation of an expectation under a rescaling of the measure π which is parametrized by u. Namely for $(u, \pi) \in \mathbb{R}^{d_u} \times \mathfrak{P}_\delta(\mathbb{R}^{d_w})$ it holds that

$$\mathfrak{A}(u; \pi) = \mathbb{E}_{w \sim p(\cdot; u, \pi)} Vw, \tag{6.38}$$

where

$$p(dw;u,\pi) = \frac{\exp\big(\langle Qu, Kw\rangle\big)\pi(dw)}{\int_{\mathbb{R}^{d_w}} \exp\big(\langle Qu, Kz\rangle\big)\pi(dz)}. \tag{6.39}$$

Definition 6.3.5 allows to state and prove the following lemma regarding the finiteness of the normalization constant appearing in (6.39).

Lemma 6.3.7. *Let Q, K, V be fixed matrices of appropriate dimensions. The measure $p(dw;u,\pi)$ is a well-defined probability measure for $\pi \in \mathcal{P}_\delta(\mathbb{R}^{d_w})$.*

Proof. It suffices to show $\int_{\mathbb{R}^{d_w}} \exp\big(\langle Qu, Kz\rangle\big)\pi(dz) < \infty$. Using the Cauchy–Schwarz and Young's inequalities we obtain that

$$\exp\big(\langle Qu, Kz\rangle\big) \leqslant \exp\big(\|Q\|\|K\||u||z|\big) \leqslant \exp\left(\frac{(\|Q\|\|K\||u|)^{\frac{1+\delta}{\delta}}}{(1+\delta)/\delta}\right)\exp\left(\frac{|z|^{1+\delta}}{(1+\delta)}\right). \tag{6.40}$$

We note that

$$\begin{aligned}
\int_{\mathbb{R}^{d_w}} \exp\big(\langle Qu, Kz\rangle\big)\pi(dz) &\leqslant C(u)\int_{\mathbb{R}^{d_w}} \exp\left(\frac{|z|^{1+\delta}}{(1+\delta)}\right)\pi(dz) \\
&= C(u)\int_0^\infty \pi\left(\exp\left(\frac{|z|^{1+\delta}}{(1+\delta)}\right) > t\right)dt \\
&= C(u)\int_0^1 \pi\left(\exp\left(\frac{|z|^{1+\delta}}{(1+\delta)}\right) > t\right)dt + \\
&\quad + C(u)\int_1^\infty \pi\left(\exp\left(\frac{|z|^{1+\delta}}{(1+\delta)}\right) > t\right)dt \\
&\leqslant C(u) + C(u)\int_1^\infty \pi(|z| > \tau)\exp\left(\frac{\tau^{1+\delta}}{(1+\delta)}\right)\tau^\delta d\tau \\
&\leqslant C(u) + C(u)\int_1^\infty \exp\left(\frac{-\delta}{(1+\delta)}\cdot\tau^{1+\delta}\right)\tau^\delta d\tau < \infty,
\end{aligned}$$

where $C(u)$ is a constant depending on u that changes from line to line, and where in the fourth line we used that the first integral is less than or equal to 1 and applied the substitution $t = \exp\big(\tau^{1+\delta}/(1+\delta)\big)$ to the second integral; furthermore, we used $\pi \in \mathcal{P}_\delta(\mathbb{R}^{d_w})$ for the final inequality. □

For implementable algorithms it is useful to consider the finite sample case, where we assume we have access to empirical approximations of the measures $\mu \in \mathfrak{P}(\mathbb{R}^{d_u})$ and $\pi \in \mathfrak{P}_\delta(\mathbb{R}^{d_w})$ via a collection of samples, or ensembles, $\{u(1), \dots, u(N)\} \subset \mathbb{R}^{d_u}$ and $\{w(1), \dots, w(M)\} \subset \mathbb{R}^{d_w}$. The following proposition provides a description of

the attention operator when applied to empirical measures. The resulting expressions will be useful more generally to formulate the trainable algorithms we will use for data assimilation.

Proposition 6.3.8 (Attention on Empirical Measures). *Assume that, for $\mu \in \mathfrak{P}(\mathbb{R}^{d_u}), \pi \in \mathfrak{P}_\delta(\mathbb{R}^{d_w})$, samples $u(1), \cdots, u(N)$ are drawn i.i.d. from μ and samples $w(1), \cdots, w(M) \sim \pi$ are drawn i.i.d. from π. Then for $\mu^N = \frac{1}{N}\sum_{j=1}^N \delta_{u(j)}$ and $\pi^M = \frac{1}{M}\sum_{j=1}^M \delta_{w(j)}$ it holds that*

$$\mathsf{A}(\mu^N, \pi^M) = \frac{1}{N}\sum_{j=1}^N \delta_{\mathfrak{A}\left(u(j);\pi^M\right)},$$

where

$$\mathfrak{A}\left(u(j);\pi^M\right) = \frac{\sum_{k=1}^M \exp\left(\langle Qu(j), Kw(k)\rangle\right) Vw(k)}{\sum_{\ell=1}^M \exp\left(\langle Qu(j), Kw(\ell)\rangle\right)}. \tag{6.42}$$

Proof. This follows by application of identity (6.27) to Definition 6.3.6. □

We note that the expression (6.42) corresponds to the definition of attention on discrete sequences as highlighted in Vaswani et al. (2017).

6.4 Particle Approximation of Learned Mean-field Filters

In this section, we detail our proposed machine-learning-based DA method, the *measure neural mapping enhanced ensemble filter (MNMEF)*, for the state estimation problem. In particular we build on the mean-field formulation introduced in Section 6.2, and in Subsection 6.2.5 in particular, by introducing an interacting particle system approximation. The resulting machine-learned ensemble approach is based on updating $\{v_j^{(n)}\}_{n=1}^N$ to $\{v_{j+1}^{(n)}\}_{n=1}^N$ through the following steps:

$$\text{Predict:} \qquad \widehat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, \tag{6.43a}$$

$$\text{Extend to observation space:} \qquad \widehat{y}_{j+1}^{(n)} = h(\widehat{v}_{j+1}^{(n)}) + \eta_{j+1}^{(n)}, \tag{6.43b}$$

$$\text{Analysis:} \qquad v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + (K_\theta)_{j+1}\left(y_{j+1}^\dagger - \widehat{y}_{j+1}^{(n)}\right). \tag{6.43c}$$

After the prediction steps (6.43a) and (6.43b), we obtain a set of particles $\{(\widehat{v}_{j+1}^{(n)}, h(\widehat{v}_{j+1}^{(n)}))\}_{n=1}^N$. The computation of K_θ is elaborated through equations (6.22), (6.23), and (6.24) in a mean-field perspective, now extended to also include this particle setting. As in Section 6.2, we drop time index j in what follows and recall that $\rho_h = \mathrm{Law}\left((\widehat{v}, h(\widehat{v}))\right)$ is

the joint distribution of $(\widehat{v}, h(\widehat{v}))$ in the mean-field limit. In the particle case, under their implied empirical distribution, we have

$$\rho_h^{(N)} := \frac{1}{N}\sum_{n=1}^{N} \delta_{\left(\widehat{v}_{j+1}^{(n)}, h(\widehat{v}_{j+1}^{(n)})\right)} \tag{6.44}$$

and use this as an input measure to K_θ. The core idea is to train a parameterized neural operator $\mathfrak{F}\left(\widehat{v}, h(\widehat{v}), y^\dagger, \rho_h; \theta\right)$ that outputs the correction terms used to compute K_θ. This can be done using the transformer measure neural mapping (TMNM) architecture F (6.28) introduced in Section 6.3. When applied to empirical measures such as (6.44), this TMNM scheme is equivalent to the set transformer architecture (J. Lee et al., 2019) acting on sequences, as outlined in Remark 6.3.2. In this section, which is concerned with implementation, we will work with the formulation of the scheme on sequences; we will also show that the resulting scheme is invariant to permutation of the elements of the sequence. To distinguish the resulting operator on sequences to the one acting on measures from Section 6.3, we will employ the notation F^{ST} opposed to F; see (6.60) for the definition of F^{ST}. Due to the permutation invariance property of the set transformer, we can treat the input as an unordered set: we define the corresponding operator acting on finite sets using the notation F^{ST} ; see (6.67).

The parameters θ learned in K_θ implicitly depend on N, the number of ensemble members used in training. However, the relation between the set transformer and its mean-field limit from Section 6.3 allows us to deploy the model parameters θ, trained using an ensemble size N, to an ensemble of a different sizes N'. This is because the N and N' systems may both be thought of as approximating the same mean-field limit. Similarly to the standard use of the EnKF, our learning framework also incorporates techniques such as inflation and localization to enhance the performance of the filter (Bach, Baptista, Sanz-Alonso, et al., 2025) as ensemble sizes vary. We view this as a fine-tuning: fixing the majority of the parameters in θ as ensemble sizes vary, but allowing a small subset of parameters—including those related to inflation and localization—to vary with ensemble size.

We present the complete implementation details of our method in the following subsections. In Subsection 6.4.1, we review the set transformer architecture. In Subsection 6.4.2, we present our architecture that learns a parameterized gain matrix, generalizing the EnKF formulation, and enhanced by use of adaptively learned inflation and localization parameters. Subsection 6.4.3 explains the loss function

and training process. In Subsection 6.4.4, we describe how to efficiently fine-tune our method to allow for ensemble size dependent inflation and localization.

Note that all aspects are based on a discrete setting, where our approach evolves an ensemble of particles, as in the EnKF. Consequently, all measure-related operations are transformed into computations based on the empirical measure, or equivalently, the ensemble, viewed as a set of points in state space. When the time-step subscript is omitted, this indicates that the analysis applies to any time step j.

6.4.1 The Set Transformer

In this section, we review the set transformer architecture (J. Lee et al., 2019) used to process ensembles $\{(\widehat{v}^{(n)})\}_{n=1}^N$ or $\{(h(\widehat{v}^{(n)}))\}_{n=1}^N$ into fixed-dimensional feature vectors. While Section 6.3 introduced the set transformer from a measure-theoretic perspective, we now consider our input as a sequence that, due to the architecture's design, can be treated either as a set or an empirical measure.

For two sequences $u \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u})$ and $w \in \mathcal{U}(\llbracket 1, M \rrbracket; \mathbb{R}^{d_w})$, we provide the definition for the sequence-based attention mechanism, which is given by

$$\mathfrak{A}\big(u(j); w\big) = \frac{\sum_{k=1}^M \exp\big(\langle Qu(j), Kw(k)\rangle\big) Vw(k)}{\sum_{\ell=1}^M \exp\big(\langle Qu(j), Kw(\ell)\rangle\big)}. \tag{6.45}$$

We recall that this is equivalent to the formulation (6.42) in Proposition 6.3.8 obtained by applying the measure theoretic definition of attention to empirical measures. Defining the set of sequences of finite length in $\mathbb{R}^d$ by

$$\mathcal{U}_F(\mathbb{R}^d) = \cup_{N=1}^\infty \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^d), \tag{6.46}$$

we can define an attention operator A^{ST} (where the superscript ST indicates it is used in the set transformer architecture) as a sequence-to-sequence operator, analogous to the attention A acting on measures (6.34): [2]

$$\mathsf{A}^{\mathrm{ST}} : \mathcal{U}_F(\mathbb{R}^{d_u}) \times \mathcal{U}_F(\mathbb{R}^{d_w}) \to \mathcal{U}_F(\mathbb{R}^{d_v}). \tag{6.47}$$

In view of (6.45), the output sequence length is the same as the length of the first input. Specifically, for $u \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u})$ and $w \in \mathcal{U}(\llbracket 1, M \rrbracket; \mathbb{R}^{d_w})$,

$$\mathsf{A}^{\mathrm{ST}}(u, w)(j) = \mathfrak{A}\big(u(j); w\big), \quad \forall j \in \llbracket 1, N \rrbracket. \tag{6.48}$$

[2] As in Section 6.3, here d_v does not necessarily refer to the state space dimension in the dynamical system; the exposition in this subsection is quite general, defining the set transformer and linking it to the MNM in Section 6.3.

The learnable parameters in A^{ST} are denoted by

$$\theta(\mathsf{A}^{\mathrm{ST}}) = \{Q, K, V\}. \tag{6.49}$$

In the context of practical implementations, A^{ST} is commonly instantiated as a multihead attention mechanism, which allows the model to jointly attend to information from different representation subspaces (Vaswani et al., 2017).

The set transformer architecture is composed of several key building blocks (J. Lee et al., 2019). The fundamental computation unit is the Cross-Attention Block (CAB) given in the following definition.

Definition 6.4.1 (CAB for Sequences). *The Cross-Attention Block (CAB)* C^{ST} *is an operator that maps two sequences into one sequence:*

$$\mathsf{C}^{ST}: \quad \mathcal{U}_F(\mathbb{R}^{d_u}) \times \mathcal{U}_F(\mathbb{R}^{d_w}) \to \mathcal{U}_F(\mathbb{R}^{d_u}). \tag{6.50}$$

For $u \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u})$ *and* $w \in \mathcal{U}(\llbracket 1, M \rrbracket; \mathbb{R}^{d_w})$, C^{ST} *is defined as*

$$\mathsf{C}^{ST}(u, w) = \mathsf{F}_1^{LN}(u_1 + \mathsf{F}^{NN}(u_1)) \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u}), \tag{6.51a}$$

$$u_1 = \mathsf{F}_2^{LN}\big(u + \mathsf{A}^{ST}(u, w)\big) \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u}). \tag{6.51b}$$

Here A^{ST} *is the sequence-to-sequence attention* (6.47)*;* F^{LN} *denotes application of the layer normalization operator* $\mathfrak{F}^{LN}$ (6.31)*, elementwise in the sequence;* F^{NN} *denotes application of the feedforward operator* $\mathfrak{F}^{NN}$ (6.33)*, also elementwise in the sequence. The* + *operator acting on sequences is interpreted as adding elements with the same index, i.e.,*

$$(u + v)(j) = u(j) + v(j). \tag{6.52}$$

The trainable parameters in C^{ST} *include the parameters from* F_1^{LN}*,* F_2^{LN}*,* F^{NN}*, and* A^{ST}*, i.e.,*

$$\theta(\mathsf{C}^{ST}) = \theta(\mathsf{F}_1^{LN}) \cup \theta(\mathsf{F}_2^{LN}) \cup \theta(\mathsf{F}^{NN}) \cup \theta(\mathsf{A}^{ST}), \tag{6.53}$$

where the layer normalization layers $\mathsf{F}_1^{LN}, \mathsf{F}_2^{LN}$ *have trainable parameters according to* (6.31)(6.32)*, the feedforward layer* F^{NN} *has trainable parameters according to* (6.33)*, and the attention layer* A^{ST} *has trainable parameters according to* (6.49)*. We note that practically the cross-attention block on sequences is the multi-head attention block from Vaswani et al. (2017) and J. Lee et al. (2019) and may be viewed as the sequence-to-sequence analog of the cross-attention block* C *on measures from* (6.29b)*.*

The set transformer architecture is composed of several self-attention blocks (SAB) (Definition 6.4.2) and a pooling by multi-head attention block (PMA) (Definition 6.4.3), which are defined in the following using CAB.

Definition 6.4.2 (SAB). *The self-attention block (SAB)* S^{ST}*, also called the set attention block (J. Lee et al., 2019), maps a sequence to another sequence with the same length and dimension, so that*

$$\mathsf{S}^{ST}: \quad \mathcal{U}_F(\mathbb{R}^{d_u}) \to \mathcal{U}_F(\mathbb{R}^{d_u}). \tag{6.54}$$

The application of S^{ST} *is calculated by setting both inputs of* C^{ST} *to be the same, i.e.,*

$$\mathsf{S}^{ST}(u) = \mathsf{C}^{ST}(u, u). \tag{6.55}$$

Here, the trainable parameters satisfy

$$\theta(\mathsf{S}^{ST}) = \theta(\mathsf{C}^{ST}). \tag{6.56}$$

See (6.29a) *for the measure theoretic analog* Σ*.*

Definition 6.4.3 (PMA). *The pooling by multi-head attention block (PMA)* $\mathsf{C}^{ST}(s, \bullet)$ *is defined by replacing the first input in the CAB* $\mathsf{C}^{ST}(\bullet, \bullet)$ *by a trainable parameter* s*, called the seed. The seed* $s \in \mathcal{U}(\llbracket 1, N_s \rrbracket, \mathbb{R}^{d_s})$ *is a sequence, where* N_s *and* d_s *are fixed hyperparameters. Therefore* $\mathsf{C}^{ST}(s, \bullet)$ *maps a sequence of any finite length to a sequence with fixed length,*

$$\mathsf{C}^{ST}(s, \bullet): \quad \mathcal{U}_F(\mathbb{R}^{d_u}) \to \mathcal{U}(\llbracket 1, N_s \rrbracket; \mathbb{R}^{d_s}). \tag{6.57}$$

The trainable parameters in PMA are the parameters in the CAB and the seed s*,*

$$\theta\big(\mathsf{C}^{ST}(s, \bullet)\big) = \theta(\mathsf{C}^{ST}) \cup \{s\}. \tag{6.58}$$

See Remark 6.3.2 for the measure theoretic analog $\mathsf{C}(s, \bullet)$*.*

The set transformer maps a sequence of any finite length to a fixed-dimensional feature vector:

$$\mathsf{F}^{\mathrm{ST}}: \mathcal{U}_F(\mathbb{R}^{d_u}) \to \mathbb{R}^{d_{\mathrm{ST}}}. \tag{6.59}$$

To fully specify the detailed architecture of F^{ST}, we will utilize two multilayer perceptrons (6.33) $\mathsf{F}^{\mathrm{NN}}_{\mathrm{in}}: \mathbb{R}^{d_u} \to \mathbb{R}^{d_{\mathrm{latent}}}$, and $\mathsf{F}^{\mathrm{NN}}_{\mathrm{out}}: \mathbb{R}^{N_s \times d_s} \to \mathbb{R}^{d_{\mathrm{ST}}}$ (that act elementwise on sequences), and a parameter-free concatenation layer $\mathsf{F}^{\mathrm{Cat}}: \mathcal{U}(\llbracket 1, N_s \rrbracket, \mathbb{R}^{d_s}) \to \mathbb{R}^{N_s \times d_s}$ that concatenates all elements in a sequence into a vector.

For a sequence $u \in \mathcal{U}(\llbracket 1, N \rrbracket; \mathbb{R}^{d_u})$, the set transformer is hence given by

$$\mathsf{F}^{\mathrm{ST}}(u) = \mathsf{F}^{\mathrm{Dec}}(\bullet) \circ \mathsf{C}^{\mathrm{ST}}(s, \bullet) \circ \mathsf{F}^{\mathrm{Enc}}(u), \tag{6.60a}$$

$$\mathsf{F}^{\mathrm{Enc}}(u) = \big(\mathsf{S}^{\mathrm{ST}}_{N_{\mathrm{e}}}(\bullet) \circ \cdots \circ \mathsf{S}^{\mathrm{ST}}_{1}(\bullet)\big) \circ \mathsf{F}^{\mathrm{NN}}_{\mathrm{in}}(u), \tag{6.60b}$$

$$\mathsf{F}^{\mathrm{Dec}}(u) = \mathsf{F}^{\mathrm{NN}}_{\mathrm{out}}(\bullet) \circ \mathsf{F}^{\mathrm{Cat}}(\bullet) \circ \big(\mathsf{S}^{\mathrm{ST}}_{N_{\mathrm{d}}+N_{\mathrm{e}}}(\bullet) \circ \cdots \circ \mathsf{S}^{\mathrm{ST}}_{1+N_{\mathrm{e}}}(\bullet)\big)(u). \tag{6.60c}$$

The subscripts $\ell \in [N_{\mathrm{e}} + N_{\mathrm{d}}]$ emphasize that the trainable parameters of these SABs are different. Integer N_{e} is the number of SABs, $\mathsf{S}^{\mathrm{ST}}_{\ell}$, $\ell = 1, 2, \ldots, N_{\mathrm{e}}$, in the encoder $\mathsf{F}^{\mathrm{Enc}}$, while N_{d} is the number of SABs, $\mathsf{S}^{\mathrm{ST}}_{\ell}$, $\ell = 1 + N_{\mathrm{e}}, 2 + N_{\mathrm{e}}, \ldots, N_{\mathrm{d}} + N_{\mathrm{e}}$, in the decoder $\mathsf{F}^{\mathrm{Dec}}$. Specifically, we have

$$\mathsf{S}^{\mathrm{ST}}_{\ell} : \mathcal{U}_F(\mathbb{R}^{d_{\mathrm{latent}}}) \to \mathcal{U}_F(\mathbb{R}^{d_{\mathrm{latent}}}), \qquad \ell = 1, 2, \ldots, N_{\mathrm{e}}, \tag{6.61a}$$

$$\mathsf{S}^{\mathrm{ST}}_{\ell} : \mathcal{U}_F(\mathbb{R}^{d_s}) \to \mathcal{U}_F(\mathbb{R}^{d_s}), \qquad \ell = 1 + N_{\mathrm{e}}, 2 + N_{\mathrm{e}}, \ldots, N_{\mathrm{d}} + N_{\mathrm{e}}. \tag{6.61b}$$

The trainable parameters in F^{ST} are given by

$$\begin{aligned} \theta(\mathsf{F}^{\mathrm{ST}}) &= \theta\big(\mathsf{F}^{\mathrm{Dec}}\big) \cup \theta\big(\mathsf{C}^{\mathrm{ST}}(s, \bullet)\big) \cup \theta\big(\mathsf{F}^{\mathrm{Enc}}\big) \\ &= \theta\big(\mathsf{F}^{\mathrm{NN}}_{\mathrm{in}}\big) \cup \theta\big(\mathsf{F}^{\mathrm{NN}}_{\mathrm{out}}\big) \cup \big(\cup_{\ell=1}^{N_{\mathrm{e}}+N_{\mathrm{d}}} \theta\big(\mathsf{S}^{\mathrm{ST}}_{\ell}\big)\big) \cup \theta\big(\mathsf{C}^{\mathrm{ST}}(s, \bullet)\big). \end{aligned} \tag{6.62}$$

The set transformer architecture exhibits two key properties when processing an input sequence, namely, the permutation invariance (Proposition 6.4.4) and adaptation to variable input lengths (Proposition 6.4.5). These two propositions are proved in J. Lee et al. (2019) under a slightly different setting: in the original work a length-N sequence in $\mathbb{R}^d$ is represented as an $N \times d$ matrix, whereas here it is a mapping $u \in \mathcal{U}(\llbracket 1, N \rrbracket, \mathbb{R}^d)$.

