

Generative Modeling of Earthquake Ground Motion

Thesis by
Yaozhong Shi

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy



CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2026
Defended May 20, 2026

© 2026

Yaozhong Shi

ORCID: 0000-0001-8863-2026

All rights reserved

ACKNOWLEDGEMENTS

My time at Caltech has been shaped by extraordinary mentors, collaborators, colleagues, and friends. I am deeply grateful to all those who have supported me throughout this journey.

I am especially grateful to my advisors, Prof. Domniki Asimaki and Prof. Zachary E. Ross, for their guidance, trust, and encouragement over the course of my Ph.D. Their mentorship has shaped not only the direction of this dissertation, but also the way I think about research, scientific rigor, and meaningful problems. I am also thankful to my thesis committee chair, Prof. Joel W. Burdick, and to my committee members, Prof. Monica D. Kohler and Prof. Grigorios Lavrentiadis.

I have also been fortunate to work with Dr. Kamyar Azizzadenesheli, whose mentorship and mathematical perspective on machine learning have deeply influenced my thinking and broadened the scope of my research. I also thank the members and collaborators of the Asimaki and Ross groups. In particular, I am grateful to Prof. Grigorios Lavrentiadis for his guidance and collaboration beyond his committee role, and to Caifeng Zou for many helpful discussions and collaborations. I am equally grateful to all of my co-authors listed in the Published Content and Contributions section.

I would like to thank Caltech, the Division of Engineering and Applied Science, and the Seismological Laboratory for providing an exceptional environment for learning and research. I am also grateful to the faculty, staff, and administrative teams whose support has been essential throughout my graduate studies.

Finally, I thank my family and friends, both within and outside Caltech, for their unwavering support, encouragement, and companionship. Their care and confidence in me have been a constant source of strength.

ABSTRACT

Reliable earthquake ground-motion time histories are essential for seismic hazard analysis, structural design, and the risk assessment of distributed infrastructure systems. However, observed records of large earthquakes remain sparse, while physics-based simulations are computationally expensive and difficult to deploy at the scale required for uncertainty quantification. This thesis develops generative modeling frameworks for efficient, scenario-specific synthesis of earthquake ground motion, with an emphasis on neural operators, function-space learning, and scalable regional wavefield generation.

The work begins with conditional generative adversarial neural operators for broadband three-component ground-motion synthesis at individual sites, conditioned on earthquake source, path, and site parameters. It then develops broader methodological foundations for functional generative modeling, including operator flow matching for stochastic-process learning and mesh-informed neural operators for domain- and discretization-agnostic generation on complex geometries. Building on these advances, the thesis introduces Ground-Motion Flow, a physics-inspired latent operator flow-matching framework for regional coherent ground-motion synthesis. By generating wavefields in a physics-aligned latent representation and reconstructing full spatiotemporal motions through neural operators, the framework produces spatially coherent earthquake wavefields conditioned on physical parameters with orders-of-magnitude acceleration relative to direct numerical simulation.

Together, these studies establish a progression from single-site waveform synthesis to regional-scale, uncertainty-aware ground-motion wavefield generation. The results demonstrate the potential of functional generative models to complement physics-based simulation and empirical ground-motion modeling, opening a path toward fast, probabilistic, and scalable seismic hazard workflows for resilient infrastructure systems.

PUBLISHED CONTENT AND CONTRIBUTIONS

Shi, Yaozhong, Grigorios Lavrentiadis, Konstantinos Tsalouchidis, Zachary E Ross, David McCallen, Caifeng Zou, Kamyar Azizzadenesheli, and Domniki Asimaki (2026a). “Large-Scale 3D Ground-Motion Synthesis with Physics-Inspired Latent Operator Flow Matching”. In: *arXiv preprint arXiv:2603.17403 (Under review)*.

Contribution: Y.S. led the conception, method development, implementation, experiments, analysis, and writing of the manuscript.

Shi, Yaozhong, Zachary Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2026b). “Stochastic process learning via operator flow matching”. In: *Advances in Neural Information Processing Systems* 38, pp. 38186–38226.

Contribution: Y.S. led the conception, method development, implementation, experiments, analysis, and writing of the manuscript.

Shi, Yaozhong, Zachary E. Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2025). “Mesh-Informed Neural Operator : A Transformer Generative Approach”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=K8qAuRfv0G>.