Proposition 6.4.4 (Permutation Invariance). *A permutation σ is a bijective function from $\llbracket 1, N \rrbracket$ to $\llbracket 1, N \rrbracket$. We extend the definition of σ to sequences in $u \in \mathcal{U}(\llbracket 1, N \rrbracket, \mathbb{R}^{d_u})$ so that*

$$\sigma(u)(n) = u(\sigma(n)), \quad n \in \llbracket 1, N \rrbracket. \tag{6.63}$$

Then for any permutation $\sigma : \llbracket 1, N \rrbracket \mapsto \llbracket 1, N \rrbracket$ and any input sequence $u \in \mathcal{U}(\llbracket 1, N \rrbracket, \mathbb{R}^{d_u})$, we have

$$\mathsf{F}^{ST}(u) = \mathsf{F}^{ST}(\sigma(u)). \tag{6.64}$$

Proposition 6.4.5 (Adaptation to Variable Input Length). *A set transformer architecture with fixed parameters takes input sequences with different lengths while the output dimension is fixed:*

$$\mathsf{F}^{ST}(u) \in \mathbb{R}^{d_{ST}}, \quad \forall u \in \mathcal{U}(\llbracket 1, N \rrbracket, \mathbb{R}^{d_u}), \quad \forall N \in \mathbb{N}. \tag{6.65}$$

Based on these two properties, we can bridge the sequence input and the probability measure input for the set transformer architecture. The permutation invariance property (Proposition 6.4.4) enables us to abstract away from the sequential nature of the input and treat it as an unordered set. More fundamentally, this allows us to interpret the input as an empirical distribution, where each element in the sequence represents a sample from some underlying distribution. Furthermore, Proposition 6.4.5 demonstrates that we can accommodate varying input lengths without architectural modifications, effectively allowing us to adjust the number of samples in the empirical distribution. Together, these properties (Propositions 6.4.4 and 6.4.5) provide theoretical foundation for analysis of the main components of the set transformer in the context of probability measures (Section 6.3).

Remark 6.4.6 (Common Notation for Operations on Measures and on Sequences). *In this section, much of the notation coincides with that in Section 6.3. This is because the underlying definitions are almost identical, with the key distinction being that in Section 6.3 the operators act on probability measures, whereas in this section they act on sequences. The preceding developments in this subsection explain the connection. For example, in Section 6.3,* Σ *and* C *denote the measure-based self-attention* (6.29a) *and cross-attention* (6.29b) *blocks, while in this section,* S^{ST} *and* C^{ST} *represent the sequence-based self-attention* (6.55) *(Definition 6.4.2) and cross-attention* (6.51) *(Definition 6.4.1) blocks, respectively. In addition,* F *refers to the transformer measure neural mapping* (6.28) *in Section 6.3 while* F^{ST} *refers to the set transformer* (6.60) *acting on sequences.*

6.4.2 Our Architecture

In this section we provide a detailed description the architecture in our proposed learnable DA method, MNMEF. The core of our approach is to learn a parameterized gain matrix (Subsection 6.4.2.1), generalizing how the classical EnKF utilizes a Gaussian-inspired gain. Additionally, to enhance performance, we incorporate inflation and localization techniques (Subsections 6.4.2.2 and 6.4.2.3), akin to their role in EnKF.

In our architecture, we use a set transformer (Subsection 6.4.1) to process the ensemble $\{(\widehat{v}^{(n)}, h(\widehat{v}^{(n)}))\}_{n=1}^{N}$ into a fixed-dimensional feature vector. We define the set of finite subsets in $\mathbb{R}^d$ by

$$\mathcal{F}(\mathbb{R}^d) = \{A \subseteq \mathbb{R}^d \mid A \text{ contains a finite set of elements}\}. \tag{6.66}$$

Because of the permutation invariance property (Proposition 6.4.4), we may reinterpret the set transformer F^{ST} as a map F^{ST} that acts on sets rather than sequences: $F^{\mathrm{ST}} : \mathcal{F}(\mathbb{R}^{d_v} \times \mathbb{R}^{d_y}) \to \mathbb{R}^{d_{\mathrm{ST}}}$, given by

$$F^{\mathrm{ST}}\left(\{(\widehat{v}^{(n)}, h(\widehat{v}^{(n)}))\}_{n=1}^{N}; \theta_{\mathrm{ST}}\right) = f_v \in \mathbb{R}^{d_{\mathrm{ST}}}. \tag{6.67}$$

The feature vector f_v can be viewed as a representation of the empirical distribution $\rho_h^{(N)}$ given in (6.44) and similarly as an approximate summary of ρ_h. This representation is subsequently utilized to learn the gain matrix, inflation, and the localization.

Remark 6.4.7 (Choice of d_{ST}). *By Proposition 6.4.5, F^{ST} outputs a fixed dimension d_{ST} regardless of ensemble size N. Therefore, we do not need to adjust d_{ST} based on N. Larger d_{ST} values offer more expressiveness, but increase computational cost. Our tests with the Lorenz '96 model (Subsection 6.5.2) using $d_{ST} = 32, 64, 128$ showed only marginal improvements with increasing d_{ST}, thus we selected $d_{ST} = 64$ for all experiments.*

6.4.2.1 Learning the Gain Matrix

From the analysis step (6.43c) and the proposed form (6.22), we have the following form for the learnable analysis step:

$$v^{(n)} = \widehat{v}^{(n)} + K_\theta\left(y^\dagger - \widehat{y}^{(n)}\right), \tag{6.68a}$$

$$K_\theta = K_\theta^{(1)}\left(K_\theta^{(2)} + \Gamma\right)^{-1}. \tag{6.68b}$$

Recall that, in the mean-field limit, we impose an inductive bias on our architecture by defining $K_\theta^{(1)}$ and $K_\theta^{(2)}$ in the form (6.23), allowing for a learned correction through (6.24). Recall that $\widehat{m}$ and $\widehat{h}$ are the mean of $\{\widehat{v}^{(n)}\}_{n=1}^N$ and $\{h(\widehat{v}^{(n)})\}_{n=1}^N$, respectively. The empirical version of (6.23), used in the practical implementation, takes the form

$$K_\theta^{(1)} = \frac{1}{N}\sum_{n=1}^{N}\left(\widehat{v}^{(n)} - \widehat{m} + \widehat{w}_\theta^{(n)}\right) \otimes \left(h(\widehat{v}^{(n)}) - \widehat{h} + \widehat{z}_\theta^{(n)}\right), \tag{6.69a}$$

$$K_\theta^{(2)} = \frac{1}{N}\sum_{n=1}^{N}\left(h(\widehat{v}^{(n)}) - \widehat{h} + \widehat{z}_\theta^{(n)}\right) \otimes \left(h(\widehat{v}^{(n)}) - \widehat{h} + \widehat{z}_\theta^{(n)}\right), \tag{6.69b}$$

where $\widehat{w}_\theta^{(n)}$ and $\widehat{z}_\theta^{(n)}$ are trainable correction terms that depend on the state $\widehat{v}^{(n)}$, the observation $h(\widehat{v}^{(n)})$, the true observation $y^\dagger$, and the ensemble $\{(\widehat{v}^{(n)}, h(\widehat{v}^{(n)}))\}_{n=1}^N$.

The correction terms are defined by a neural operator $\mathfrak{F}$ as in (6.24) for the mean-field perspective presented in Section 6.2. In our practical implementation, we replace the dependence on ρ_h in $\mathfrak{F}$ by the ensemble $\{(\widehat{v}^{(\ell)}, h(\widehat{v}^{(\ell)}))\}_{\ell=1}^N$ and use the set transformer F^{ST} (6.67) to process the ensemble into a feature vector f_v. Recall that this can be viewed as using the empirical approximation $\rho_h^{(N)} \approx \rho_h$ given by (6.44).

To learn the correction terms for each particle, we then train a neural network $F^{\mathrm{gain}} : \mathbb{R}^{d_v} \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_{\mathrm{ST}}} \to \mathbb{R}^{d_v} \times \mathbb{R}^{d_y}$ with the standard multilayer perceptron (MLP) architecture that takes the feature vector for the ensemble as an input:

$$F^{\mathrm{gain}}\left(\widehat{v}^{(n)}, h(\widehat{v}^{(n)}), y^\dagger, f_v; \theta_{\mathrm{gain}}\right) = \left(\widehat{w}_\theta^{(n)}, \widehat{z}_\theta^{(n)}\right), \tag{6.70}$$

where θ_{gain} denotes trainable parameters.

From (6.69a) and (6.69b), it is evident that if the correction terms satisfy $\widehat{w}_\theta^{(n)} = \widehat{z}_\theta^{(n)} = 0$, the formulation reduces to the EnKF (6.16). When the dynamics and observation models are linear, or in other words, when the filtering distribution of the states v is Gaussian, the Kalman filter achieves optimality (Kalman, 1960; Evensen, 2003; Calvello, Reich, and Andrew M. Stuart, 2025). Therefore, the motivation for our proposed scheme (6.22) lies in its optimality in the linear (or Gaussian) setting, while further aiming to improve the performance of the Kalman filter in nonlinear problems by learning the correction terms.

Remark 6.4.8 (On the Invertibility in the Gain Computation). *The matrix $K_\theta^{(2)}$, defined in* (6.69b) *as an average of outer products, is positive semi-definite. Since Γ is a fixed positive definite matrix, the sum $K_\theta^{(2)} + \Gamma$ is also positive definite and thus invertible. Crucially, the minimum eigenvalue of $K_\theta^{(2)} + \Gamma$ is bounded from below by the minimum eigenvalue of Γ, which is a positive constant independent of any trainable parameters. This property ensures the numerical stability of computing $(K_\theta^{(2)} + \Gamma)^{-1}$ during backpropagation.*

Remark 6.4.9 (Correction Approach Versus End-to-end Gain Learning). *In addition to the proposed approach with correction terms, we have implemented the end-to-end learning of the operator $\mathfrak{B}$ presented in Subsection 6.2.4 and learning the Kalman gain K_θ directly without any constraint on its form. We find that the end-to-end approach or directly learning K_θ results in unstable and slower learning. A poorly initialized or an unconstrained K_θ leads to substantial divergence of the resulting filtered trajectory away from the ground truth trajectory. The EnKF-like*

structure in (6.68b) *with learnable correction terms provides better stability while maintaining adaptability to nonlinear dynamics.*

6.4.2.2 Learning the Inflation

Inflation is a technique used to increase the ensemble spread, mitigating the underestimation of analysis covariance caused by sampling errors in ensemble-based methods. Finite ensemble sizes often lead to sampling errors in the forecast covariance, causing the filter to overly trust forecasts and risk filter divergence. Inflation helps maintain the response of the filter to observations by counteracting these effects.

The most common form of inflation is multiplicative inflation, where the analysis ensemble $\{v^{(n)}\}_{n=1}^{N}$ that is calculated after the analysis step (6.43c) is modified as follows,

$$v^{(n)} \leftarrow v^{(n)} + (\alpha - 1)\left(v^{(n)} - \frac{1}{N}\sum_{\ell=1}^{N} v^{(\ell)}\right), \tag{6.71}$$

where $\alpha \geqslant 1$ is the inflation parameter. In our learning framework, we replace the term proportional to $(\alpha - 1)$ by a learned correction term $\widehat{u}_{\theta}^{(n)}$,

that is output by an MLP-based neural network $F^{\text{infl}} : \mathbb{R}^{d_v} \times \mathbb{R}^{d_{\text{ST}}} \to \mathbb{R}^{d_v}$ given by

$$F^{\text{infl}}\left(v^{(n)}, f_v; \theta_{\text{infl}}\right) = \widehat{u}_{\theta}^{(n)}, \tag{6.72}$$

where $v^{(n)}$ is the estimated state given by (6.43c), and θ_{infl} denotes trainable parameters. The learned inflation step is processed after the analysis step (6.43c), resulting in the particle update:

$$v^{(n)} \leftarrow v^{(n)} + \widehat{u}_{\theta}^{(n)}. \tag{6.73}$$

Remark 6.4.10 (Dependence and Implications of the Learned Inflation Term). *The common form of inflation is a function of the ensemble states, i.e., not the observations, in* (6.71)*. Similarly, the learned inflation term* $\widehat{u}_{\theta}^{(n)}$ *in* (6.72) *depends on the ensemble states* $\widehat{v}^{(n)}$ *and the feature vector* f_v*, without information of the true observation* $y^{\dagger}$ *or the observation noise covariance* Γ*. We avoid the risk that the inflation term would evolve into an end-to-end correction term, ensuring the learning performed by other components in our framework remains meaningful and effective.*

6.4.2.3 Learning the Localization

In spatially extended systems, sampling errors can lead to inaccurate covariance estimation, particularly when the sample size is small. True correlations typically decay with distance, but spurious long-range correlations may arise, distorting the covariance structure. Localization addresses this issue by suppressing artificial correlations while preserving meaningful local correlations.

Localization is implemented by damping covariances in an empirical covariance matrix based on distance. This is typically achieved by applying a Hadamard product (elementwise product, denoted by $\circ$) between the empirical covariance matrix and a predefined localization weight matrix L. For the dynamic model (6.1) with the state in d_v-dimensional space (i.e., $v \in \mathbb{R}^{d_v}$), $L \in \mathbb{R}^{d_v \times d_v}$ is calculated by

$$(L)_{k\ell} = g(D_{k\ell}), \quad \forall k, \ell \in [d_v], \tag{6.74}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a function that maps distance values to localization weights (e.g., the Gaspari–Cohn function (Gaspari and Cohn, 1999)), and $D_{k\ell}$ is the distance between index k and ℓ. To incorporate the localization technique in the EnKF, the Kalman gain formula (6.11b) is modified as

$$K = \left(\widehat{C}^{vh} \circ L^{vh}\right)\left(\widehat{C}^{hh} \circ L^{hh} + \Gamma\right)^{-1}, \tag{6.75}$$

where $L^{vh} \in \mathbb{R}^{d_v \times d_y}$ and $L^{hh} \in \mathbb{R}^{d_v \times d_y}$ are localization weight matrices corresponding to $\widehat{C}^{vh}$ and $\widehat{C}^{hh}$, respectively.

In our learning-based formulation of localization we proceed as follows: we incorporate localization into our parameterized Kalman gain by modifying (6.68b) to take the form

$$K_\theta = \left(K_\theta^{(1)} \circ L_\theta^{(1)}\right)\left(K_\theta^{(2)} \circ L_\theta^{(2)} + \Gamma\right)^{-1}, \tag{6.76}$$

The localization weight matrices $L_\theta^{(1)}$ and $L_\theta^{(2)}$ follow the structure in (6.74), where g is replaced by a parameterized distance-to-weight function $g_\theta : \mathbb{R} \to \mathbb{R}$.

In practice, it is not necessary to learn the entire function g_θ since we only have a finite number of different distances, defined by the spatial grid for discretized PDEs, and time indices in the setting of ODEs. We denote the unique distance values in $\{D_{k\ell}\}_{k,\ell=1}^{d_v}$ as $\{D_1, D_2, \ldots, D_{N_D}\}$, where N_D is the number of distinct distances. Given a vector $\widehat{g} = [\widehat{g}_1, \widehat{g}_2, \ldots, \widehat{g}_{N_D}] \in \mathbb{R}^{N_D}$, we can define the function g_θ by

$$g_\theta(D_i) = \widehat{g}_i, \quad \forall i \in [N_D]. \tag{6.77}$$

We employ a MLP-based neural network $F^{\text{loc}} : \mathbb{R}^{d_{\text{ST}}} \to \mathbb{R}^{N_D}$ given by

$$F^{\text{loc}}\left(f_v; \theta_{\text{loc}}\right) = \widehat{g}. \tag{6.78}$$

The parameterized localization weight matrices $L_\theta^{(1)}$ and $L_\theta^{(2)}$ are calculated based on the output of this learned function g_θ.

Remark 6.4.11 (Implementation Insights for Learning Localization Weights). *Here we provide more details for the learning process of the localization weight matrices and make some comments on our proposed approach.*

- *The final layer of* F^{loc} *is a* softmax *function scaled by 2, ensuring all outputs lie within the interval* $[0, 2]$. *Although amplifying underestimated covariances is typically the responsibility of inflation alone, our architecture jointly performs localization and inflation. Thus, we allow the weights to range in* $[0, 2]$ *rather than* $[0, 1]$ *to grant additional flexibility in the localization component.*

- *Instead of learning a continuous parametric form for* g_θ*, we learn its values only at discrete distances* $\{D_i\}_{i=1}^{N_D}$. *This reduces model complexity while preserving localization expressivity.*

- *While EnKF-based localization methods typically use fixed weight matrices across time steps (e.g., using the Gaspari–Cohn function (Gaspari and Cohn, 1999)), there is much interest in adaptive approaches (Vishny et al., 2024). Our learning-based approach adapts these matrices for each time step* j *using neural networks via its dependence on the ensemble. As shown in* (6.78) *(time index* j *omitted for simplicity), the network processes time-step-specific inputs, allowing the localization scheme to dynamically adjust based on the current system state.*

Remark 6.4.12 (On sharing the distance-to-weight function across $L_\theta^{(1)}$ and $L_\theta^{(2)}$). *In forthcoming experiments in this chapter, the observation operator* h *subsamples state components, so the distances between observation indices are induced by the same state-space metric used for state indices. Accordingly, we learn a single distance-to-weight function* g_θ *and use it consistently to build* $L_\theta^{(1)}$ *and* $L_\theta^{(2)}$. *For more general* h *(e.g., nonlinear or indirect observations), one may instead learn two distinct functions,* $g_\theta^{(1)}$ *and* $g_\theta^{(2)}$*, tailored to state–observation pairs for* $L_\theta^{(1)}$ *and observation–observation pairs for* $L_\theta^{(2)}$*, respectively.*

6.4.3 Loss Function

This section outlines the procedure for training the neural network parameters, including the gain, inflation and localization (Subsection 6.4.2). Based on the stochastic dynamic model (6.1) and the data observation model (6.2), we make the follow assumption for the training:

Data Assumption 1. *Our training is conducted for fixed dynamics Ψ, observation operator h, and noise covariance matrices Σ and Γ. The time step Δt is also fixed (and incorporated in Ψ). The initial condition is a Gaussian with a fixed covariance C_0, i.e. $v_0 \sim \mathsf{N}(\bullet, C_0)$. For some $M, J \in \mathbb{N}$, the training data consists of M trajectories of length $J+1$, extracted from a single long trajectory generated by propagating the dynamic model* (6.1) *for $M(J+1)$ steps. Each sub-trajectory $\{v_j^\dagger\}_{j=0}^J$ has corresponding observations $\{y_j^\dagger\}_{j=1}^J$ generated according to* (6.2)*.*

For each trajectory, we generate estimated states from our ensemble-based data assimilation model to compare with the true states from the dynamic model, enabling us to train the parameters θ_{ST}, θ_{gain}, θ_{infl}, and θ_{loc}, for the set transformer F^{ST} (6.67), and MLPs F^{gain} (6.70), F^{infl} (6.72) and F^{loc} (6.78) respectively.

We initialize an ensemble $\{v_0^{(n)}\}_{n=1}^N$ at the initial time step $j = 0$ for each sub-trajectory, with members sampled i.i.d. from a Gaussian distribution with the mean $v_0^{(n)}$ and the covariance C_0:

$$v_0^{(n)} \sim \mathsf{N}(v_0^\dagger, C_0), \quad n = 1, \dots, N. \tag{6.79}$$

For $j = 0, 1, \dots, J-1$ and trainable parameters $\theta = \{\theta_{\text{ST}}, \theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}\}$, we update the ensemble $\{v_j^{(n)}(\theta)\}_{n=1}^N$ to $\{v_{j+1}^{(n)}(\theta)\}_{n=1}^N$ according to

$$\widehat{v}_{j+1}^{(n)}(\theta) = \Psi(v_j^{(n)}(\theta)) + \xi_j^{(n)}, \quad \xi_j^{(n)} \sim \mathsf{N}(0, \Sigma), \tag{6.80a}$$

$$\widehat{y}_{j+1}^{(n)}(\theta) = h(\widehat{v}_{j+1}^{(n)}(\theta)) + \eta_{j+1}^{(n)}, \quad \eta_{j+1}^{(n)} \sim \mathsf{N}(0, \Gamma), \tag{6.80b}$$

$$v_{j+1}^{(n)}(\theta) = (\widehat{v}_{j+1}^{(n)}(\theta) + \widehat{u}_\theta^{(n)}) + (K_\theta)_{j+1}\left(y_{j+1}^\dagger - \widehat{y}_{j+1}^{(n)}(\theta)\right), \tag{6.80c}$$

where $(K_\theta)_{j+1}$ is our learned parameterized gain matrix, including localization, and $\widehat{u}_\theta^{(n)}$ is the inflation term according to Subsection 6.4.2. For all parameters θ we choose $v_0^{(n)}(\theta) = v_0^{(n)}$. The ensembles $\{v_j^{(n)}(\theta)\}_{n=1}^N$ depend on the trainable parameters θ for $j = 1, 2, \dots, J$.