Contribution: Y.S. led the conception, method development, implementation, experiments, analysis, and writing of the manuscript.

Shi, Yaozhong, Grigorios Lavrentiadis, Domniki Asimaki, Zachary E Ross, and Kamyar Azizzadenesheli (2024). “Broadband ground-motion synthesis via generative adversarial neural operators: Development and validation”. In: *Bulletin of the Seismological Society of America* 114.4, pp. 2151–2171. ISSN: 0037-1106. DOI: <https://doi.org/10.1785/0120230207>.

Contribution: Y.S. led the conception, method development, implementation, experiments, analysis, and writing of the manuscript.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	iv
Published Content and Contributions	v
Table of Contents	v
List of Illustrations	viii
List of Tables	xviii
Chapter I: Introduction	1
1.1 Modeling Earthquake Ground Motions	2
1.2 Generative Modeling for Ground-Motion Synthesis	3
1.3 Functional Generative Models	3
1.4 From Single Stations to Regional Synthesis	4
1.5 Structure and Scope of This Thesis	4
Chapter II: Broadband Ground Motion Synthesis via Generative Adversarial	
Neural Operators: Development and Validation	6
2.1 Introduction	6
2.2 Methods	10
2.3 Training with Simulated Ground Motions: Verification	16
2.4 Training with Recorded Ground Motions: Verification and Validation	19
2.5 Summary and Discussion	30
Chapter III: Stochastic Process Learning via Operator Flow Matching	36
3.1 Introduction	36
3.2 Related Work	39
3.3 Operator Flow Matching	41
3.4 Experiments	47
3.5 Conclusion	51
Chapter IV: Mesh-Informed Neural Operator: A Transformer Generative	
Approach	52
4.1 Introduction	52
4.2 Preliminaries	54
4.3 Methods	58
4.4 Experiments	61
4.5 Conclusions	64
Chapter V: Scalable Physics-Inspired Generative Modeling of Coherent Re-	
gional Earthquake Wavefields	66
5.1 Introduction	67
5.2 Results	69
5.3 Discussion	79
5.4 Methods	84
Chapter VI: Conclusions and Future Work	92

6.1 Summary of Previous Chapters	92
6.2 Future Directions	93
Appendix A: Appendix for Chapter 2	95
Appendix B: Appendix for Chapter 3	101
B.1 Potential questions and answers	101
B.2 Background: Flow Matching, Gaussian Measures on Function Spaces, and the Cameron–Martin Theorem	106
B.3 Stochastic process learning	107
B.4 Model stochastic process with infinite-dimensional flow matching via Kolmogorov Extension Theorem	108
B.5 Universal Functional Regression	111
B.6 Marginal (dynamic) optimal-transport flow matching in function space via optimal coupling and dynamic Kantorovich formulation . .	111
B.7 Derivation of Eq. B.5	113
B.8 Derivation of Eq. 3.14	115
B.9 Proof of Proposition 3.3.1	115
B.10 Example of Posterior Samples	117
B.11 Co-domain functional regression with OFM	119
B.12 Posterior sampling with Stochastic Gradient Langevin Dynamics . .	120
B.13 Prior learning with OFM	120
B.14 Additional results of 1D GP regression with more complex kernels .	123
B.15 Details of experimental setup	125
B.16 Ablation and scaling studies for mini-batch optimal transport and Hutchinson trace estimator	127
B.17 Detailed analysis of OFM and comparison with existing methods . .	129
Appendix C: Appendix for Chapter 4	133
C.1 Experimental setup	133
C.2 SWD and MMD as general metrics for functional generation tasks . .	136
C.3 Connection and Difference to Prior Works	139
C.4 Additional Results	141
C.5 Ablation Study	141
C.6 Empirical Analysis of Scaling and Stability	143
C.7 Comparison with Transolver under Identical Settings	144
Appendix D: Appendix for Chapter 5	147
D.1 Details of simulation dataset preparation	147
D.2 Details of experimental setup	154
D.3 Additional Results	154
D.4 Ablation Study	159
Bibliography	167