In this chapter we focus on state estimation, given by the mean of the estimated ensemble $\{v_j^{(n)}(\theta)\}_{n=1}^N$, i.e.,

$$\overline{v}_j(\theta) = \frac{1}{N}\sum_{n=1}^{N} v_j^{(n)}(\theta). \tag{6.81}$$

To train our neural network parameters effectively, we introduce a loss function that quantifies the discrepancy between the estimated states and the true states across the m-th sub-trajectory $V_m = \{v_j^\dagger\}_{j=1}^J$, given by

$$\mathcal{L}_m(\theta) = \frac{1}{J}\sum_{j=1}^{J} \frac{\|\overline{v}_j(\theta) - v_j^\dagger\|_2^2}{\|v_j^\dagger\|_2^2}, \tag{6.82}$$

which measures the relative accuracy of our ensemble mean predictions compared to the true trajectory. Our loss (6.82) is a relative squared L_2 loss, which is preferred over the standard L_2 loss for two main reasons. First, using the squared L_2 norm avoids computing square roots, which can lead to numerical instabilities when calculating gradients, especially when the error approaches zero. Second, normalizing by the true state magnitude ($\|v_j^\dagger\|_2^2$) makes the loss scale-invariant across different state variables and trajectory segments, ensuring balanced parameter updates regardless of the absolute magnitudes of the state components. This relative formulation is particularly important in dynamical systems where state variables may span different orders of magnitude.

In the training process, the loss is averaged across M training trajectories indexed by $m \in [\![1, M]\!]$,

$$\mathcal{L}_{[\![1,M]\!]}(\theta) = \frac{1}{M}\sum_{m\in[\![1,M]\!]} \mathcal{L}_m(\theta). \tag{6.83}$$

The trainable parameter θ is optimized based on the above-defined loss $\mathcal{L}_{[\![1,M]\!]}$. In practice, the loss $\mathcal{L}_{[\![1,M]\!]}$ is approximated by a mini-batch as a subset of $[\![1, M]\!]$.

Remark 6.4.13 (Gradient Truncation in Sequential Learning). *Computing the loss* (6.82) *requires evaluating ensemble estimates* $\{\{v_j^{(n)}(\theta)\}_{n=1}^N\}_{j=1}^J$ *across the entire trajectory. For the ensemble at step* j*, this involves nested evaluations of* K_θ *up to* j *times. These nested evaluation arise from the sequential nature of the trajectory, where each step depends on the computing of the Kalman gain from the previous steps, requiring backpropagation through the entire computational history. With large trajectory lengths* J*, this causes slow training and numerical instability since it is nearly equivalent to training a very deep neural network.*

To address this issue, we introduce a gradient-detach hyperparameter $J_0 \in \mathbb{N}$. This hyperparameter limits the gradient computation for the loss related to time step j to only the steps $j, j-1, \ldots, j-J_0+1$. This approach assumes that the trajectory at time step j is not significantly affected by parameter changes in the Kalman gain for earlier time steps, which is reasonable given the diminishing influence of distant past states in many dynamical systems. This technique significantly improves training speed and stability.

Remark 6.4.14 (Training Targets and Probabilistic Losses). *In the loss* (6.82)*, the ground-truth states used for supervised training are obtained by simulating the known dynamical model (forward integration of Ψ with prescribed noise model). Thus, the availability of training data hinges on access to an explicit and tractable dynamic model. In high-dimensional settings where such a model is unavailable or impractical, a surrogate simulator (e.g., a reduced-order or data-driven one) can be used to synthesize training trajectories and observations. Furthermore, while the proposed loss* (6.82) *focuses on state estimation, a promising direction for future work is to adopt probabilistic loss functions that target the full filtering distribution $\mathbb{P}(v_j \mid Y_j)$ rather than only the mean-based point estimator; see Subsection 6.1.1.*

6.4.4 Fine-Tuning Inflation & Localization

Following the developments in Subsection 6.4.2, we may apply a set transformer (6.67) to process ensembles of any size into fixed-dimensional feature vectors. This allows for pretraining our MNMEF with ensemble size N and perform inference with $N' \neq N$. However in classical EnKF formulations, hyperparameters such as inflation and localization depend on the ensemble size, and allowing for this dependence significantly enhances performance. Thus, we consider fine-tuning a subset of the parameters for inference at ensemble size $N' \neq N$. We now detail this methodology.

The architecture defined in Subsection 6.4.2 is based on trainable parameters θ_{ST}, for the set transformer F^{ST} (6.67), and θ_{gain}, θ_{infl}, and θ_{loc} for the MLPs F^{gain} (6.70), F^{infl} (6.72) and F^{loc} (6.78) respectively. For efficient fine-tuning when transitioning from ensemble size N to N', we freeze θ_{ST} and only fine-tune θ_{gain}, θ_{infl}, and θ_{loc}. This specific choice of fine-tuning is effective because: (1) freezing θ_{ST} significantly reduces computational costs, as the set transformer contains the majority of network parameters; (2) the remaining parameters, though small in number, measurably improve performance when they are adapted to the ensemble size.

The training loss for the m-th sub-trajectory $V_m = \{v_j^\dagger\}_{j=1}^J$ is defined by (6.81), (6.82):

$$\mathcal{L}_m^{(N)}(\theta_{\text{ST}}, \theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}) = \frac{1}{J}\sum_{j=1}^{J} \frac{\|\overline{v}_j(\theta) - v_j^\dagger\|_2^2}{\|v_j^\dagger\|_2^2}, \tag{6.84}$$

where we add the superscript N to emphasize dependence on the ensemble size. In pretraining with size N we solve

$$\theta_{\text{ST}}^{(N)}, \theta_{\text{gain}}^{(N)}, \theta_{\text{infl}}^{(N)}, \theta_{\text{loc}}^{(N)} = \underset{\theta_{\text{ST}}, \theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}}{\arg\min} \; \mathcal{L}_{[\![1,M]\!]}^{(N)}(\theta_{\text{ST}}, \theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}). \tag{6.85}$$

During fine-tuning for size $N' \neq N$, using the same training trajectories we solve

$$\theta_{\text{gain}}^{(N')}, \theta_{\text{infl}}^{(N')}, \theta_{\text{loc}}^{(N')} = \underset{\theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}}{\arg\min} \; \mathcal{L}_{[\![1,M]\!]}^{(N')}(\theta_{\text{ST}}^{(N)}, \theta_{\text{gain}}, \theta_{\text{infl}}, \theta_{\text{loc}}) \tag{6.86}$$

with fixed $\theta_{\text{ST}}^{(N)}$. In practice, we use small mini-batches of $[\![1, M]\!]$ to estimate the losses in (6.85) and (6.86).

With the updated parameters, we proceed with ensemble size N' using:

$$\text{Feature Vector:} \qquad f_v^{(N')} = F^{\text{ST}}\left(\{(\widehat{v}^{(n)}, h(\widehat{v}^{(n)}))\}_{n=1}^{N'}; \theta_{\text{ST}}^{(N)}\right), \tag{6.87a}$$

$$\text{Gain:} \qquad F^{\text{gain}}\left(\bullet, f_v^{(N')}; \theta_{\text{gain}}^{(N')}\right), \tag{6.87b}$$

$$\text{Inflation:} \qquad F^{\text{infl}}\left(\bullet, f_v^{(N')}; \theta_{\text{infl}}^{(N')}\right), \tag{6.87c}$$

$$\text{Localization:} \qquad F^{\text{loc}}\left(f_v^{(N')}; \theta_{\text{loc}}^{(N')}\right). \tag{6.87d}$$

Although $\theta_{\text{ST}}^{(N)}$ remains unchanged after the fine-tuning, the feature vector $f_v^{(N')}$ differs from the pretraining features f_v since the input ensemble size changes from N to N'.

Remark 6.4.15 (Efficient Training Strategy). *While the per-particle update in the analysis step has linear complexity with respect to the ensemble size N for fixed gain matrix, the overall cost of one forward pass is dominated by the attention operations used to compute correction terms. These attention blocks scale quadratically in N. In practice, we adopt an efficient training strategy by pretraining on smaller ensemble sizes N and applying fine-tuning for larger ensembles $N' > N$, which maintains comparable performance while substantially reducing computational cost. Further improvements are possible by employing localized or linear-attention variants that can reduce the scaling in N.*

In Subsection 6.5.8, we demonstrate this fine-tuning method's effectiveness, achieving comparable performance to full parameter fine-tuning with significantly improved efficiency.

6.5 Numerical Experiments

In this section we present numerical experiments to demonstrate the improved numerical benefit of our proposed method, MNMEF, in comparison with leading existing ensemble methods; the experiments also serve to validate the effectiveness of several specific details within our approach. First, in Subsections 6.5.2, 6.5.3, and 6.5.4, we compare our method with several classical approaches across several nonlinear dynamical systems — Lorenz '96, Kuramoto-Sivashinsky, and Lorenz '63. In Subsection 6.5.5, we compare the computational time between our MNMEF and other benchmarks. In Subsection 6.5.6, we show the robustness of our method to the inherent randomness in test trajectories. In Subsection 6.5.7, we conduct experiments to study the learning-based inflation and localization methods proposed in Subsections 6.4.2.2 and 6.4.2.3. Subsequently, in Subsection 6.5.8, we perform experiments to illustrate the efficiency and effectiveness of our proposed fine-tuning method, introduced in Subsection 6.4.4. The code for our method is publicly available[3] and can be used to reproduce all of our experiments.

We reiterate that all reported experiments are for nonlinear problems. Our methodology is based on learning corrections to the Kalman filter structure inherent in the EnKF. For nonlinear problems there is bias in the EnKF which can be effectively learned and corrected for, and this is the essence of our approach. For linear problems there is no bias in the mean-field limit (although there will still be a bias with a finite ensemble (Sacher and Bartello, 2008)), so our methodology would attempt to fit neural networks to what is largely noise. Without any regularization or early stopping, this can lead to filter divergence for large enough training epochs.

6.5.1 Experimental Set-Up

In the following three Subsections 6.5.2, 6.5.3, and 6.5.4, we compare our method against several benchmark approaches:

1. **Localized EnKF Perturbed Observation (EnKF)** Burgers, Van Leeuwen, and Evensen (1998) and Jeffrey L Anderson (2001): This is the classic implementation of EnKF that employs a stochastic update approach, where the observations are perturbed by adding random noise samples drawn from the observation noise distribution.

2. **Ensemble Square Root Filter (ESRF)** Tippett et al. (2003) and Pavel Sakov

[3] `https://github.com/wispcarey/DALearning`

and Oke (2008): This method is a deterministic variant of the EnKF. It directly transforms the ensemble using matrix square root operations to match the second order statistics of the EnKF, but without requiring stochastic perturbations.

3. **Local Ensemble Transform Kalman Filter (LETKF)** Brian R. Hunt, Kostelich, and Szunyogh (2007): This method builds upon the Ensemble Transform Kalman Filter (ETKF) (C. H. Bishop, Etherton, and Majumdar, 2001) by incorporating spatial localization. ETKF is also a deterministic version of EnKF. Under linear–Gaussian assumptions and in the mean-field limit, ESRF and ETKF are theoretically equivalent, but they differ in practical implementation.

4. **Iterative Ensemble Kalman Filter (IEnKF)** Pavel Sakov, Dean S. Oliver, and Laurent Bertino (2012): This method enhances the EnKF (perturbed observation) by incorporating a variational scheme to better handle strongly nonlinear systems. It iteratively solves a minimization problem between adjacent time steps, progressively refining the solution for the analysis step through multiple iterations. In our experiments, we set the maximum number of iterations to 10 and use a convergence tolerance of 10^{-5}.

Remark 6.5.1. *Inflation and localization are important for DA in high-dimensional systems. In our experiments, all benchmark filters (EnKF, ESRF, LETKF, and IEnKF) employ post–analysis inflation as in* (6.71)*. For the higher-dimensional Lorenz–96 (Subsection 6.5.2) and Kuramoto–Sivashinsky (Subsection 6.5.3) systems, we apply localization to the* ***EnKF*** *and* ***LETKF****. We employ the Gaspari–Cohn (GC) function (Gaspari and Cohn, 1999) as the distance-to-weight function for the localization according to the DAPPER package (Raanes, Yumeng Chen, and Grudzien, 2024). We do* not *localize ESRF or IEnKF for the following reasons:*

1. *LETKF can be viewed as ETKF with localization, and ETKF is theoretically equivalent to ESRF in the mean-field/linear–Gaussian regime; adding localization to ESRF would thus be essentially redundant with LETKF.*

2. *For IEnKF, a localized implementation is omitted because it requires multiple iterations with the dynamic model integrations at each step, making a joint grid search over inflation and localization computationally prohibitive for Lorenz '96 and KS, and because without careful radius tuning the method exhibited numerical instability (frequent NaNs due to filter divergence).*

The hyperparameters for inflation and localization are optimized separately for each ensemble size via grid search.

In addition to the methods described above, we evaluate our MNMEF approach at various ensemble sizes. Unless otherwise specified, our MNMEF is pretrained with ensemble size $N = 10$ as described in Subsection 6.4.3. We then examine our model's performance across different ensemble sizes $N \in \{5, 10, 15, 20, 40, 60, 100\}$ to demonstrate the generalizability of our approach.

Our evaluation includes two variants of our method:

1. **Pretrained MNMEF**: Trained once at $N = 10$ for 1000 epochs and applied directly to all ensemble sizes, demonstrating transfer capability across different ensemble configurations. This appears as **'Pretrain'** in our figures.

2. **Fine-tuned MNMEF**: Fine tune the pretrained model specific to each target ensemble size for 20 epochs, as detailed in Subsection 6.4.4. This appears as **'Tuned'** in our figures.

For a fair comparison, all benchmark methods (EnKF, LETKF, and IEnKF) are tuned by RMSE with respect to the true state $v_j^\dagger$. Specifically, hyperparameters for inflation and localization are selected via comprehensive grid searches at multiple ensemble sizes to minimize true-state RMSE. This approach ensures that all benchmark methods achieve close to their optimal performance; the benchmarks thus present a robust test of our proposed methodology. Notably, we performed hyperparameter selection for the benchmarks directly on their test set performance, without using a validation set, which only strengthens the benchmark methods' advantage in our comparisons.

The performance of the methods is evaluated using the difference between the estimated state (i.e., the mean of particles) and the ground truth state over the entire trajectory. In the continuous-time setting, let $\mathbf{v}^\dagger(t) : [0, T] \to \mathbb{R}^{d_v}$ denote the ground truth trajectory between time 0 and T, and let $\mathbf{v}(t) : [0, T] \to \mathbb{R}^{d_v}$ denote the estimated state trajectory. Let $\|\cdot\|_2$ denote the Euclidean norm in $\mathbb{R}^{d_v}$. Then, the performance can be evaluated using the relative $L^1 := L^1([0, T]; \mathbb{R}^{d_v})$ error:

$$\mathcal{E}(\mathbf{v}, \mathbf{v}^\dagger) = \frac{\|\mathbf{v} - \mathbf{v}^\dagger\|_{L^1}}{\|\mathbf{v}^\dagger\|_{L^1}} = \frac{\int_0^T \|\mathbf{v}(t) - \mathbf{v}^\dagger(t)\|_2 dt}{\int_0^T \|\mathbf{v}^\dagger(t)\|_2 dt}. \tag{6.88}$$

In the discrete-time setting, we interpret the discrete time index j as time $t_j = j\Delta t$, where $\Delta t = T/J$ is the time step size. The estimated trajectory is given by the sequence $V^{(N)} = \{\overline{v}_j^{(N)}\}_{j=1}^J$ where $\overline{v}_j^{(N)} = \frac{1}{N}\sum_{n=1}^N v_j^{(n)}$, and the ground truth trajectory is given by $V^\dagger = \{v_j^\dagger\}_{j=1}^J$. The relative discrete L^1 error can then be defined as

$$\mathcal{E}(V^{(N)}, V^\dagger) = \frac{\sum_{j=1}^J \|\mathbf{v}(j\Delta t) - \mathbf{v}^\dagger(j\Delta t)\|_2 \, \Delta t}{\sum_{j=1}^J \|\mathbf{v}^\dagger(j\Delta t)\|_2 \, \Delta t} = \frac{\sum_{j=1}^J \|\overline{v}_j - v_j^\dagger\|_2}{\sum_{j=1}^J \|v_j^\dagger\|_2}. \tag{6.89}$$

This error metric can be interpreted as a form of relative root-mean-square error (RMSE). At each time step j, the quantity $\|\overline{v}_j - v_j^\dagger\|_2$ corresponds to the root-mean-square error over the d_v-dimensional state vector. The overall metric thus represents a time-averaged spatial RMSE, normalized by the magnitude of the ground truth trajectory. In what follows, we refer to this error $\mathcal{E}(V^{(N)}, V^\dagger)$ as the relative RMSE (R-RMSE).

For M_{test} trajectories $\{V_m^\dagger\}_{m=1}^{M_{\text{test}}}$ and their corresponding estimated trajectories $\{V_m^{(N)}\}_{m=1}^{M_{\text{test}}}$, the averaged R-RMSE $\overline{\mathcal{E}}$ is given by

$$\overline{\mathcal{E}} = \frac{1}{M_{\text{test}}} \sum_{m=1}^{M_{\text{test}}} \mathcal{E}(V_m^{(N)}, V_m^\dagger). \tag{6.90}$$

In the experiments presented in the remainder of this section, we adopt the averaged R-RMSE $\overline{\mathcal{E}}$ as the primary evaluation metric, allowing us to ameliorate the effect of randomness inherent in each individual run and providing a more reliable assessment of the overall performance of different methods. Unless otherwise specified, the number of test trajectories is fixed at $M_{\text{test}} = 64$.

To further highlight the advantages of our method, we consider the relative improvement in terms of R-RMSE $\mathcal{E}_{\text{RI}}$ defined as

$$\mathcal{E}_{\text{RI}} = \frac{\overline{\mathcal{E}}_{\text{benchmark}} - \overline{\mathcal{E}}_{\text{ours}}}{\overline{\mathcal{E}}_{\text{benchmark}}}. \tag{6.91}$$

This metric quantifies the percentage reduction in R-RMSE achieved by our method compared to the benchmark method. When $\mathcal{E}_{RI} < 0$, our method performs worse than the benchmark.

We apply the following general hyperparameter setting in all of our experiments unless otherwise specified. The dynamic operator Ψ in Equation (6.1) is obtained by solving a differential equation using the fourth-order Runge–Kutta (RK4) (Stoer

et al., 1980) method. The time step Δt denotes the interval between consecutive observations. For accurate numerical integration of the dynamics, we perform multiple RK4 steps within each interval Δt, utilizing an integration step size smaller than Δt. The dynamic noise $\xi \sim \mathcal{N}(0, \Sigma)$ is typically set at $\Sigma = \sigma_v^2 I$ where I is the identity matrix and $\sigma_v = 10^{-3}$ or 0. For the observation noise, $\eta \sim \mathcal{N}(0, \Gamma)$, we define its covariance matrix as $\Gamma = \sigma_y^2 I$, where we set σ_y to either 0.7 or 1.0.

6.5.2 Lorenz '96

The Lorenz '96 model is a widely-used set of ordinary differential equations that possess features of large-scale atmospheric dynamics (Edward N Lorenz, 1996b). The model takes the form of a damped-driven linear system, with additional energy conserving quadratic nonlinearity that induces cyclic interactions (waves) among the components of the system; this makes it particularly valuable for studying atmospheric predictability. The system dynamics are governed by the following equations:

$$\frac{du_i}{dt} = \big(u_{i+1} - u_{i-2}\big)u_{i-1} - u_i + F, \quad i = 1, 2, \ldots, d. \tag{6.92}$$

The parameter settings for the Lorenz '96 system are shown in Table 6.1. The forcing parameter $F = 8$ places the Lorenz '96 system in a chaotic regime when $d = 40$.

Table 6.1: Lorenz '96 System Settings

Category	**Values**
Parameters	$F = 8$
States	$v = (u_1, u_2, \ldots, u_{40}) \in \mathbb{R}^d, \quad d = 40$
Observations	$h(v) = (u_1, u_5, \cdots, u_{33}, u_{37}) \in \mathbb{R}^{d_{\text{obs}}}, \quad d_{\text{obs}} = 10$
Time Step	Observation time step: $\Delta t = 0.15$; 5 RK4 integration steps: $\Delta t/5 = 0.03$

In Figure 6.1, we compare the performance of our pretrained MNMEF and fine-tuned MNMEF against four benchmark methods: EnKF, ESRF, LETKF, and IEnKF, on the Lorenz '96 system. The upper row of plots in Figure 6.1 displays the R-RMSE (6.90) for observation noise levels $\sigma_y = 0.7$ (left plot) and $\sigma_y = 1.0$ (right plot). The lower row of plots in Figure 6.1 shows the relative improvement $\mathcal{E}_{\text{RI}}$ (6.91) of our fine-tuned method as compared to EnKF, IEnKF, and LETKF for these respective noise levels.

We observe that our fine-tuned model consistently outperforms all benchmarks. Indeed, our fine-tuned method achieves substantial improvement for small ensemble

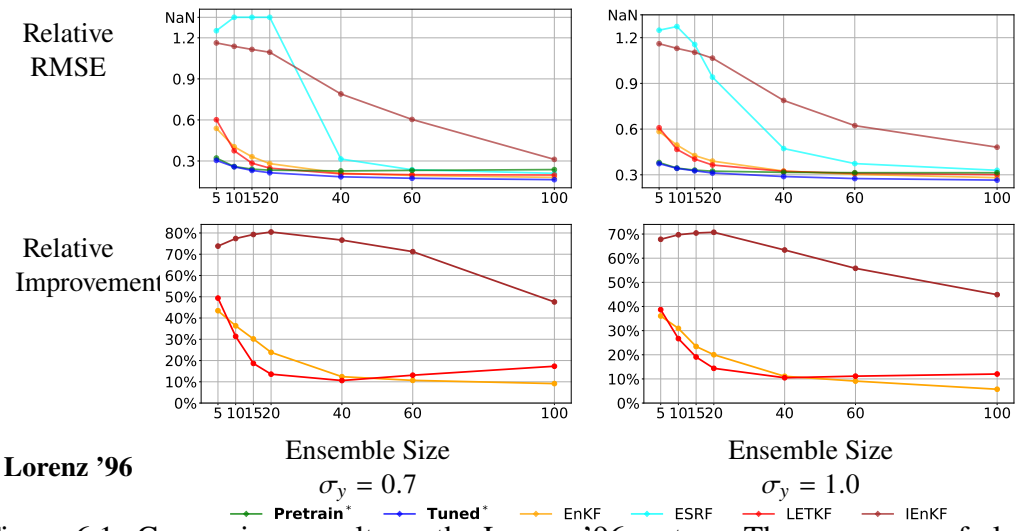


Figure 6.1: Comparison results on the Lorenz '96 system. The upper row of plots shows direct R-RMSE comparisons between different methods, while the lower row illustrates the relative improvement of our fine-tuning method compared to benchmarks. These comparisons are presented for observation noise levels $\sigma_y = 0.7$ (left column) and $\sigma_y = 1.0$ (right column). Our proposed MNMEF method is highlighted in the legends with bold font and an asterisk (e.g. **Pretrain***). In summary, our fine-tuned MNMEF model consistently outperforms benchmarks, showing substantial improvements for small ensembles and a 15-20% advantage over LETKF at larger ensemble sizes (eg. 60, 100).

sizes and maintains a notable advantage at larger ensemble sizes, outperforming LETKF by approximately 15–20%. Our pretrained model, trained with $N = 10$ ensembles also behaves competitively, but performs slightly worse than LETKF for large ensemble size 40, 60, and 100.

To further illustrate the performance differences between our MNMEF and the best-performance benchmark LETKF, we provide visualizations in Figure 6.2 and Figure 6.3 for $N = 10$ and $\sigma_y = 1.0$. The visualization is on the last 100 time steps (from 1401 to 1500) from a test trajectory of length 1500 and $\Delta t = 0.15$. In both figures, our MNMEF approach is the one pretrained with the ensemble size $N = 10$.

Figure 6.2 presents a comparison of the estimated state trajectory. Panel (a) shows the ground truth, and panel (b) shows the sparse observations. The subsequent rows compare MNMEF and LETKF. Visually, MNMEF displays smaller magnitudes for the absolute error (panel (d)). In addition, the ensemble spread (panel (e)) for MNMEF is more consistent than LETKF. Figure 6.3 focuses on the estimation of two specific dimensions: one observed (Dimension 1, panel (a)) and one unobserved (Dimension 2, panel (b)). For the observed dimension, both MNMEF and LETKF

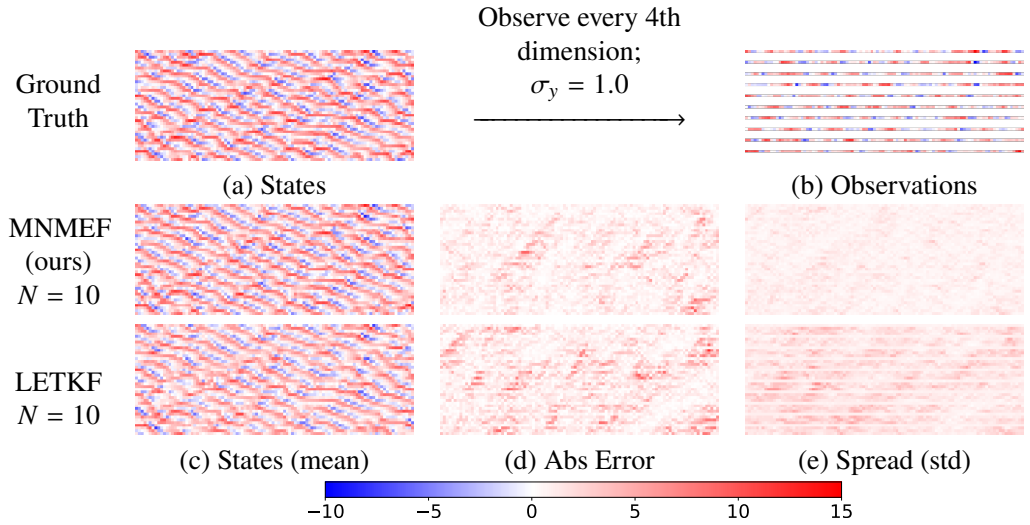


Figure 6.2: Visualization of one test trajectory (time steps 1401-1500, $\Delta t = 0.15$) for Lorenz '96 states (vertical axis) over time (horizontal axis)with the observation noise $\sigma_y = 1.0$. Panel (a): Ground-truth states (unknown). Panel (b): Observations (known, every 4th dimension observed). Rows 2-3 show our method MNMEF pretrained on $N = 10$, and the benchmark LETKF with the ensemble size $N = 10$. Panel (c): state estimation (ensemble mean), Panel (d): absolute error of mean with respect to the ground truth, and Panel (e): ensemble spread (standard deviation).

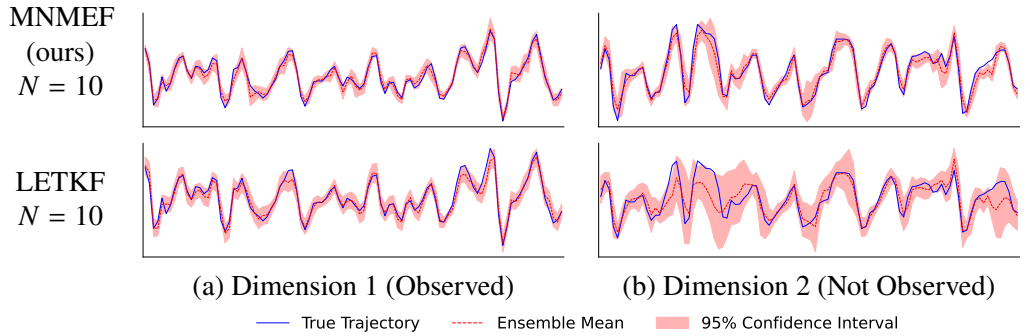


Figure 6.3: Visualization of two dimensions (index 1 and 2) in one test trajectory (time steps 1401-1500, $\Delta t = 0.15$) for Lorenz '96 state values (vertical axis) over time (horizontal axis) with the ensemble size $N = 10$ and the observation noise $\sigma_y = 1.0$. Panel (a): Dimension 1, observed. Panel (b): Dimension 2, not observed. The first row is our method MNMEF, pretrained on $N = 10$, and the second row is the benchmark LETKF. The 95% confidence intervals shown in figures are calculated as the ensemble mean $\pm$ 1.96 $\times$ the ensemble standard deviation. In summary, MNMEF performs significantly better on the unobserved dimension and comparably to LETKF on the observed dimension; meanwhile, MNMEF maintains an appropriate spread without suffering from filter degeneracy.

perform comparably well, with their ensemble means closely following the ground truth. However, for the unobserved dimension, MNMEF demonstrates a clear advantage in state estimation. Furthermore, MNMEF maintains an appropriate

ensemble spread while avoiding filter degeneracy.

6.5.3 Kuramoto-Sivashinsky

The Kuramoto—Sivashinsky (KS) equation (Kuramoto, 1978; Michelson and Sivashinsky, 1977) is a widely studied nonlinear partial differential equation that describes the evolution of instabilities in spatially extended systems, capturing complex spatiotemporal chaotic dynamics arising in flame fronts, thin fluid films, and reaction-diffusion systems. Due to its chaotic behavior and sensitivity to initial conditions, the KS equation serves as a valuable testbed for developing and evaluating data assimilation algorithms. With the periodicity in space imposed to identity $x = L$ and $x = 0$, the one-dimensional KS equation for $u : [0, L] \times \mathbb{R}^+ \to \mathbb{R}$ takes the form

$$\frac{\partial u}{\partial t} + \frac{\partial^4 u}{\partial x^4} + \frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} = 0, \quad (x, t) \in (0, L) \times \mathbb{R}^+, \tag{6.93a}$$

$$u(x, 0) = u_0, \quad \forall x \in [0, L]. \tag{6.93b}$$

Table 6.2: Kuramoto-–Sivashinsky (KS) System Settings

Category	**Values**
Parameters	$L = 32\pi$
States	$x_j = jL/128, \quad j = 0, 1, \ldots, 127$ $v = \big(u(x_0), u(x_1), \ldots, u(x_{127})\big) \in \mathbb{R}^d, \quad d = 128$
Observations	$h(v) = \big(u(x_0), u(x_8), \cdots, u(x_{120})\big) \in \mathbb{R}^{d_{\text{obs}}}, \quad d_{\text{obs}} = 16$
Time Step	Observation time step: $\Delta t = 1$; 4 ETDRK4 integration steps: $\Delta t/4 = 0.25$

The detailed experimental settings and parameters for the KS system are summarized in Table 6.2. Numerically, the KS equation (6.93) is discretized spatially using a Fourier pseudo-spectral method to handle periodic boundary conditions effectively. Time integration is performed with the exponential time-differencing fourth-order Runge–Kutta (ETDRK4) scheme (Kassam and Trefethen, 2005), which advances the stiff linear operator analytically and is therefore not subject to the explicit linear Courant–Friedrichs–Lewy (CFL) restriction ($\Delta t \leqslant C\Delta x^4$) that would apply to fully explicit RK methods; the time step is chosen based on accuracy and nonlinear resolution rather than linear stability.

In Figure 6.4, we compare the performance of our pretrained model and fine-tuned model against four benchmark methods: EnKF, ESRF, LETKF, and IEnKF, for the

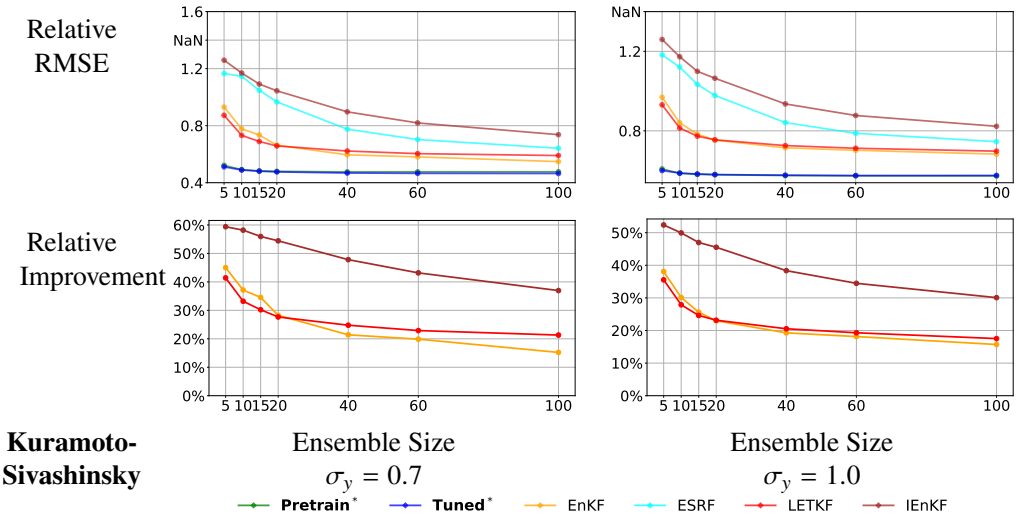


Figure 6.4: Comparison results on the Kuramoto—Sivashinsky (KS) system. The upper row of plots shows direct R-RMSE comparisons between different methods, while the lower row illustrates the relative improvement of our fine-tuning method compared to benchmarks. These comparisons are presented for observation noise levels $\sigma_y = 0.7$ (left column) and $\sigma_y = 1.0$ (right column). Our proposed MNMEF method is highlighted in the legends with bold font and an asterisk (e.g. **Pretrain***). In summary, our fine-tuned MNMEF model consistently outperforms benchmarks, showing substantial improvements for small ensembles and around 20% advantage over LETKF at larger sizes (eg. 60, 100).

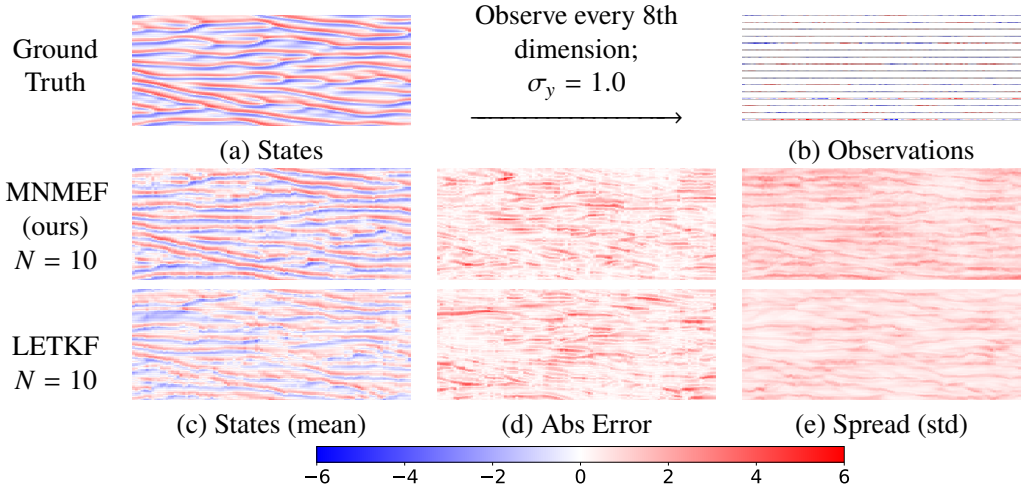


Figure 6.5: Visualization of one test trajectory (time steps 1901-2000, $\Delta t = 1$) for Kuramoto-Sivashinshy (KS) states (vertical axis) over time (horizontal axis) with the observation noise $\sigma_y = 1.0$. Panel (a): Ground-truth states (unknown). Panel (b): Observations (known, every 8th dimension observed). Rows 2-3 show our method MNMEF pretrained on $N = 10$, and the benchmark LETKF with the ensemble size $N = 10$. Panel (c): state estimation (ensemble mean), Panel (d): absolute error with ground truth, and Panel (e): ensemble spread (standard deviation).

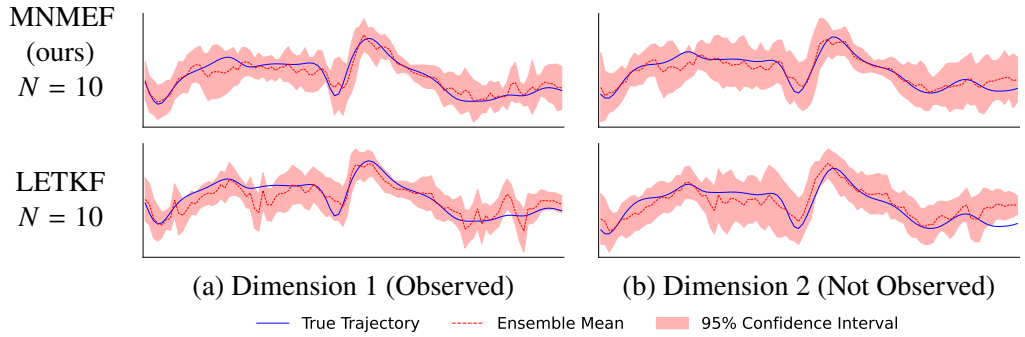


Figure 6.6: Visualization of two dimensions (index 1 and 2) in one test trajectory (time steps 1901-2000, $\Delta t = 1$) for Kuramoto-Sivashinshy (KS) state values (vertical axis) over time (horizontal axis) with the ensemble size $N = 10$ and the observation noise $\sigma_y = 1.0$. Panel (a): Dimension 1, observed. Panel (b): Dimension 2, not observed. The first row is our method MNMEF, pretrained on $N = 10$, and the second row is the benchmark LETKF. The 95% confidence intervals shown in figures are calculated as the ensemble mean $\pm$ 1.96 $\times$ the ensemble standard deviation. In summary, MNMEF performs slightly better on both the observed and the unobserved dimensions; meanwhile, MNMEF maintains an appropriate spread without suffering from filter degeneracy.

KS system with $\sigma_y = 0.7$ and 1.0, and provide the relative improvement of our fine-tuned method as compared to EnKF, IEnKF, and LETKF.

The comparison between our pretrained MNMEF and fine-tuned MNMEF methods and the benchmark methods, for the KS dynamical system, slightly differs from what we observed under the Lorenz '96 dynamics. In the KS setting, fine-tuning provides minimal additional benefit, with both our pretrained and fine-tuned MNMEF models performing nearly identically. Both variants significantly outperform benchmark methods across all ensemble sizes. The advantage is particularly pronounced at smaller ensemble sizes. Even at the largest ensemble size ($N = 100$), both our pretrained and fine-tuned methods still maintain approximately 20% improvement over LETKF, the benchmark method with the best performance.

Similarly to the Lorenz '96 discussion in Subsection 6.5.2, we provide visualizations in Figure 6.2 and Figure 6.3 for $N = 10$ and $\sigma_y = 1.0$ to further illustrate the performance differences between our MNMEF and the best-performance benchmark LETKF on the KS system. The visualization is on the last 100 time steps (from 1901 to 2000) of a test trajectory with length 2000 and $\Delta t = 1$. In both figures, our MNMEF is the one pretrained with the ensemble size $N = 10$.

Figure 6.5 presents a comparison of the estimated state trajectory. Figure 6.6 visualizes the estimation of two specific dimensions: one observed (Dimension 1,

panel (a)) and one unobserved (Dimension 2, panel (b)). In summary, our MNMEF performs slightly better than LETKF on both the observed and the unobserved dimensions, which is verified by the R-RMSE quantitative comparison in Figure 6.4. MNMEF maintains an appropriate ensemble spread, similarly to the LETKF.

Remark 6.5.2 (Implementation of Localization). *It is worth noting that methods other than LETKF could potentially accommodate localization techniques. For example, LETKF is essentially the ESRF combined with localization. However, since our implementation is based on the DAPPER package, which only implements localization for LETKF, we do not consider localization for the other four methods. Our results on Lorenz '96 and KS problems sufficiently demonstrate the advantages of our approach over other methods, even without localization. For larger ensemble sizes (e.g., $N = 60, 100$), where the benefits of localization diminish, our method still significantly outperforms the alternatives, like IEnKF.*

6.5.4 Lorenz '63

The Lorenz '63 system is a simplified mathematical model for atmospheric convection (Edward N. Lorenz, 1963b), proposed prior to the Lorenz '96 model discussed in Subsection 6.5.2. It represents a highly simplified version of atmospheric dynamics with only three dimensions, described by the following set of ordinary differential equations:

$$\frac{dx}{dt} = \sigma(y - x), \tag{6.94a}$$

$$\frac{dy}{dt} = x(\rho - z) - y, \tag{6.94b}$$

$$\frac{dz}{dt} = xy - \beta z. \tag{6.94c}$$

Table 6.3: Lorenz '63 System Settings

Category	**Values**
Parameters	$\sigma = 10, \quad \rho = 28, \quad \beta = \frac{8}{3}$
States	$v = (x, y, z) \in \mathbb{R}^d, \quad d = 3$
Observations	$h(v) = x \in \mathbb{R}^{d_{\text{obs}}}, \quad d_{\text{obs}} = 1$
Time Step	Observation time step: $\Delta t = 0.15$; 5 RK4 integration steps: $\Delta t/5 = 0.03$

The parameter settings for the Lorenz '63 system are shown in Table 6.3. The three parameters σ, ρ, and β represent the Prandtl number, Rayleigh number, and geometric factor, respectively, chosen by Lorenz based on simplified thermal

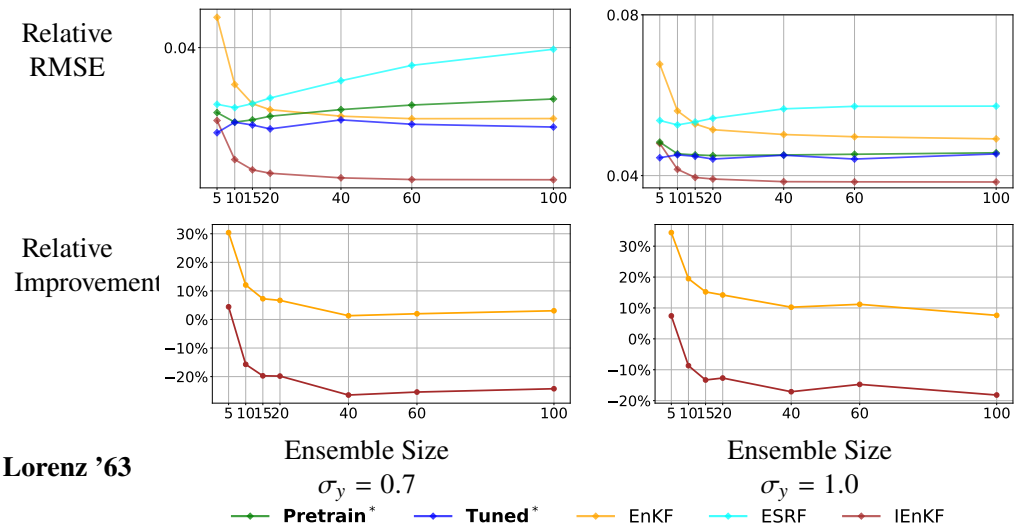


Figure 6.7: Comparison results on the Lorenz '63 system. The upper row of plots shows direct R-RMSE comparisons between different methods, while the lower row illustrates the relative improvement of our fine-tuning method compared to benchmarks. These comparisons are presented for observation noise levels $\sigma_y = 0.7$ (left column) and $\sigma_y = 1.0$ (right column). Our proposed MNMEF method is highlighted in the legends with bold font and an asterisk (e.g. **Pretrain***).

convection experiments to produce chaotic dynamics characterized by sensitivity to initial conditions.