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Schematic illustration of earthquake rupture, seismic-wave propagation, and earthquake early warning.	1
1.2 Common approaches for characterizing and generating earthquake ground motions.	2
2.1 Schematic of GANO input-output structure used in this work: given a set of conditional variables (left), the trained GANO (middle) produces three-component acceleration time series (right).	13
2.2 Schematic of cGM-GANO architecture, and generator - discriminator interaction. The input conditional variables are $\mathbf{M}$, R_{rup} , V_{S30} and f_{type} . The input function for the generator is finite duration Gaussian random field sample while the output is the normalized three-component acceleration time series and three PGAs. The input function of the discriminator are the real or synthetic three-component acceleration time series while the output is score evaluating the realism of the input ground motions.	13
2.3 Data distribution of BBP dataset. Subfigures (a) and (b) show cross-plot between $\log_{10}(\text{PGA})$ and magnitude and rupture distance, respectively	19
2.4 Scenario event comparisons for cGM-GANO trained on the BBP dataset. Each vertical sub-panel depicts the Fourier Amplitude Spectrum (FAS), aleatory standard deviation of FAS, Rotd50 Pseudo-spectral acceleration (PSA) spectrum, and aleatory standard deviation of Rotd50 PSA. The title of each plot corresponds to the midpoint of each bin, and N_{obs} is the number of observed records within each bin. The lines represent the mean in log space of the horizontal components of the synthetic (solid line) or BBP (dashed line) data, while the shaded zone represents the 16 th to 84 th percentile range.	20

2.5	Comparison of cGM-GANO magnitude and distance scaling with BBP dataset for PSA RotD50($T=1$ s). Solid lines represent the mean scaling in log space. The shaded region shows the 2^{nd} to 98^{th} percentile of aleatory variability. Solid dots correspond to the training data.	21
2.6	Residual plots of the BBP trained model. Subfigures (a) and (b) show the EAS residuals for strike-slip and reverse faults, respectively. Subfigures (c) and (d) show the RotD50 PSA residuals for strike-slip and reverse faults, respectively.	22
2.7	Distribution of earthquake magnitude, rupture distance, and V_{S30} versus $\log_{10}(\text{PGA})$ for the subduction (first row) and shallow crustal (second row) events.	22
2.8	Accelerograms comparisons real and synthetically generated ground motions for cGM-GANO trained on the Kik-net dataset.	23
2.9	Scenario event comparisons for cGM-GANO trained on the kik-net dataset. Each sub-panel depicts the Fourier Amplitude Spectrum (FAS), aleatory standard deviation of FAS, Rotd50 Pseudo-spectral acceleration (PSA) spectrum, and aleatory standard deviation of Rotd50 PSA. The title of each plot corresponds to the midpoint of each bin, and N_{obs} is the number of observed records within each bin. The lines represent the mean in log space of the horizontal components of the synthetic (solid line) or kik-net (dashed line) data, while the shaded zone represents the 16^{th} to 84^{th} percentile range.	25
2.10	Residual plots of the kik-net trained model. Subfigures (a) and (b) show the horizontal EAS residuals for subduction and shallow crustal events, respectively. Subfigures (c) and (d) show the vertical EAS residuals for subduction and shallow crustal events, respectively. Subfigures (e) and (f) show the horizontal RotD50 PSA residuals for subduction and shallow crustal events, respectively. Subfigures (g) and (h) show the vertical PSA residuals for subduction and shallow crustal events, respectively.	26
2.11	Comparison of horizontal and vertical FAS residuals for $f=5.0$ Hz against magnitude (subfigure (a) and (d)), R_{rup} (subfigure (b) and (e)), and V_{S30} (subfigure (c) and (f))	27

- 2.12 Comparison of horizontal and vertical PSA residuals for $T=0.2$ s ($f=5.0$ Hz) against magnitude (subfigure (a) and (d)), R_{rup} (subfigure (b) and (e)), and V_{S30} (subfigure (c) and (f)) 28
- 2.13 Inter-period correlation of horizontal PSA residuals. Between $T = 1$ s and $T = 2$ s in subfigure (a), and between $T = 1$ s and $T = 0.2$ s in subfigure (b). 29
- 2.14 Comparison PSA distance, magnitude, and V_{S30} scaling of cmGM-GANO with Kik-net dataset. Left panels, from top to bottom, show the distance scaling for $T = 0.1, 1.0$, and 10 s. Top right panel shows the magnitude scaling for $T = 1.0$ s. Bottom right panel shows the V_{S30} scaling for $T = 1.0$ s. Solid lines represent the mean scaling in log space. The shaded region shows the 2^{nd} to 98^{th} percentile of aleatory variability. Solid dots correspond to the kik-net data. 30
- 2.15 Comparison of cGM-GANO PSA scaling for shallow crustal events against the NGA-West2 GMMs. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $T = 0.1, 0.2, 1, 2$ s ($f = 10, 5, 1, 0.5$ Hz), respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16^{th} to 84^{th} percentile. 31
- 2.16 Comparison of cGM-GANO EAS scaling for shallow crustal events against the BA18 EAS GMMs. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $f = 0.5, 1, 5$, and 10 Hz, respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16^{th} to 84^{th} percentile. 31
- 2.17 Comparison of cGM-GANO PSA scaling for subduction events against BC-Hydro GMM. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $T = 0.1, 0.5, 1$, and 2 s ($f = 10, 2, 1$, and 0.5 Hz), respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16^{th} to 84^{th} percentile. 32