In Figure 6.7, we compare the R-RMSE (6.90) performance of our pretrained MNMEF and fine-tuned MNMEF models against benchmark methods EnKF, ESRF, and IEnKF for the Lorenz '63 system. The upper row of plots in Figure 6.7 displays these R-RMSE comparisons for observation noise levels $\sigma_y = 0.7$ (left plot) and $\sigma_y = 1.0$ (right plot). The LETKF method is not included in this comparison since the system has only three dimensions, making localization unnecessary. For a fair comparison, we turn off the localization part in our architecture by setting all entries in the localization weight matrices to be 1. We only fine-tune the parameters θ_{gain} for F^{gain} (6.70) and θ_{infl} for F^{infl} (6.72) for ensemble sizes $N \neq 10$. The lower row of plots in Figure 6.7 shows the relative improvement $\mathcal{E}_{\text{RI}}$ (6.91) of our fine-tuned MNMEF method as compared to EnKF and IEnKF, for the respective noise levels shown.

We observe that the performance of different methods does not significantly change with increasing ensemble size. Our method consistently outperforms the EnKF and ESRF, but in the low-dimensional Lorenz '63 problem, it is outperformed by the IEnKF. As noted in Remark 6.5.3, IEnKF demonstrates strong performance in

low-dimensional, highly nonlinear problems while struggling in high-dimensional systems without localization. This is consistent with our findings in the higher-dimensional experiments in Subsections 6.5.2 and 6.5.3, where our method substantially outperforms IEnKF. Furthermore, the stability analysis in Subsection 6.5.6 demonstrates that our method exhibits higher stability (lower standard deviation across test trajectories) than IEnKF.

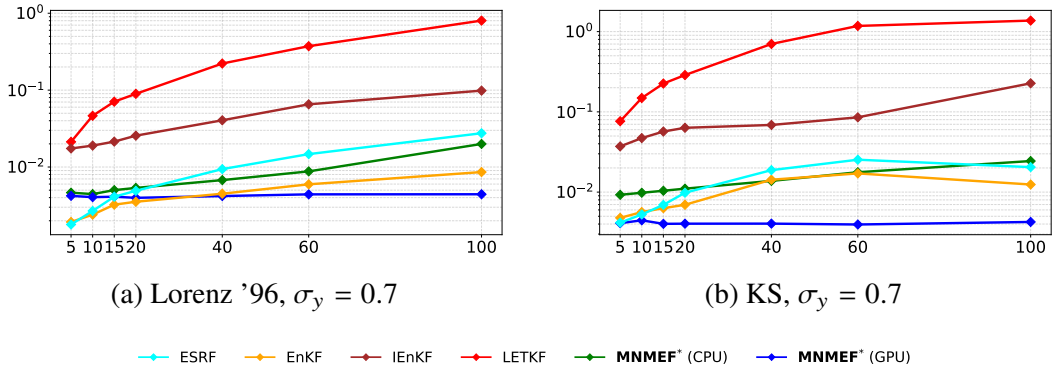
Remark 6.5.3 (IEnKF Dimensionality Challenges). *The IEnKF (Pavel Sakov, Dean S. Oliver, and Laurent Bertino, 2012) uses the same observational information as the standard EnKF, but reformulates the filtering problem as a lag-1 smoothing problem, and solves it iteratively to better handle nonlinearities. While the IEnKF demonstrates excellent performance in low-dimensional systems (e.g., the Lorenz '63 model, $d = 3$), its effectiveness diminishes as dimensionality increases (e.g., the Lorenz '96 model, $d = 40$). This deterioration occurs due to lack of localization in the implementation of the IEnKF we use; however, even with large ensembles with less need for localization, our method outperforms the IEnKF in the higher-dimensional systems. Moreover, IEnKF generally provides a substantial improvement over non-iterative methods only when the observations are sufficiently dense, as shown in Pavel Sakov, Dean S. Oliver, and Laurent Bertino (2012) using experiments with the Lorenz '96 model.*

6.5.5 Computational Cost

In this subsection, we compare the run time per assimilation step (s/step) of MNMEF against benchmark methods (EnKF, ESRF, LETKF, and IEnKF) under matched CPU-only settings to ensure a fair comparison.

For an ensemble of size N, the per-step cost of MNMEF scales as $O(N^2)$. The dominant terms are (i) the attention blocks in the set transformer, which entail pairwise interactions among N members, and (ii) the computation of the learned Kalman gain K_θ and its gradients, implemented via batched linear solves with $N \times N$ systems. We do not form matrix inverses explicitly. While classical baselines admit textbook complexities, their practical scaling depends on solver details, localization, and parallelization, so a full theoretical comparison is beyond the scope of this chapter.

To reflect practical efficiency, we conduct a CPU-only head-to-head timing in a single Python environment. All CPU timings are run on Intel(R) Core(TM) i9-14900KF. We follow implementations in DAPPER (Raanes, Yumeng Chen, and



(a) Lorenz '96, $\sigma_y = 0.7$ (b) KS, $\sigma_y = 0.7$

Figure 6.8: Run time (s/analysis step), σ_y=0.7 only. All methods are timed on CPU (Intel(R) Core(TM) i9-14900KF). MNMEF is comparable in speed to EnKF and ESRF, and is slightly slower than EnKF because MNMEF additionally computes the correction terms. MNMEF remains much faster than LETKF and IEnKF. For MNMEF, we additionally report GPU timing on a single GPU (RTX 4080 Super), shown as the blue curve. The GPU version shows very stable runtime across these ensemble sizes due to efficient parallelization.

Grudzien, 2024) and provide our own CPU-based implementations of EnKF, ESRF, LETKF, and IEnKF. We deliberately avoid GPU ports for these baselines because methods such as LETKF and IEnKF do not map cleanly to efficient, large-batch vectorization on current accelerators.

We evaluate on Lorenz '96 and KS with the same configurations as in Subsections 6.5.2 and 6.5.3. For each ensemble size N and observation noise level $\sigma_y = 0.7$, we report the mean of the per-step run time (s/analysis step), averaged over trajectories and time steps. We do not distinguish the pretrained and fine-tuned variants of our MNMEF since they share the same architecture. The set-transformer parameter counts follow Table 6.5. Results are visualized in Figure 6.8. MNMEF is comparable in speed to EnKF and ESRF, and is slightly slower than EnKF because MNMEF additionally computes the correction terms. MNMEF remains much faster than LETKF and IEnKF.

For MNMEF, we provide an additional GPU-based run time in Figure 6.8, since its original implementation is designed for GPU. MNMEF's additional GPU timings are obtained on a single NVIDIA RTX 4080 Super. The GPU version of MNMEF shows very stable runtime across these ensemble sizes due to efficient parallelization.

6.5.6 Robustness To Inherent Randomness

Due to the inherent randomness in both the filtering problem and the algorithm, the same method may exhibit varying performance across different test trajectories. To

mitigate this randomness and obtain a more reliable assessment of each method's effectiveness, we evaluate their performance on M_{test} trajectories and report the mean R-RMSE $\overline{\mathcal{E}}$ (6.90). While the mean R-RMSE provides a measure of overall accuracy, we are interested in the stability of each method across different trajectories. To quantify this stability, we compute the standard deviation (std) of the R-RMSE values across all test trajectories $\{V_m^\dagger\}_{m=1}^{M_{\text{test}}}$ and their corresponding estimated trajectories $\{V_m\}_{m=1}^{M_{\text{test}}}$, defined as

$$\sigma_{\mathcal{E}} = \sqrt{\frac{1}{M_{\text{test}}} \sum_{m=1}^{M_{\text{test}}} \left(\mathcal{E}(V_m, V_m^\dagger) - \overline{\mathcal{E}}\right)^2} \tag{6.95}$$

A smaller $\sigma_{\mathcal{E}}$ indicates better stability of the method. Therefore, in this section, we consider methods with lower std values to be more stable and thus preferable.

Remark 6.5.4. *In ensemble-based data assimilation, variance- or standard-deviation–type quantities are commonly used to characterize ensemble spread (Bach, Baptista, Sanz-Alonso, et al., 2025). In our work, the spread is visualized in Figs. 6.3 and 6.6 via the shaded bands: the 95% confidence interval is provided as the ensemble mean* $\pm 1.96\times$ *ensemble standard deviation. Our focus here is state estimation rather than full uncertainty calibration, so we do not present a quantitative comparison of spread metrics. Note that the stability measure* $\sigma_{\mathcal{E}}$ *in* (6.95) *serves a different purpose: it assesses the consistency of estimation errors across test trajectories, not the dispersion of ensemble members.*

In Table 6.4, we show the std of different methods on the dataset Lorenz '96, KS and Lorenz '63 with the observation noise $\sigma_y = 1.0,\ 0.7$. The std values in the table is the averaged std over the ensemble sizes $N = 5, 10, 15, 20, 40, 60, 100$. Based on Table 6.4, we observe that our pretrained MNMEF and fine-tuned MNMEF demonstrate comparable standard deviation values, both of which are significantly lower than all benchmark methods. This indicates that our approaches maintain more consistent performance across different test trajectories. The enhanced stability suggests that our methods are more reliable in practical applications.

6.5.7 Inflation and Localization

In this section, we will conduct further experiments on learning the inflation (Subsection 6.4.2.2) and localization (Subsection 6.4.2.3) components in the architecture of our MNMEF method. These experiments demonstrate that what our method

Table 6.4: Standard deviation (std) of the relative root mean square error (R-RMSE) as defined in (6.95). We consider both **Pretrain*** and **Tuned*** variants of our MNMEF method. The std shown in the table is an averaged std over ensemble sizes $N = 5, 10, 15, 20, 40, 60, 100$. A smaller std indicates better stability of the method. The lowest std in each column is highlighted in bold. The standard deviations for both the pretrained and fine-tuned variants of our MNMEF method are similar and consistently lower than those of the benchmark methods, indicating greater robustness of our approach to the inherent randomness across different test trajectories.

Method	Lorenz '96 (6.92)		KS (6.93)		Lorenz '63 (6.94)	
	$\sigma_y = 1.0$	$\sigma_y = 0.7$	$\sigma_y = 1.0$	$\sigma_y = 0.7$	$\sigma_y = 1.0$	$\sigma_y = 0.7$
Pretrain*	**1.03e-2**	1.26e-2	**8.60e-3**	1.07e-2	**1.78e-3**	1.40e-3
Tuned*	1.08e-2	**1.12e-2**	8.91e-3	**1.07e-2**	1.79e-3	**1.29e-3**
EnKF	2.16e-2	3.10e-2	2.66e-2	3.15e-2	3.41e-3	2.32e-3
ESRF	3.09e-2	1.95e-2	2.59e-2	3.27e-2	2.78e-3	2.01e-3
LETKF	2.63e-2	6.13e-2	2.66e-2	3.58e-2	-	-
IEnKF	1.83e-2	2.61e-2	2.23e-2	3.03e-2	2.15e-3	1.44e-3

learns has many similarities with what is used in the application of inflation and localization techniques to the benchmark methods.

6.5.7.1 Inflation Experiments

In Subsection 6.4.2.2, we introduce our learning-based inflation scheme (6.73) which depends on the learned output $\widehat{u}_\theta^{(n)}$ from the neural network F^{infl} (6.72).

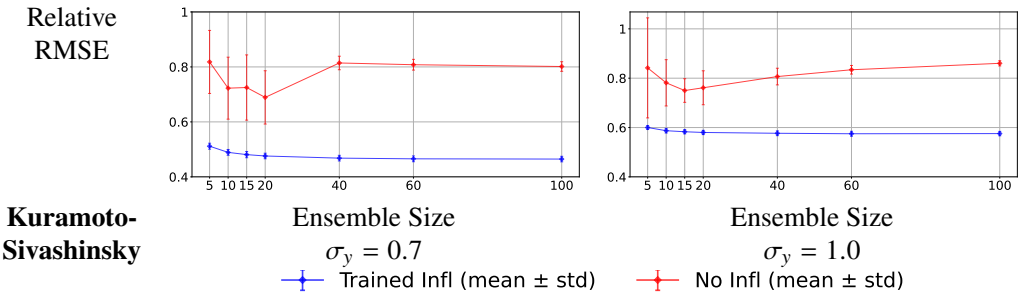


Figure 6.9: Comparison of the trained inflation versus the no inflation scenario when applied to the KS system (6.93), using our MNMEF method, fine-tuned for different ensemble sizes. The plots compare R-RMSE with and without inflation at observation noise levels $\sigma_y = 0.7$ (left plot) and $\sigma_y = 1.0$ (right plot). In summary, our trained inflation effectively improves the performance, but even without inflation, our MNMEF does not completely fail.

We are particularly interested in quantifying the effect of our proposed inflation scheme on the overall performance of our method. Additionally, despite not pro-

viding the inflation term $\widehat{u}_\theta^{(n)}$ with information about the true observation $y^\dagger$ or observation noise Γ in F^{infl}, $\widehat{u}_\theta^{(n)}$ has significantly more degrees of freedom than classical inflation approaches. This raises concerns that $\widehat{u}_\theta^{(n)}$ might exceed the scope of inflation: that it might not act merely as a correction term but might completely overwhelm the updates, thus rendering other parts of our architecture ineffective. We show that this is not the case, but at the same time we showcase the benefits of including this learned inflation-like correction.

For our experimental approach, we first pretrain the models and then fine-tune them separately for various ensemble sizes. During inference, we deliberately disable the learned inflation by setting $\widehat{u}_\theta^{(n)} = 0$ and analyze the resulting change in the average R-RMSE $\overline{\mathcal{E}}$ (6.90). We stress that this manipulation does not represent an optimal or even well-trained MNMEF filter without inflation; instead, it intentionally removes the correction term to create a more challenging robustness test—probing whether the filter remains stable and avoids divergence in the absence of inflation. These experiments are conducted on the KS dynamical system (6.93) only.

Our results are presented in Figure 6.9. We compare the mean R-RMSE $\overline{\mathcal{E}}$ between our standard architecture with the trained inflation (blue curve) and the no inflation version by setting $\widehat{u}_\theta^{(n)} = 0$ (red curve) for observation noise $\sigma_y = 0.7$ and 1.0. The results in Figure 6.9 clearly demonstrate the beneficial impact of our trained inflation approach. In addition, our method remains functional, with increased R-RMSE values, when we remove inflation by setting $\widehat{u}_\theta^{(n)} = 0$ (equivalent to $\alpha = 1$ in the EnKF). This confirms that our experiment, while intentionally suboptimal, demonstrates that the proposed MNMEF remains numerically stable and does not diverge even in this deliberately more difficult configuration. This addresses our concern, because, if inflation were dominating the architecture by masking the contribution of the state $v^{(n)}$, removing it would cause model collapse.

6.5.7.2 Localization Experiments

According to Subsection 6.4.2.3, our learning-based localization approach essentially learns a parameterized function g_θ, which maps distance to weight values. During training, we only enforce the constraint that the output of this function must lie within the interval [0,2], i.e., $g_\theta : \mathbb{R} \to [0, 2]$. Unlike traditional localization functions used in EnKF such as the Gaspari–Cohn function, our approach does not impose any explicit constraints on the shape or monotonicity of g_θ.

In this subsection, we present experimental comparisons between our learned func-

tion g_θ and the Gaspari–Cohn (GC) function with the optimal localization radius, found by grid-search, for LETKF; see Figure 6.10 . For ensemble sizes $N = 5, 10, 15, 20, 40, 60, 100$, the optimal GC radius values chosen are $1, 1, 2, 3, 4, 4, 6$ (multiplied by $\sqrt{10/3}$ when implemented (Raanes, Yumeng Chen, and Grudzien, 2024)), respectively. The learned function g_θ (shown as red curves with the mean and standard deviation of the learned g_θ across assimilation time steps) demonstrates a similar shape to the GC function with optimally chosen radius (blue curves). Notably, our approach dynamically adjusts the localization weights at each assimilation step based on ensemble size and distribution, while the GC function remains static throughout the assimilation process.

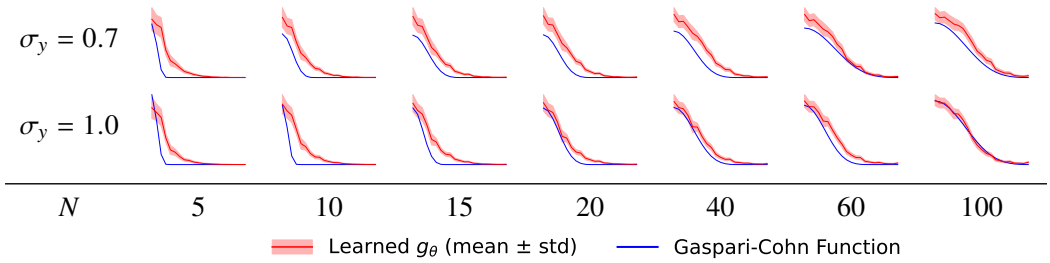


Figure 6.10: Comparison of our learned distance-to-weight function g_θ (red, mean ± 1 std across different time steps due to its adaptivity) with the LETKF Gaspari–Cohn (GC) function (blue) on the Lorenz '96 model for $\sigma_y = 1.0,\ 0.7$ and ensemble sizes $N = 5, 10, 15, 20, 40, 60, 100$. The learned distance-to-weight function is from our MNMEF pretrained on $N = 10$ and fine-tuned for different ensemble sizes. The LETKF GC radius parameters are optimally selected via grid search. Remarkably, even without any explicit constraints on the shape of g_θ, the learned weights naturally decrease as the distance increases, similar to the empirical GC function in LETKF.

The results in Figure 6.10 demostrates the effectiveness of our learned localization. Despite imposing no explicit constraints on shape or monotonicity, the learned function g_θ naturally exhibits similar behavior to the GC function: it decreases monotonically with distance and shares a similar shape. A key difference is that our function occasionally exceeds values of 1 at short distances, reflecting a combined localization and inflation effect where covariance between closely spaced dimensions is amplified, since we are learning the inflation and localization simultaneously. This adaptivity allows g_θ to dynamically adjust to the ensemble distribution, leading to improved performance over the fixed GC function, as supported by RMSE results in Subsections 6.5.2 and 6.5.3.

Remark 6.5.5 (Using GC localization directly). *Experimental results suggest that one can use the GC function to avoid training the localization. However, this*

approach still requires selecting the radius hyperparameter and typically yields inferior performance compared to our learned localization.

6.5.8 Fine-Tuning for Different Ensemble Size

According to Subsection 6.4.4, our proposed fine-tuning strategy freezes parameters θ_{ST} of the set transformer F^{ST} (6.67), while updating parameters θ_{gain}, θ_{infl} and θ_{loc} of the MLPs that control the gain (6.70), inflation (6.72), and localization (6.78). Here we conduct experiments to illustrate the efficiency and effectiveness of our proposed fine-tuning approach.

Firstly, we consider the time complexity between the pretraining stage and fine-tuning stage. Let P_{pretrain} and P_{ft} denote the total number of pretraining and fine-tuning parameters respectively, i.e.

$$P_{\text{pretrain}} = |\theta_{\text{ST}}| + |\theta_{\text{gain}}| + |\theta_{\text{infl}}| + |\theta_{\text{loc}}|, \tag{6.96a}$$

$$P_{\text{ft}} = |\theta_{\text{gain}}| + |\theta_{\text{infl}}| + |\theta_{\text{loc}}|. \tag{6.96b}$$

For fixed dynamic and training trajectories, the time complexity for pretraining with the ensemble size N is $O\big(ENP_t\big)$ while the fine-tuning with the ensemble size N' has the time complexity $O\big(E'N'P_{ft}\big)$, where E and E' are the epochs for pretraining and fine-tuning, respectively.

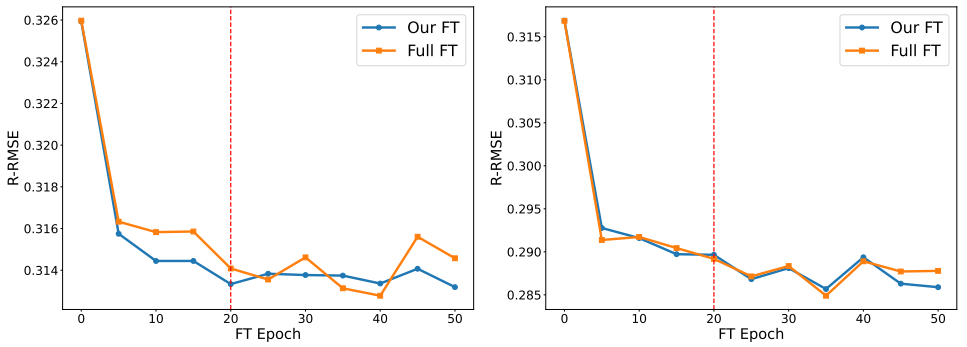
We provide a detailed comparison in Table 6.5, presenting the total number of pretraining parameters P_{pretrain}, fine-tuning parameters P_{ft}, and the number of epochs E used during the pretraining and fine-tuning phases. When the ensemble size during pretraining (N) is close to the ensemble size used in fine-tuning (N'), Table 6.5 shows that the fine-tuning stage requires less than 1% of the pretraining computational time. To quantitatively support this observation, we present detailed computational time comparisons in Table 6.6. All training times reported here are obtained using a single RTX 4080 Super GPU. The training batch sizes are adaptively chosen to fully use the 16 GB memory.

Table 6.5: Comparison of total parameters, fine-tuning parameters, and epochs for various datasets.

Dataset	Total Params	FT Params	Pretrain Epochs	FT Epochs
Lorenz '96	341551	63343	1000	20
KS	374289	90065	1000	20
Lorenz '63	309383	34119	1000	20

Table 6.6: Comparison of pretraining and fine-tuning computational time for observation noise $\sigma_y = 1.0$ with different ensemble sizes on a single RTX 4080 Super GPU. The ratio is between the computation time of the fine-tuning stage and the time of the pretraining stage. The fine-tuning time increases with the ensemble size, but it remains significantly faster compared to the pretraining stage.

Ens Size	$N = 10$	$N' = 20$		$N' = 40$		$N' = 100$	
Dataset	Pretrain (hrs)	FT (hrs)	Ratio	FT (hrs)	Ratio	FT (hrs)	Ratio
Lorenz '96	4.03	0.051	1.27%	0.065	1.61%	0.133	3.30%
KS	8.22	0.106	1.29%	0.132	1.61%	0.260	3.16%
Lorenz '63	1.82	0.022	1.21%	0.038	2.09%	0.086	4.73%



(a) Ensemble Size 20 (b) Ensemble Size 40

Figure 6.11: Comparison of R-RMSE across training epochs on the Lorenz '96 model for two fine-tuning strategies under two ensemble sizes, (a) 20 and (b) 40. The vertical dashed red line indicates the stopping epoch of our proposed fine-tuning approach. Results demonstrate that our fine-tuning on a small subset of parameters achieves comparable RMSE performance to fine-tuning all parameters.

Combining the information from Tables 6.5 and 6.6, we observe that our fine-tuning strategy demonstrates remarkable efficiency compared to the pretraining stage. As detailed in Subsections 6.5.4–6.5.3, this highly efficient fine-tuning significantly improves the model's performance, particularly when the ensemble sizes for inference and pretraining differ substantially.

To further validate the effectiveness of our proposed fine-tuning approach, we perform additional experiments comparing our method (Subsection 6.4.4) against alternative fine-tuning strategies. Specifically, we investigate and quantitatively compare the RMSE performance of the following fine-tuning scenarios: (1) fine-tuning all model parameters and (2) fine-tuning for a longer period (i.e., more epochs).

Figure 6.11 shows RMSE results from fine-tuning experiments on the Lorenz '96 model with ensemble sizes $N' = 20, 40$, using a model pretrained with ensemble

size $N = 10$. We compare our proposed method from Subsection 6.4.4 against fine-tuning all parameters, both running up to 50 epochs. The results demonstrate that both approaches achieve nearly identical RMSE performance, with minimal improvement observed after 20 epochs. Combined with our efficiency analysis in Tables 6.5 and 6.6, these findings confirm that fine-tuning all parameters is unnecessary, and that 20 epochs is sufficient for the near optimal performance.

6.6 Conclusions

In this chapter, we propose a novel machine learning-based approach, MNMEF, for state estimation in data assimilation. Our method features a scheme which subsumes the ensemble Kalman filter (EnKF), as a special case, ensuring optimality for linear, Gaussian problems, while extending beyond Gaussian applications through our learnable correction terms. A key advantage of our method is its ability to be trained at one ensemble size and then directly applied, possibly with a cheap fine-tuning of a small subset of parameters, for different ensemble sizes. This property is a consequence of the mean-field interpretation of the set transformer, the machine learning architecture underlying the methodology. Indeed, in this chapter we introduce neural operators acting on the metric space of probability measures, which we call *measure neural mappings* (MNM). We generalize transformers to this setting and show how the set transformer itself may be viewed as a particle approximation of such a mean-field model.

Our method adopts a form similar to the Kalman filter in the mean-field perspective, with the core innovation being a parameterized gain matrix analogous to the Kalman gain (Section 6.2). Since this parameterized gain matrix depends on the current filtering distribution, we employ the set transformer neural network architecture to process such measure inputs (Section 6.3). In practical applications, our method operates on an ensemble of particles (viewed as an empirical measure) and incorporates learnable mechanisms analogous to inflation and localization techniques in EnKF-based methods, enhancing performance with smaller ensemble sizes (Section 6.4). Our experiments across multiple challenging systems (Lorenz '63, Lorenz '96, and Kuramoto-Sivashinsky) demonstrate that our proposed method consistently outperforms classical approaches including the local ensemble transform Kalman filter (LETKF), with relative improvements of 15–30% for most scenarios (Section 6.5).

Despite the promising results, our method has limitations and several directions for future improvement. Our current approach is limited by the loss function design,

which primarily addresses state estimation rather than general filtering problems. Future work will consider probabilistic loss functions that can be used to estimate the filtering distribution.

Additionally, we plan to extend our approach beyond EnKF to include other filtering schemes, such as the ensemble square root filter (ESRF), which is a deterministic version of the EnKF, or variational methods such as iterative EnKF (iEnKF). The flexibility of our approach stems from the proposed MNM-based learning architecture, which can optimize variables related to the empirical distributions represented by ensembles.

We currently formulate the transformer as a neural operator accepting probability measures as inputs. In future work, we aim to enhance both the architecture and training methodology to simultaneously handle measure and function valued inputs, enabling amortized neural operators that generalize across different dynamics and observation models. Establishing universal approximation results and statistical guarantees for this formulation will also be of interest to further strengthen the theoretical foundations of our approach.

In conclusion, our work advances the field of data assimilation by introducing a novel machine learning approach, which is efficient for deployment, requiring only one pretraining stage and lightweight fine-tuning. Beyond immediate applications in state estimation, our theoretical contribution of extending attention mechanisms to operate as measure-to-measure transformations opens new research directions across multiple disciplines.

BIBLIOGRAPHY

A.S. Stordal, H.A. Karlsen, G. Nævdal, H.J. Skaug, and B. Vallés (2011). "Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter." In: *Comput. Geosci.* 15, pp. 293–305.

Abarbanel, Henry (2013). *Predicting the future: Completing models of observed complex systems*. Springer.

Abdulle, Assyr, E Weinan, Björn Engquist, and Eric Vanden-Eijnden (2012). "The heterogeneous multiscale method." In: *Acta Numerica* 21, pp. 1–87.

Adrian, Melissa, Daniel Sanz-Alonso, and Rebecca Willett (2025). "Data Assimilation with Machine Learning Surrogate Models: A Case Study with FourCastNet." In: *Artificial Intelligence for the Earth Systems* 4.3.

Agapiou, Sergios, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and AM Stuart (2017). "Importance sampling: Intrinsic dimension and computational cost." In: *Statistical Science*, pp. 405–431.

Alexander, Romeo and Dimitrios Giannakis (2020). "Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques." In: *Physica D: Nonlinear Phenomena* 409, p. 132520.

Alexe, Mihai, Eulalie Boucher, Peter Lean, Ewan Pinnington, Patrick Laloyaux, Anthony McNally, Simon Lang, Matthew Chantry, Chris Burrows, Marcin Chrust, Florian Pinault, Ethel Villeneuve, Niels Bormann, and Sean Healy (2024). "GraphDOP: Towards skilful data-driven medium-range weather forecasts learnt and initialised directly from observations." In: *arXiv preprint arXiv:2412.15687*.

Allen, Anna, Stratis Markou, Will Tebbutt, James Requeima, Wessel P. Bruinsma, Tom R. Andersson, Michael Herzog, Nicholas D. Lane, Matthew Chantry, J. Scott Hosking, and Richard E. Turner (May 2025). "End-to-end data-driven weather prediction." In: *Nature* 641.8065, pp. 1172–1179.

Anderson, Brian DO and John B Moore (2012). *Optimal filtering*. Courier Corporation.

Anderson, Jeffrey L (2001). "An ensemble adjustment Kalman filter for data assimilation." In: *Monthly weather review* 129.12, pp. 2884–2903.

Anderson, Jeffrey L. (2007). "An adaptive covariance inflation error correction algorithm for ensemble filters." In: *Tellus A: Dynamic Meteorology and Oceanography* 59.2, pp. 210–224. DOI: 10.1111/j.1600-0870.2006.00216.x. eprint: https://doi.org/10.1111/j.1600-0870.2006.00216.x.

Anderson, Jeffrey L. and Stephen L. Anderson (1999). "A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts." In: *Monthly Weather Review* 127.12, pp. 2741–2758. DOI: 10.1175/1520-0493(1999)127<2741:AMCIOT>2.0.CO;2.

Asch, Mark, Marc Bocquet, and Maëlle Nodet (2016). *Data assimilation: Methods, algorithms, and applications*. SIAM.

Ashwin, Peter (2003). "Synchronization from chaos." In: *Nature* 422.6930, pp. 384–385.

Åström, Karl Johan and Richard M Murray (2021). *Feedback systems: An introduction for scientists and engineers*. Princeton University Press.

B.R. Hunt, E.J. Kostelich, and I. Szunyogh (2007). "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter." In: *Physica D* 230, pp. 112–126.

Bach, Eviatar, Ricardo Baptista, Edoardo Calvello, Bohan Chen, and Andrew Stuart (2026). "Learning enhanced ensemble filters." In: *Journal of Computational Physics* 547, p. 114550. DOI: https://doi.org/10.1016/j.jcp.2025.114550.

Bach, Eviatar, Ricardo Baptista, Enoch Luk, and Andrew Stuart (Mar. 2025). *Learning Optimal Filters Using Variational Inference*. DOI: 10.48550/arXiv.2406.18066. arXiv: 2406.18066 [cs]. (Visited on 03/26/2025).

Bach, Eviatar, Ricardo Baptista, Daniel Sanz-Alonso, and Andrew Stuart (2025). *Machine Learning for Inverse Problems and Data Assimilation*. arXiv: 2410.10523 [stat.ML].

Bach, Eviatar and Michael Ghil (2023). "A multi-model ensemble Kalman filter for data assimilation and forecasting." In: *J. Adv. Model. Earth Syst.* 15.1, e2022MS003123.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate." In: *arXiv preprint* arXiv:2106.01506.

Bain, Alan and Dan Crisan (2009). *Fundamentals of stochastic filtering*. Vol. 3. Springer.

Bergemann, Kay and Sebastian Reich (2010a). "A localization technique for ensemble Kalman filters." In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 136.648, pp. 701–707.

Bergemann, Kay and Sebastian Reich (2010b). "A mollified ensemble Kalman filter." In: *Quarterly Journal of the Royal Meteorological Society* 136.651, pp. 1636–1643.

Bergemann, Kay and Sebastian Reich (2012). "An ensemble Kalman-Bucy filter for continuous data assimilation." In: *Meteorologische Zeitschrift* 21.3, p. 213.

Bi, Kaifeng, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian (July 2023). "Accurate medium-range global weather forecasting with 3D neural networks." In: *Nature* 619, pp. 1–6.

Bickel, Peter, Bo Li, and Thomas Bengtsson (2008). "Sharp failure rates for the bootstrap particle filter in high dimensions." In: *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*. Institute of Mathematical Statistics, pp. 318–329.

Bishop, Adrian N. and Pierre Del Moral (2023). "On the mathematical theory of ensemble (linear-Gaussian) Kalman-Bucy filtering." In: *Math. Control Signals Systems* 35.4, pp. 835–903. ISSN: 0932-4194,1435-568X. DOI: 10.1007/s00498-023-00357-2. URL: https://doi.org/10.1007/s00498-023-00357-2.

Bishop, Ch.M. (2011). *Pattern Recognition and Machine Learning*. 2nd. New York: Springer-Verlag.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York. DOI: 10.1007/978-0-387-45528-0. URL: https://doi-org.extranet.enpc.fr/10.1007/978-0-387-45528-0.

Bishop, Craig H, Brian J Etherton, and Sharanya J Majumdar (2001). "Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects." In: *Monthly weather review* 129.3, pp. 420–436.

Bocquet, M. and P. Sakov (2014). "An iterative ensemble Kalman smoother." In: *Q. J. R. Meteorol. Soc.* 140, pp. 1521–1535. DOI: 10.1002/qj.2236.

Bocquet, Marc, Alban Farchi, Tobias S. Finn, Charlotte Durand, Sibo Cheng, Yumeng Chen, Ivo Pasmans, and Alberto Carrassi (Sept. 2024). "Accurate deep learning-based filtering for chaotic dynamics by identifying instabilities without an ensemble." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34.9, p. 091104. ISSN: 1054-1500. DOI: 10.1063/5.0230837. (Visited on 10/06/2024).

Bocquet, Marc, Karthik S Gurumoorthy, Amit Apte, Alberto Carrassi, Colin Grudzien, and Christopher KRT Jones (2017). "Degenerate Kalman filter error covariances and their convergence onto the unstable subspace." In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1, pp. 304–333.

Bocquet, Marc and Pavel Sakov (2012). "Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems." In: *Nonlinear Processes in Geophysics* 19.3, pp. 383–399.

Boudier, Pierre, Anthony Fillion, Serge Gratton, Selime Gürol, and Sixin Zhang (2023). "Data Assimilation Networks." en. In: *Journal of Advances in Modeling Earth Systems* 15.4, e2022MS003353. ISSN: 1942-2466. DOI: 10.1029/2022MS003353. (Visited on 07/17/2023).

Bröcker, Jochen, David Engster, and Ulrich Parlitz (Dec. 2009). "Probabilistic evaluation of time series models: A comparison of several approaches." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19.4, p. 043130. ISSN: 1054-1500. DOI: 10.1063/1.3271343. (Visited on 09/26/2024).

Burgers, Gerrit, Peter Jan Van Leeuwen, and Geir Evensen (1998). "Analysis scheme in the ensemble Kalman filter." In: *Monthly weather review* 126.6, pp. 1719–1724.

Burov, Dmitry, Dimitrios Giannakis, Krithika Manohar, and Andrew Stuart (2021). "Kernel analog forecasting: Multiscale test problems." In: *Multiscale Modeling & Simulation* 19.2, pp. 1011–1040.

C.H. Bishop, B.J. Etherton, and S.J. Majumdar (2001). "Adaptive sampling with ensemble transform Kalman filter. Part I: Theoretical aspects." In: *Monthly Weather Review* 129, pp. 420–436.

Calvello, Edoardo, Nikola B. Kovachki, Matthew E. Levine, and Andrew M. Stuart (2025). "Continuum Attention for Neural Operators." In: *Journal of Machine Learning Research* 26.300, pp. 1–52. URL: https://jmlr.org/papers/volume26/24-0879/24-0879.pdf.

Calvello, Edoardo, Pierre Monmarché, Anadrew M. Stuart, and Urbain Vaes (2026). "Accuracy of the Ensemble Kalman Filter in the Near-Linear Setting." In: *SIAM Journal on Numerical Analysis* 64.2, pp. 391–429. DOI: https://doi.org/10.1017/S0962492924000060.

Calvello, Edoardo, Sebastian Reich, and Andrew M. Stuart (2025). "Ensemble Kalman Methods: A Mean Field Perspective." In: *Acta Numerica* 34, pp. 123–291. DOI: https://doi.org/10.1017/S0962492924000060.

Cao, Shuhao (2021). "Choose a Transformer: Fourier or Galerkin." In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 24924–24940.

Carrillo, J. A., F. Hoffmann, A. M. Stuart, and U. Vaes (2024). "The Mean-Field Ensemble Kalman Filter: Near-Gaussian Setting." In: *SIAM Journal on Numerical Analysis* 62.6, pp. 2549–2587. DOI: 10.1137/24M1628207. URL: https://doi.org/10.1137/24M1628207.

Carrillo, José A, Young-Pil Choi, Claudia Totzeck, and Oliver Tse (2018). "An analytical framework for consensus-based global optimization method." In: *Mathematical Models and Methods in Applied Sciences* 28.06, pp. 1037–1066.

Castin, Valérie, Pierre Ablin, José Antonio Carrillo, and Gabriel Peyré (2025). "A Unified Perspective on the Dynamics of Deep Transformers." In: *arXiv preprint arXiv:2501.18322*.

Chandler, Gary J. and Rich R. Kerswell (2013). "Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow." In: *Journal of Fluid Mechanics* 722, pp. 554–595. DOI: 10.1017/jfm.2013.122.

Chattopadhyay, A., M. Mustafa, P. Hassanzadeh, E. Bach, and K. Kashinath (2022). "Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based U-NET in a case study with ERA5." In: *Geoscientific Model Development* 15.5, pp. 2221–2237.

Chen, Long, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua (2017). "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5659–5667.

Chen, Yuming, Daniel Sanz-Alonso, and Rebecca Willett (2022). "Autodifferentiable Ensemble Kalman Filters." In: *SIAM Journal on Mathematics of Data Science* 4.2, pp. 801–833.

Cheng, Sibo, César Quilodrán-Casas, Said Ouala, Alban Farchi, Che Liu, Pierre Tandeo, Ronan Fablet, Didier Lucor, Bertrand Iooss, Julien Brajard, Dunhui Xiao, Tijana Janjic, Weiping Ding, Yike Guo, Alberto Carrassi, Marc Bocquet, and Rossella Arcucci (2023a). "Machine Learning With Data Assimilation and Uncertainty Quantification for Dynamical Systems: A Review." In: *IEEE/CAA Journal of Automatica Sinica* 10.6, pp. 1361–1387.

Cheng, Sibo, César Quilodrán-Casas, Said Ouala, Alban Farchi, Che Liu, Pierre Tandeo, Ronan Fablet, Didier Lucor, Bertrand Iooss, Julien Brajard, Dunhui Xiao, Tijana Janjic, Weiping Ding, Yike Guo, Alberto Carrassi, Marc Bocquet, and Rossella Arcucci (June 2023b). "Machine Learning With Data Assimilation and Uncertainty Quantification for Dynamical Systems: A Review." In: *IEEE/CAA Journal of Automatica Sinica* 10.6, pp. 1361–1387. ISSN: 2329-9274. DOI: 10.1109/JAS.2023.123537.

Chernov, Alexey, Håkon Hoel, Kody JH Law, Fabio Nobile, and Raul Tempone (2021). "Multilevel ensemble Kalman filtering for spatio-temporal processes." In: *Numerische Mathematik* 147.1, pp. 71–125.

Chipilski, Hristo G (2025). "Exact nonlinear state estimation." In: *Journal of the Atmospheric Sciences* 82.4, pp. 809–827. DOI: 10.1175/JAS-D-24-0171.1. URL: https://journals.ametsoc.org/view/journals/atsc/82/4/JAS-D-24-0171.1.xml.

Chopin, N. and O. Papaspiliopoulos (2020). *An Introduction to Sequential Monte Carlo*. Cham, Switzerland: Springer Nature Switzerland AG.

Chung, Junyoung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." In: *Neural Information Processing Systems 2014 Workshop on Deep Learning, December 2014*.

Corenflos, Adrien, James Thornton, George Deligiannidis, and Arnaud Doucet (2021). "Differentiable particle filtering via entropy-regularized optimal transport." In: *International Conference on Machine Learning*. PMLR, pp. 2100–2111.

Cotter, C. and Sebastian Reich (2013). "Ensemble filter techniques for intermittent data assimilation." In: *Radon Ser. Comput. Appl. Math.* 13, pp. 91–134. DOI: 10.1515/9783110282269.91.

Cotter, Simon L, Gareth O Roberts, Andrew M Stuart, and David White (2013). "MCMC methods for functions: modifying old algorithms to make them faster." In: *Statistical Science*, pp. 424–446.

Crisan, Dan, Pierre Del Moral, and Terry Lyons (1999). "Discrete filtering using branching and interacting particle systems." In: *Markov Processes and Related Fields* 5, pp. 293–318.

Crisan, Dan and Jie Xiong (2010). "Approximate McKean-Vlasov representations for a class of SPDEs." In: *Stochastics* 82.1-3, pp. 53–68. ISSN: 1744-2508. DOI: 10.1080/17442500902723575.

Cuturi, Marco (2013). "Sinkhorn distances: Lightspeed computation of optimal transport." In: *Advances in neural information processing systems* 26.

Daum, Fred and Jim Huang (2011). "Particle flow for nonlinear filters." In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5920–5923.

Daum, Fred, Jim Huang, and Arjang Noushin (2010). "Exact particle flow for nonlinear filters." In: *Signal Processing, Sensor Fusion, and Target Recognition XIX*. Ed. by Ivan Kadar. Vol. 7697. International Society for Optics and Photonics. SPIE, pp. 92–110. DOI: 10.1117/12.839590.

Del Moral, P. (2004). *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer-Verlag.

Del Moral, P. and J. Tugaut (2018). "On the stability and the uniform propagation of chaos properties of ensemble Kalman-Bucy filters." In: *Ann. Appl. Probab.* 28.2, pp. 790–850. ISSN: 1050-5164. DOI: 10.1214/17-AAP1317. URL: https://doi-org.extranet.enpc.fr/10.1214/17-AAP1317.

Del Moral, P. and J. Tugaut (2018). "ON THE STABILITY AND THE UNIFORM PROPAGATION OF CHAOS PROPERTIES OF ENSEMBLE KALMAN–BUCY FILTERS." In: *The Annals of Applied Probability* 28, pp. 790–850.

Del Moral, Pierre (1997). "Nonlinear filtering: interacting particle resolution." In: *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 325.6, pp. 653–658.

Del Moral, Pierre, Arnaud Doucet, and Ajay Jasra (2006). "Sequential Monte Carlo samplers." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3, pp. 411–436.

Del Moral, Pierre and Alice Guionnet (2001). "On the stability of interacting processes with applications to filtering and genetic algorithms." In: *Annales de l'Institut Henri Poincaré (B) Probability and Statistics* 37.2, pp. 155–194.

Del Moral, Pierre and Emma Horton (2023a). "A theoretical analysis of one-dimensional discrete generation ensemble Kalman particle filters." In: *The Annals of Applied Probability* 33.2, pp. 1327–1372.

Del Moral, Pierre and Emma Horton (2023b). "A theoretical analysis of one-dimensional discrete generation ensemble Kalman particle filters." In: *Ann. Appl. Probab.* 33.2, pp. 1127–1172. ISSN: 1050-5164,2168-8737. DOI: 10.1214/22-aap1843. URL: https://doi.org/10.1214/22-aap1843.

Ding, Zhiyan and Qin Li (2021a). "Ensemble Kalman inversion: Mean-field limit and convergence analysis." In: *Statistics and Computing* 31.1, pp. 1–21.

Ding, Zhiyan and Qin Li (2021b). "Ensemble Kalman sampler: Mean-field limit and convergence analysis." In: *SIAM Journal on Mathematical Analysis* 53.2, pp. 1546–1578.

Ding, Zhiyan and Qin Li (2021c). "Ensemble Kalman sampler: mean-field limit and convergence analysis." In: *SIAM J. Math. Anal.* 53.2, pp. 1546–1578. ISSN: 0036-1410,1095-7154. DOI: 10.1137/20M1339507. URL: https://doi.org/10.1137/20M1339507.

Ding, Zhiyan, Qin Li, and Jianfeng Lu (2021). "Ensemble Kalman Inversion for nonlinear problems: Weights, consistency, and variance bounds." In: *Foundations of Data Science* 3.3, pp. 371–411.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Doucet, Arnaud, Nando De Freitas, and Neil Gordon (2001). "An introduction to sequential Monte Carlo methods." In: *Sequential Monte Carlo Methods in Practice*. Springer, pp. 3–14.

Doucet, Arnaud, Nando de Freitas, and Neil Gordon, eds. (2001). *Sequential Monte Carlo methods in practice*. Statistics for Engineering and Information Science. Springer-Verlag, New York. ISBN: 0-387-95146-6. DOI: 10.1007/978-1-4757-3437-9. URL: https://doi-org.extranet.enpc.fr/10.1007/978-1-4757-3437-9.

Doucet, Arnaud, Simon Godsill, and Christophe Andrieu (2000). "On sequential Monte Carlo sampling methods for Bayesian filtering." In: *Statistics and Computing* 10.3, pp. 197–208.

Eaton, Morris L (2007). "Multivariate Statistics: A Vector Space Approach." In: *Lecture Notes-Monograph Series* 53, pp. i–512. ISSN: 07492170. URL: http://www.jstor.org/stable/20461449 (visited on 08/13/2022).

El Moselhy, Tarek A. and Youssef M. Marzouk (2012). "Bayesian inference with optimal maps." In: *Journal of Computational Physics* 231, pp. 7815–7850. DOI: https://doi.org/10.1016/j.jcp.2012.07.022.

Evensen, Geir (1994). "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics." In: *Journal of Geophysical Research: Oceans* 99.C5, pp. 10143–10162.

Evensen, Geir (2003). "The ensemble Kalman filter: Theoretical formulation and practical implementation." In: *Ocean dynamics* 53, pp. 343–367.

Evensen, Geir (2019). "Accounting for model errors in iterative ensemble smoothers." In: *Computational Geosciences* 23.4, pp. 761–775.

Evensen, Geir, Femke C. Vossepoel, and Peter Jan van Leeuwen (2022). *Data assimilation fundamentals: a unified formulation of the state and parameter estimation problem*. Springer Textbooks in Earth Sciences, Geography and Environment. Springer, Cham. DOI: 10.1007/978-3-030-96709-3. URL: https://doi-org.extranet.enpc.fr/10.1007/978-3-030-96709-3.

Farchi, Alban, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet (2021). "Using machine learning to correct model error in data assimilation and forecast applications." In: *Quarterly Journal of the Royal Meteorological Society* 147.739, pp. 3067–3084.

Fatkullin, Ibrahim and Eric Vanden-Eijnden (2004). "A computational strategy for multiscale systems with applications to Lorenz 96 model." In: *Journal of Computational Physics* 200.2, pp. 605–638.

Fertig, Elana J, John Harlim, and Brian R Hunt (2007). "A comparative study of 4D-VAR and a 4D ensemble Kalman filter: Perfect model simulations with Lorenz-96." In: *Tellus A: Dynamic Meteorology and Oceanography* 59.1, pp. 96–100.

Foias, Ciprian, Cecilia F Mondaini, and Edriss S Titi (2016). "A Discrete Data Assimilation Scheme for the Solutions of the Two-Dimensional Navier–Stokes Equations and Their Statistics." In: *SIAM Journal on Applied Dynamical Systems* 15.4, pp. 2109–2142.

Foias, ClPRIAN and Giovanni Prodi (1967). "Sur le comportement global des solutions non-stationnaires des équations de Navier-Stokes en dimension 2." In: *Rendiconti del Seminario matematico della Università di Padova* 39, pp. 1–34.

Frei, Marco and Hans R Künsch (2013). "Bridging the ensemble Kalman and particle filters." In: *Biometrika* 100.4, pp. 781–800.

G. Burgers, P.J. van Leeuwen, and G. Evensen (1998). "Analysis scheme in the ensemble Kalman filter." In: *Monthly Weather Review* 126, pp. 1719–1724.

G. Evensen, F.C. Vossepoel, and P.J. van Leeuwen (2022). *Data Assimilation Fundamentals: A unified Formulation of the State and Parameter Estimation Problem*. Cham, Switzerland: Springer Nature Switzerland AG.

Gaspari, Gregory and Stephen E. Cohn (1999). "Construction of correlation functions in two and three dimensions." In: *Quarterly Journal of the Royal Meteorological Society* 125.554, pp. 723–757. DOI: https://doi.org/10.1002/qj.49712555417. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49712555417.

Gerber, Nicolai Jurek, Franca Hoffmann, and Urbain Vaes (2023). "Mean-field limits for Consensus-Based Optimization and Sampling." In: *arXiv preprint* 2312.07373.

Geshkovski, Borjan, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet (2023). "The emergence of clusters in self-attention dynamics." In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Geshkovski, Borjan, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet (2024). "A mathematical perspective on Transformers." In: *arXiv preprint arXiv:22312.10794*.

Geshkovski, Borjan, Philippe Rigollet, and Domènec Ruiz-Balet (2024). "Measure-to-measure interpolation using Transformers." In: *arXiv preprint arXiv:2411.04551*. eprint: 2411.04551.

Ghil, M., S. Cohn, J. Tavantzis, K. Bube, and E. Isaacson (1981). "Applications of Estimation Theory to Numerical Weather Prediction." In: *Dynamic Meteorology: Data Assimilation Methods*. Ed. by Lennart Bengtsson, Michael Ghil, and Erland Källén. New York, NY: Springer New York, pp. 139–224. ISBN: 978-1-4612-5970-1. DOI: 10.1007/978-1-4612-5970-1_5. URL: https://doi.org/10.1007/978-1-4612-5970-1_5.

Ghil, Michael, S Cohn, John Tavantzis, K Bube, and Eugene Isaacson (1981). "Applications of estimation theory to numerical weather prediction." In: *Dynamic meteorology: Data assimilation methods*. Springer, pp. 139–224.

Goldstein, Michael and David Wooff (2007). *Bayes linear statistics: Theory and methods*. Vol. 716. John Wiley & Sons.

González-Tokman, Cecilia and Brian R Hunt (2013). "Ensemble data assimilation for hyperbolic systems." In: *Physica D: Nonlinear Phenomena* 243.1, pp. 128–142.

Goodman, Jonathan and Jonathan Weare (2010). "Ensemble samplers with affine invariance." In: *Communications in applied mathematics and computational science* 5.1, pp. 65–80.

Gottwald, Georg A and Andrew J Majda (2013). "A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks." In: *Nonlinear Processes in Geophysics* 20.5, pp. 705–712.

Gottwald, Georg A and Sebastian Reich (2021). "Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation." In: *Physica D: Nonlinear Phenomena* 423, p. 132911.

Grooms, Ian (2021). "Analog ensemble data assimilation and a method for constructing analogs with variational autoencoders." In: *Quarterly Journal of the Royal Meteorological Society* 147.734, pp. 139–149.

Grooms, Ian, Yoonsang Lee, and Andrew J Majda (2014). "Ensemble Kalman filters for dynamical systems with unresolved turbulence." In: *Journal of Computational Physics* 273, pp. 435–452.

Grooms, Ian, Yoonsang Lee, and Andrew J Majda (2015). "Ensemble filtering and low-resolution model error: Covariance inflation, stochastic parameterization, and model numerics." In: *Monthly Weather Review* 143.10, pp. 3912–3924.

Gu, Yaqing and Dean S Oliver (2007). "An iterative ensemble Kalman filter for multiphase fluid flow data assimilation." In: *SPE Journal* 12.04, pp. 438–446.

Guibas, John, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro (2022). "Efficient Token Mixing for Transformers via Adaptive Fourier Neural Operators." In: *International Conference on Learning Representations*.

Gupta, Aayush, Akshay Subramaniam, Michael S Pritchard, Karthik Kashinath, Sergey Frolov, Kelsey Lieberman, Christopher Miller, Nicholas Silverman, and Noah D Brenowitz (2026). "HealDA: Highlighting the importance of initial errors in end-to-end AI weather forecasts." In: *arXiv preprint arXiv:2601.17636*.

Gurumoorthy, Karthik S, Colin Grudzien, Amit Apte, Alberto Carrassi, and Christopher KRT Jones (2017). "Rank deficiency of Kalman error covariance matrices in linear time-varying system with deterministic evolution." In: *SIAM Journal on Control and Optimization* 55.2, pp. 741–759.

Guth, Philipp A., Claudia Schillings, and Simon Weissmann (2022). "Ensemble Kalman filter for neural network-based one-shot inversion." In: *Optimization and Control for Partial Differential Equations*. Ed. by Roland Herzog, Matthias Heinkenschloss, Dante Kalise, Georg Stadler, and Emmanuel Trélat. De Gruyter, pp. 393–418.

Haber, Eldad, Felix Lucka, and Lars Ruthotto (2018). "Never look back - The EnKF method and its application to the training of neural networks without back propagation." In: *arXiv preprint arXiv:1805.08034*.

Hairer, Martin, Andrew M Stuart, and Sebastian J Vollmer (2014). "Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions." In: *The Annals of Applied Probability* 24.6, pp. 2455–2490.

Hao, Zhongkai, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu (2023). "GNOT: A General Neural Operator Transformer for Operator Learning." In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org.

Harlim, John and Brian R Hunt (2007a). "A non-Gaussian ensemble filter for assimilating infrequent noisy observations." In: *Tellus A: Dynamic Meteorology and Oceanography* 59.2, pp. 225–237.

Harlim, John and Brian R Hunt (2007b). "Four-dimensional local ensemble transform Kalman filter: Numerical experiments with a global circulation model." In: *Tellus A: Dynamic Meteorology and Oceanography* 59.5, pp. 731–748.

Harlim, John, Adam Mahdi, and Andrew J Majda (2014). "An ensemble Kalman filter for statistical estimation of physics constrained nonlinear regression models." In: *Journal of Computational Physics* 257, pp. 782–812.

Harlim, John and Andrew J Majda (2010). "Filtering turbulent sparsely observed geophysical flows." In: *Monthly Weather Review* 138.4, pp. 1050–1083.

Hayden, Kevin, Eric Olson, and Edriss S Titi (2011). "Discrete data assimilation in the Lorenz and 2D Navier–Stokes equations." In: *Physica D: Nonlinear Phenomena* 240.18, pp. 1416–1425.

He, Juncai, Xinliang Liu, and Jinchao Xu (2024). "MgNO: Efficient Parameterization of Linear Operators via Multigrid." In: *The Twelfth International Conference on Learning Representations*.

Hermann, Robert and Arthur Krener (1977). "Nonlinear controllability and observability." In: *IEEE Transactions on automatic control* 22.5, pp. 728–740.

Ho, YC and RCKA Lee (1964). "A Bayesian approach to problems in stochastic estimation and control." In: *IEEE transactions on automatic control* 9.4, pp. 333–339.

Hoang, H.S., P. De Mey, and O. Talagrand (Dec. 1994). "A simple adaptive algorithm of stochastic approximation type for system parameter and state estimation." In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. Vol. 1, 747–752 vol.1. DOI: 10.1109/CDC.1994.410863. (Visited on 10/24/2023).

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory." In: *Neural Computation* 9.8, pp. 1735–1780.

Hoel, Håkon, Kody JH Law, and Raul Tempone (2016). "Multilevel ensemble Kalman filtering." In: *SIAM Journal on Numerical Analysis* 54.3, pp. 1813–1839.

Höhlein, Kevin, Benedikt Schulz, Rüdiger Westermann, and Sebastian Lerch (Jan. 2024). "Postprocessing of Ensemble Weather Forecasts Using Permutation-invariant Neural Networks." In: *Artificial Intelligence for the Earth Systems* 3.1, e230070. ISSN: 2769-7525. DOI: 10.1175/AIES-D-23-0070.1. (Visited on 09/17/2024).

Holland, Mark and Ian Melbourne (2007). "Central limit theorems and invariance principles for Lorenz attractors." In: *Journal of the London Mathematical Society* 76.2, pp. 345–364.

Hoop, Maarten V. de, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart (2022). "The Cost-Accuracy Trade-Off in Operator Learning with Neural Networks." In: *Journal of Machine Learning* 1.3, pp. 299–341. ISSN: 2790-2048.

Houtekamer, P.L. and H.L. Mitchell (2005). "Ensemble Kalman filtering." In: *Q. J. Royal Meteorological Soc.* 131, pp. 3269–3289.

Hu, Chih-Chi, Peter Jan Van Leeuwen, and Jeffrey L Anderson (2024). "An implementation of the particle flow filter in an atmospheric model." In: *Monthly Weather Review* 152.10, pp. 2247–2264.

Hunt, Brian R., Eric J. Kostelich, and Istvan Szunyogh (2007). "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter." In: *Physica D: Nonlinear Phenomena* 230.1, pp. 112–126.

J. Lei and P. Bickel (2011). "A Moment Matching Ensemble Filter for Nonlinear Non-Gaussian Data Assimilation." In: *Monthly Weather Review* 139, pp. 3964–3973.

J. Pidstrigach and S. Reich (2023). "Affine-invariant ensemble transform methods for logistic regression." In: *Foundations of Computational Mathematics* 23.2, pp. 675–708.

J. Tödter and B. Ahrens (2015). "A second-order exact ensemble square root filter for nonlinear data assimilation." In: *Mon. Wea. Rev.* 143, pp. 1347–1367.

J.S. Whitaker and T.M. Hamill (2002). "Ensemble data assimilation without perturbed observations." In: *Monthly Weather Review* 130, pp. 1913–1924.

Al-Jarrah, Mohammad, Niyizhen Jin, Bamdad Hosseini, and Amirhossein Taghvaei (2023). "Nonlinear filtering with brenier optimal transport maps." In: *arXiv preprint arXiv:2310.13886*.

Jazwinski, Andrew H (2007). *Stochastic processes and filtering theory*. Courier Corporation.

Jazwinski, Andrew H. (1970). *Stochastic Processes and Filtering Theory*. Vol. 64. Mathematics in Science and Engineering. New York: Academic Press. ISBN: 0123815509.

Julier, Simon, Jeffrey Uhlmann, and Hugh F Durrant-Whyte (2000). "A new method for the nonlinear transformation of means and covariances in filters and estimators." In: *IEEE Transactions on Automatic Control* 45.3, pp. 477–482.

Kaipio, Jari and Erkki Somersalo (2006). *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media.

Kalman, R. E. (Mar. 1960). "A New Approach to Linear Filtering and Prediction Problems." In: *Journal of Basic Engineering* 82.1, pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. URL: https://doi.org/10.1115/1.3662552.

Kalman, R. E. and R. S. Bucy (Mar. 1961). "New Results in Linear Filtering and Prediction Theory." In: *Journal of Basic Engineering* 83.1, pp. 95–108. ISSN: 0021-9223. DOI: 10.1115/1.3658902. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/83/1/95/5503549/95_1.pdf. URL: https://doi.org/10.1115/1.3658902.

Kalnay, Eugenia (Nov. 2002). *Atmospheric Modeling, Data Assimilation and Predictability*. Vol. 129.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei (2020). "Scaling Laws for Neural Language Models." In: *arXiv preprint* arXiv:2001.08361.

Kassam, Aly-Khan and Lloyd N Trefethen (2005). "Fourth-order time-stepping for stiff PDEs." In: *SIAM Journal on Scientific Computing* 26.4, pp. 1214–1233.

Kelly, David, Andrew J Majda, and Xin T Tong (2015). "Concrete ensemble Kalman filters with rigorous catastrophic filter divergence." In: *Proceedings of the National Academy of Sciences* 112.34, pp. 10589–10594.

Kelly, David TB, Kody JH Law, and Andrew M Stuart (2014). "Well-posedness and accuracy of the ensemble Kalman filter in discrete and continuous time." In: *Nonlinearity* 27.10, p. 2579.

Kitanidis, Peter K. (1995). "Quasi-linear geostatistical theory for inversion." In: *Water Resour. Res.* 31.10, pp. 2411–2419.

Kossaifi, Jean, Nikola Kovachki, Morteza Mardani, Daniel Leibovici, Suman Ravuri, Ira Shokar, Edoardo Calvello, Mohammad Shoaib Abbas, Peter Harrington, Ashay Subramaniam, Noah Brenowitz, Boris Bonev, Wonmin Byeon, Karsten Kreis, Dale Durran, Arash Vahdat, Mike Pritchard, and Jan Kautz (2026). "Demystifying Data-Driven Probabilistic Medium-Range Weather Forecasting." In: *arXiv preprint arXiv:2601.18111*.

Kovachki, Nikola, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2023a). "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs." In: *Journal of Machine Learning Research* 24.89, pp. 1–97.

Kovachki, Nikola, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2023b). "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs." In: *Journal of Machine Learning Research* 24.89, pp. 1–97. ISSN: 1533-7928. (Visited on 03/21/2025).

Kovachki, Nikola B, Samuel Lanthaler, and Andrew M Stuart (2024). "Operator learning: Algorithms and analysis." In: *arXiv preprint arXiv:2402.15715*.

Kovachki, Nikola B and Andrew M Stuart (2019). "Ensemble Kalman inversion: A derivative-free technique for machine learning tasks." In: *Inverse Problems* 35.9, p. 095005.

Kuramoto, Yoshiki (1978). "Diffusion-induced chaos in reaction systems." In: *Progress of Theoretical Physics Supplement* 64, pp. 346–367.

Kurth, Thorsten, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar (2023). "FourCastNet: Accelerating Global High-Resolution Weather Forecasting Using Adaptive Fourier Neural Operators." In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '23. Davos, Switzerland: Association for Computing Machinery.

Kuznetsov, Leonid, Kayo Ide, and Christopher KRT Jones (2003). "A method for assimilation of Lagrangian data." In: *Monthly Weather Review* 131.10, pp. 2247–2260.

Kwiatkowski, E. and J. Mandel (2015). "Convergence of the Square Root Ensemble Kalman Filter in the Large Ensemble Limit." In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1, pp. 1–17. DOI: 10.1137/140965363.

Lange, Theresa and Wilhelm Stannat (2021). "On the continuous time limit of the ensemble Kalman filter." In: *Mathematics of Computation* 90.327, pp. 233–265.

Lanthaler, Samuel, Zongyi Li, and Andrew M. Stuart (Aug. 2025). "Nonlocality and Nonlinearity Implies Universality in Operator Learning." In: *Constructive Approximation*. ISSN: 1432-0940. DOI: 10.1007/s00365-025-09718-3. URL: https://doi.org/10.1007/s00365-025-09718-3.

Law, KJH, D Sanz-Alonso, Abhishek Shukla, and AM Stuart (2016). "Filter accuracy for the Lorenz 96 model: Fixed versus adaptive observation operators." In: *Physica D: Nonlinear Phenomena* 325, pp. 1–13.

Law, Kody, Andrew Stuart, and Konstantinos Zygalakis (2015). *Data Assimilation*. Vol. 62. Texts in Applied Mathematics. Springer, Cham. DOI: 10.1007/978-3-319-20325-6. URL: https://doi-org.extranet.enpc.fr/10.1007/978-3-319-20325-6.

Law, Kody, Andrew Stuart, and Kostas Zygalakis (2015). "Data assimilation." In: *Cham, Switzerland: Springer* 214.

Law, Kody JH, Abhishek Shukla, and Andrew M Stuart (2014). "Analysis of the 3DVAR filter for the partially observed Lorenz'63 model." In: *Discrete and Continuous Dynamical Systems* 34.3, pp. 1061–1078.

Law, Kody JH and Andrew M Stuart (2012). "Evaluating data assimilation algorithms." In: *Monthly Weather Review* 140.11, pp. 3757–3782.

Law, Kody JH, Hamidou Tembine, and Raul Tempone (2016). "Deterministic mean-field ensemble Kalman filtering." In: *SIAM Journal on Scientific Computing* 38.3, A1251–A1279.

Le Gland, F., V. Monbet, and V.-D. Tran (2011a). "Large sample asymptotics for the ensemble Kalman filter." In: *The Oxford Handbook of Nonlinear Filtering*. Oxford: Oxford Univ. Press, pp. 598–631.

Le Gland, F., V. Monbet, and V.-D. Tran (2011b). "Large sample asymptotics for the ensemble Kalman filter." In: *The Oxford handbook of nonlinear filtering*. Oxford Univ. Press, Oxford, pp. 598–631.

Lee, Juho, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh (2019). *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*. arXiv: 1810.00825 [cs.LG].

Lee, Yoonsang, Andrew J Majda, and Di Qi (2017). "Preventing catastrophic filter divergence using adaptive additive inflation for baroclinic turbulence." In: *Monthly Weather Review* 145.2, pp. 669–682.

Levine, Matthew and Andrew Stuart (July 2021). "A Framework for Machine Learning of Model Error in Dynamical Systems." In: *Communications of the American Mathematical Society* 2, pp. 283–344.

Levine, Matthew and Andrew Stuart (2022). "A framework for machine learning of model error in dynamical systems." en. In: *Communications of the American Mathematical Society* 2.07, pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10. (Visited on 03/17/2023).

Li, Gaoming and Albert Coburn Reynolds (2009). "An iterative ensemble Kalman filter for data assimilation." In: *SPE Journal* 14.03, pp. 496–505.

Li, Zhijie, Tianyuan Liu, Wenhui Peng, Yuan Zelong, and Jianchun Wang (June 2024). "A transformer-based neural operator for large-eddy simulation of turbulence." In: *Physics of Fluids* 36. DOI: 10.1063/5.0210493.

Li, Zijie, Kazem Meidani, and Amir Barati Farimani (2023). "Transformer for Partial Differential Equations' Operator Learning." In: *Transactions on Machine Learning Research*. ISSN: 2835-8856.

Li, Zongyi, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2021). "Fourier Neural Operator for Parametric Partial Differential Equations." In: *International Conference on Learning Representations*.

Liu, Z., Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo (Oct. 2021). "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows." In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 9992–10002.

Livings, D.M., S.L. Dance, and N.K. Nichols (2008). "Unbiased ensemble square root filters." In: *Physica D* 237, pp. 1021–1028.

Lorenz, Edward N (1963a). "Deterministic nonperiodic flow." In: *Journal of the Atmospheric Sciences* 20.2, pp. 130–141.

Lorenz, Edward N (1996a). "Predictability: A problem partly solved." In: *Proc. Seminar on Predictability*. Vol. 1. ECMWF.

Lorenz, Edward N (1996b). "Predictability: A problem partly solved." In: *Proc. Seminar on predictability*. Vol. 1. 1. Reading, pp. 1–18.

Lorenz, Edward N. (Mar. 1963b). "Deterministic Nonperiodic Flow." In: *Journal of the Atmospheric Sciences* 20.2, pp. 130–148. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

Lorenz, Edward N. (1969). "Atmospheric Predictability as Revealed by Naturally Occurring Analogues." In: *Journal of Atmospheric Sciences* 26.4, pp. 636–646.

Luenberger, David (1971). "An introduction to observers." In: *IEEE Transactions on Automatic Control* 16.6, pp. 596–602.

Luenberger, David G (1964). "Observing the state of a linear system." In: *IEEE Transactions on Military Electronics* 8.2, pp. 74–80.

Luo, Xihaier, Xiaoning Qian, and Byung-Jun Yoon (2024). "Hierarchical neural operator transformer with learnable frequency-aware loss prior for arbitrary-scale super-resolution." In: *Proceedings of the 41st International Conference on Machine Learning*. ICML'24. Vienna, Austria.

M.K. Tippett, J.L. Anderson, C.H. Bishop, T.M. Hamill, and J.S. Whitaker (2003). "Ensemble Square Root Filters." In: *Mon. Weather Rev.*, pp. 1485–1490. DOI: 10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2.

Majda, Andrew J and Xin T Tong (2018). "Performance of ensemble Kalman filters in large dimensions." In: *Communications on Pure and Applied Mathematics* 71.5, pp. 892–937.

Mallia-Parfitt, Noeleene and Jochen Bröcker (Oct. 2016). "Assessing the performance of data assimilation algorithms which employ linear error feedback." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.10, p. 103109. ISSN: 1054-1500. DOI: 10.1063/1.4965029. (Visited on 11/09/2023).

Mandel, J., L. Cobb, and J.D. Beezley (2011). "On the convergence of the ensemble Kalman filter." In: *Appl Math* 56, pp. 533–541. DOI: 10.1007/s10492-011-0031-2.

Mandel, Jan, Loren Cobb, and Jonathan D. Beezley (2011). "On the convergence of the ensemble Kalman filter." In: *Appl. Math.* 56.6, pp. 533–541. ISSN: 0862-7940,1572-9109. DOI: 10.1007/s10492-011-0031-2. URL: https://doi.org/10.1007/s10492-011-0031-2.

Martins, André, António Farinhas, Marcos Treviso, Vlad Niculae, Pedro Aguiar, and Mario Figueiredo (2020). "Sparse and Continuous Attention Mechanisms." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 20989–21001.

McCabe, Michael and Jed Brown (2021). "Learning to Assimilate in Chaotic Dynamical Systems." In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 12237–12250. (Visited on 10/16/2023).

Mezić, Igor (Oct. 2020). "Spectrum of the Koopman Operator, Spectral Expansions in Functional Spaces, and State-Space Geometry." In: *Journal of Nonlinear Science* 30.5, pp. 2091–2145.

Michelson, Daniel M and Gregory I Sivashinsky (1977). "Nonlinear analysis of hydrodynamic instability in laminar flames—II. Numerical experiments." In: *Acta astronautica* 4.11-12, pp. 1207–1221.

Moodey, Alexander JF, Amos S Lawless, Roland WE Potthast, and Peter Jan Van Leeuwen (2013). "Nonlinear error dynamics for cycled data assimilation methods." In: *Inverse Problems* 29.2, p. 025002.

Moreno, Alexander, Zhenke Wu, Supriya Nagesh, Walter Dempsey, and James M Rehg (2022). "Kernel Multimodal Continuous Attention." In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., pp. 18046–18059.

N. Chustagulprom, S. Reich, and M. Reinhardt (2016). "A hybrid ensemble transform filter for nonlinear and spatially extended dynamical systems." In: *SIAM/ASA J. Uncertainty Quantification* 4, pp. 592–608.

Nerger, Lars (2022). "Data assimilation for nonlinear systems with a hybrid nonlinear Kalman ensemble transform filter." In: *Quarterly Journal of the Royal Meteorological Society* 148.743, pp. 620–640. DOI: https://doi.org/10.1002/qj.4221.

Oliver, D.S., A.C. Reynolds, and N. Liu (2008). *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge: Cambridge University Press.

Oliver, Dean S., Luciane B. Cunha, and Albert C. Reynolds (1997). "Markov Chain Monte Carlo Methods for Conditioning a Permeability Field to Pressure Data." In: *Mathematical Geology* 29.1, pp. 61–91. DOI: 10.1007/BF02769620.

Olson, Eric and Edriss S Titi (2003). "Determining modes for continuous data assimilation in 2D turbulence." In: *Journal of statistical physics* 113.5, pp. 799–840.

Ovadia, Oded, Adar Kahana, Panos Stinis, Eli Turkel, Dan Givoli, and George Em Karniadakis (2024). "ViTO: Vision Transformer-Operator." In: *Computer Methods in Applied Mechanics and Engineering* 428, p. 117109. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2024.117109.

P. Sakov and P.R. Oke (2008). "A deterministic formulation of the ensemble Kalman filter: An alternative to ensemble square root filters." In: *Tellus A* 60, pp. 361–371.

P.L. Houtekamer and M.L. Mitchell (1998). "Data assimilation using an ensemble Kalman filter techniques." In: *Monthly Weather Review* 126, pp. 796–811.

Pandya, Dishant, Bhasha Vachharajani, and Rohit Srivastava (2022). "A review of data assimilation techniques: Applications in engineering and agriculture." In: *Materials Today: Proceedings* 62, pp. 7048–7052.

Pavliotis, Grigoris and Andrew Stuart (2008). *Multiscale methods: Averaging and homogenization*. Springer Science & Business Media.

Pecora, Louis M and Thomas L Carroll (1990). "Synchronization in chaotic systems." In: *Physical Review Letters* 64.8, p. 821.

Peyré, Gabriel and Marco Cuturi (2019). "Computational optimal transport: With applications to data science." In: *Foundations and Trends® in Machine Learning* 11.5-6, pp. 355–607.

PhysicsNeMo Contributors (Feb. 2023). *NVIDIA PhysicsNeMo: An open-source framework for physics-based deep learning in science and engineering*. https://github.com/NVIDIA/physicsnemo.

Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks." In: *Acta Numerica* 8, pp. 143–195.

Price, Ilan, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson (Jan. 2025a). "Probabilistic weather forecasting with machine learning." In: *Nature* 637.8044, pp. 84–90. ISSN: 1476-4687. DOI: 10.1038/s41586-024-08252-9. URL: https://doi.org/10.1038/s41586-024-08252-9.

Price, Ilan, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson (Jan. 2025b). "Probabilistic weather forecasting with machine learning." In: *Nature* 637.8044, pp. 84–90.

Pulido, Manual and Peter Jan van Leeuwen (2018). "Kernel embedding of maps for Bayesian inference: The variational mapping particle filter." In: *EGU general assembly conference abstracts*, p. 3750.

Raanes, Patrick N., Yumeng Chen, and Colin Grudzien (Feb. 2024). "DAPPER: Data Assimilation with Python: a Package for Experimental Research." en. In: *Journal of Open Source Software* 9.94, p. 5150. ISSN: 2475-9066. DOI: 10.21105/joss.05150. (Visited on 03/22/2025).

Rahman, Md Ashiqur, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A. Yeh, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar (2024). "Pretraining Codomain Attention Neural Operators for Solving Multiphysics PDEs." In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rebeschini, Patrick and Ramon Van Handel (2015). "Can local particle filters beat the curse of dimensionality?" In: *The Annals of Applied Probability* 25.5, pp. 2809–2866.

Reich, Sebastian (2011). "A dynamical systems framework for intermittent data assimilation." In: *BIT Numerical Mathematics* 51.1, pp. 235–249.

Reich, Sebastian (2019). "Data assimilation: The Schrödinger perspective." In: *Acta Numerica* 28, pp. 635–711.

Reich, Sebastian and Colin Cotter (2015). *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press.

Robinson, Gregor, Ian Grooms, and William Kleiber (2018). "Improving particle filter performance by smoothing observations." In: *Monthly Weather Review* 146.8, pp. 2433–2446.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Cham: Springer International Publishing, pp. 234–241. ISBN: 978-3-319-24574-4.

S. Reich (2013). "A nonparametric ensemble transform method for Bayesian inference." In: *SIAM J. Sci. Comput.* 35, A2013–A2024. DOI: 10.1137/130907367.

Sacher, William and Peter Bartello (Aug. 2008). "Sampling Errors in Ensemble Kalman Filtering. Part I: Theory." In: *Monthly Weather Review* 136.8, pp. 3035–3049. ISSN: 0027-0644. DOI: 10.1175/2007MWR2323.1. URL: https://journals.ametsoc.org/doi/10.1175/2007MWR2323.1 (visited on 11/29/2018).

Sakov, P., D. S. Oliver, and L. Bertino (2012). "An iterative EnKF for strongly nonlinear systems." In: *Mon. Wea. Rev.* 140, pp. 1988–2004. DOI: 10.1175/MWR-D-11-00176.1.

Sakov, Pavel and Peter R Oke (2008). "A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters." In: *Tellus A: Dynamic Meteorology and Oceanography* 60.2, pp. 361–371.

Sakov, Pavel, Dean S. Oliver, and Laurent Bertino (2012). "An Iterative EnKF for Strongly Nonlinear Systems." In: *Monthly Weather Review* 140.6, pp. 1988–2004.

Salman, H, L Kuznetsov, CKRT Jones, and K Ide (2006). "A method for assimilating Lagrangian data into a shallow-water-equation ocean model." In: *Monthly Weather Review* 134.4, pp. 1081–1101.

Sampson, Christian, Alberto Carrassi, Ali Aydoğdu, and Chris KRT Jones (2021). "Ensemble Kalman filter for nonconservative moving mesh solvers with a joint physics and mesh location update." In: *Quarterly Journal of the Royal Meteorological Society* 147.736, pp. 1539–1561.

Sanz-Alonso, Daniel, Andrew Stuart, and Armeen Taeb (2023a). *Inverse Problems and Data Assimilation*. London Mathematical Society Student Texts. Cambridge University Press.

Sanz-Alonso, Daniel, Andrew Stuart, and Armeen Taeb (2023b). *Inverse Problems and Data Assimilation*. Vol. 107. Cambridge University Press.

Sanz-Alonso, Daniel and Andrew M Stuart (2015). "Long-time asymptotics of the filtering distribution for partially observed chaotic dynamical systems." In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1, pp. 1200–1220.

Sanz-Alonso, Daniel and Nathan Waniorek (2025). "Long-Time Accuracy of Ensemble Kalman Filters for Chaotic Dynamical Systems and Machine-Learned Dynamical Systems." In: *SIAM Journal on Applied Dynamical Systems* 24.3, pp. 2246–2286.

Särkkä, Simo and Lennart Svensson (2023). *Bayesian Filtering and Smoothing*. Second. Cambridge University Press.

Sauer, Tim, James A Yorke, and Martin Casdagli (1991). "Embedology." In: *Journal of statistical Physics* 65.3, pp. 579–616.

Schmid, Peter J. (2010). "Dynamic mode decomposition of numerical and experimental data." In: *Journal of Fluid Mechanics* 656, pp. 5–28.

Shi, Yuyang, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet (2022). "Conditional simulation using diffusion Schrödinger bridges." In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 1792–1802.

Snyder, Chris, Thomas Bengtsson, Peter Bickel, and Jeff Anderson (2008). "Obstacles to high-dimensional particle filtering." In: *Monthly Weather Review* 136.12, pp. 4629–4640.

Sontag, Eduardo D (2013). *Mathematical control theory: Deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media.

Spantini, Alessio, Ricardo Baptista, and Youssef Marzouk (2022a). "Coupling Techniques for Nonlinear Ensemble Filtering." In: *SIAM Review* 64.4, pp. 921–953.

Spantini, Alessio, Ricardo Baptista, and Youssef Marzouk (2022b). "Coupling techniques for nonlinear ensemble filtering." In: *SIAM Review* 64.4, pp. 921–953.

Stoer, Josef, Roland Bulirsch, R Bartels, Walter Gautschi, and Christoph Witzgall (1980). *Introduction to numerical analysis*. Vol. 1993. Springer.

Stuart, Andrew M (2010a). "Inverse problems: A Bayesian perspective." In: *Acta numerica* 19, pp. 451–559.

Stuart, Andrew M (2010b). "Inverse problems: a Bayesian perspective." In: *Acta numerica* 19, pp. 451–559.

Sznitman, Alain-Sol (1991). "Topics in propagation of chaos." In: *Ecole d'été de probabilités de Saint-Flour XIX—1989* 1464, pp. 165–251.

Taghvaei, Amirhossein and Bamdad Hosseini (2022). "An Optimal Transport Formulation of Bayes' Law for Nonlinear Filtering Algorithms." In: *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 6608–6613.

Taghvaei, Amirhossein and Prashant G Mehta (2020). "An optimal transport formulation of the ensemble Kalman filter." In: *IEEE Transactions on Automatic Control* 66.7, pp. 3052–3067.

Takens, Floris (2006). "Detecting strange attractors in turbulence." In: *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*. Springer, pp. 366–381.

Temam, Roger (2012). *Infinite-dimensional dynamical systems in mechanics and physics*. Vol. 68. Springer Science & Business Media.

Teschl, G. (2012). *Ordinary Differential Equations and Dynamical Systems*. Graduate studies in mathematics. American Mathematical Society. ISBN: 9780821883280.

Tippett, Michael K., Jeffrey L. Anderson, Craig H. Bishop, Thomas M. Hamill, and Jeffrey S. Whitaker (July 2003). "Ensemble Square Root Filters." EN. In: *Monthly Weather Review* 131.7, pp. 1485–1490. ISSN: 1520-0493, 0027-0644. DOI: 10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2. (Visited on 04/08/2021).

Tong, Xin T, Andrew J Majda, and David Kelly (2016a). "Nonlinear stability and ergodicity of ensemble based Kalman filters." In: *Nonlinearity* 29.2, p. 657.

Tong, Xin T, Andrew J Majda, and David Kelly (2016b). "Nonlinear stability of the ensemble Kalman filter with adaptive covariance inflation." In: *Communications in Mathematical Sciences* 14.5.

Tripura, Tapas and Souvik Chakraborty (2023). "Wavelet Neural Operator for solving parametric partial differential equations in computational mechanics problems." In: *Computer Methods in Applied Mechanics and Engineering* 404, p. 115783. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115783. URL: https://www.sciencedirect.com/science/article/pii/S0045782522007393.

Tsai, Yao-Hung Hubert, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov (Nov. 2019). "Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 4344–4353.

Tucker, Warwick (1999). "The Lorenz attractor exists." In: *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 328.12, pp. 1197–1202.

Vaes, U. (2024). "Sharp propagation of chaos for the ensemble Langevin sampler." In: *J. London Math. Soc.* 110.5, e13008.

Van Leeuwen, Peter Jan (2020a). "A consistent interpretation of the stochastic version of the Ensemble Kalman Filter." In: *Quarterly Journal of the Royal Meteorological Society* 146.731, pp. 2815–2825.

Van Leeuwen, Peter Jan (2020b). "A consistent interpretation of the stochastic version of the ensemble Kalman filter." In: *Quarterly Journal of the Royal Meteorological Society* 146, pp. 2815–2825. DOI: https://doi.org/10.1002/qj.3819.

Van Leeuwen, Peter Jan and Geir Evensen (1996). "Data assimilation and inverse methods in terms of a probabilistic formulation." In: *Monthly Weather Review* 124.12, pp. 2898–2913.

Van Leeuwen, Peter Jan, Hans R Künsch, Lars Nerger, Roland Potthast, and Sebastian Reich (2019a). "Particle filters for high-dimensional geoscience applications: A review." In: *Q. J. R. Meteorol. Soc.* 145.723, pp. 2335–2365.

Van Leeuwen, Peter Jan, Hans R. Künsch, Lars Nerger, Roland Potthast, and Sebastian Reich (2019b). "Particle filters for high-dimensional geoscience applications: A review." In: *Q. J. Royal Meteorol. Soc.* 145, pp. 2335–2365. DOI: 10.1002/qj.3551.

Vanden-Eijnden, Eric et al. (2003). "Fast communications: Numerical techniques for multi-scale dynamical systems with stochastic effects." In: *Communications in Mathematical Sciences* 1.2, pp. 385–391.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc.

Vetra-Carvalho, Sanita, Peter Jan Van Leeuwen, Lars Nerger, Alexander Barth, M. Umer Altaf, Pierre Brasseur, Paul Kirchgessner, and Jean-Marie Beckers (2018). "State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems." In: *Tellus A: Dynamic Meteorology and Oceanography* 70.1, pp. 1–43. DOI: 10.1080/16000870.2018.1445364.

Villani, C. (2008). *Optimal transport: Old and new*. Vol. 338. Springer Science & Business Media.

Villani, Cédric (2021). *Topics in optimal transportation*. Vol. 58. American Mathematical Soc.

Vishny, David, Matthias Morzfeld, Kyle Gwirtz, Eviatar Bach, Oliver R. A. Dunbar, and Daniel Hodyss (Aug. 2024). "High-Dimensional Covariance Estimation From a Small Number of Samples." en. In: *Journal of Advances in Modeling Earth Systems* 16.9, e2024MS004417. ISSN: 1942-2466. DOI: 10.1029/2024MS004417. (Visited on 08/30/2024).

W. Acevedo, J. de Wiljes, and S. Reich (2017). "Second-order accurate ensemble transform particle filters." In: *SIAM J. Sci. Comput.* 39, A1834–A1850. DOI: 10.1137/16M1095184.

Wang, Sifan, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris (2025a). "CViT: Continuous Vision Transformer for Operator Learning." In: *The Thirteenth International Conference on Learning Representations*.

Wang, Sifan, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris (2025b). "CViT: Continuous Vision Transformer for Operator Learning." In: *The Thirteenth International Conference on Learning Representations*.

Welch, Greg, Gary Bishop, et al. (1995). "An introduction to the Kalman filter." In.

Wen, Qingsong, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun (2023). "Transformers in time series: A survey." In: *International Joint Conference on Artificial Intelligence(IJCAI)*.

Wiljes, Jana de, Sebastian Reich, and Wilhelm Stannat (2018). "Long-time stability and accuracy of the ensemble Kalman–Bucy filter for fully observed processes and small measurement noise." In: *SIAM Journal on Applied Dynamical Systems* 17.2, pp. 1152–1181.

Wright, Matthew A and Joseph E. Gonzalez (2021). "Transformers are Deep Infinite-Dimensional Non-Mercer Binary Kernel Machines." In: *arXiv preprint* arXiv:2106.01506.

X. Wang, C.H. Bishop, and S.J. Julier (2004). "Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble?" In: *Monthly Weather Review* 132, pp. 1590–1605.

Y. Cheng and S. Reich (2015). "Data assimilation: A coupling of measure perspective." In: *Frontiers in Applied Dynamical Systems*. Vol. 2. New York: Springer-Verlag.

Yang, Lucia Minah and Ian Grooms (2021). "Machine learning techniques to construct patched analog ensembles for data assimilation." In: *Journal of Computational Physics* 443, p. 110532.

Yang, Tao, Prashant G. Mehta, and Sean P. Meyn (2013). "Feedback particle filter." In: *IEEE Trans. Automat. Control* 58.10, pp. 2465–2480. ISSN: 0018-9286. DOI: 10.1109/TAC.2013.2258825.

Yun, Chulhee, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar (2020). "Are Transformers universal approximators of sequence-to-sequence functions?" In: *International Conference on Learning Representations*.

Zech, J. and Y. Marzouk (2022). "Sparse Approximation of Triangular Transports, Part I: The Finite-Dimensional Case." In: *Constructive Approximation* 55, pp. 919–986.

Zhang, Yuanzhao and William Gilpin (2025). "Zero-shot forecasting of chaotic systems." In: *The Thirteenth International Conference on Learning Representations*.

Zhou, Xu-Hui, Zhuo-Ran Liu, and Heng Xiao (Oct. 2024). *BI-EqNO: Generalized Approximate Bayesian Inference with an Equivariant Neural Operator Framework*. DOI: 10.48550/arXiv.2410.16420. (Visited on 10/23/2024).

Zupanski, Milija (2005). "Maximum likelihood ensemble filter: Theoretical aspects." In: *Monthly Weather Review* 133.6, pp. 1710–1726.