3.1	Two-phase strategy for prior learning and posterior sampling. In the prior-learning phase, OFM leverages the marginal optimal-transport path to learn a bijective mapping between a predefined GP and the unknown stochastic process that generates the training data. In the posterior-sampling phase, the learned prior is frozen; given noisy, partial observations, the exact posterior is obtained via Bayes' theorem. SGLD, aided by the Hutchinson trace estimator, then enables efficient and robust sampling.	38
3.2	Operator Flow Matching (OFM) regression on Navier–Stokes functional data with resolution 64×64 . (a) 32 random observations (only 0.7%). (b) Ground truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.	39
3.3	OFM regression on GP data. (a) Ground-truth GP regression with observed data and posterior samples. (b) OFM regression with observed data and posterior samples. (c) Standard-deviation comparison between the true GP posterior and OFM predictions.	48
3.4	OFM regression on transformed Gaussian process data.	48
3.5	OFM regression on black hole data with resolution 64×64 . (a) 32 random observations. (b) Ground-truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.	49
4.1	Overview of the MINO architecture. The geometry encoder uses a GNO as a domain-agnostic tokenizer, followed by several cross-attention blocks and an optional latent processor. The decoder then employs a distinct cross-attention mechanism to map the latent representation back to the target locations.	54
4.2	Inference and zero-shot Generation with MINO. (a) MINO gradually transforms a GP sample to the data sample under flow matching paradigm. (b) Zero-shot generation at varying spatial scales by directly transforming finer GP samples to finer data samples.	59

4.3	Visualization of generation and zero-shot super-resolution by MINO-U. (a) Navier-Stokes samples generated on the original mesh (4,096 nodes) and a finer mesh (25,600 nodes). (b) Global Climate sample generated on the original mesh (4,140 nodes) and a finer mesh (16,560 nodes).	64
5.1	Workflow for physics-based ground-motion simulation in the San Francisco Bay Area. (left) A 3D representation of the heterogeneous geological structure beneath the Bay Area, adopted from https://www.usgs.gov/media/images/3d-geologic-model with permission. (center) The problem (computational) domain used for the numerical simulations. (right) An example of finite-fault event simulation.	69
5.2	GMFlow model architecture and two-stage synthesis pipeline. (top) The original surface wavefield u is filtered and downsampled to a low-frequency representation u_f ; the reverse mapping is performed by a super-resolution neural operator. (bottom) An encoder-decoder neural operator encodes u_f to a latent code z_1 , and decodes it back to u_f ; conditional flow matching transports $z_0 \sim \mathcal{N}(0, I)$ to z_1 given physical conditions c	70
5.3	Evaluation of GMFlow on a representative M_w 4.4 point-source scenario. (A–B) Side-by-side comparison of the ground-truth velocity wavefield and a single stochastic realization generated by GMFlow at selected time steps (snapshots). Orange dash and solid lines represent Hayward fault and coastline, respectively. (C) Peak ground velocity (PGV; m/s), comparing the ground truth, one synthetic realization, the ensemble mean ($N = 100$), and the predictive uncertainty. (D) Spatial maps of the Fourier amplitude spectrum (FAS) at 0.25 Hz (top) and 0.75 Hz (bottom). (E) Normalized cross-correlation (NCC) for spatiotemporal coherence analysis, showing the peak correlation coefficient (top) and the corresponding time lag (bottom). (F) Geographic map of the computational domain, with epicenter (purple star) and the virtual seismic profiles A1–A2 and B1–B2 indicated. (G–H) Time–distance plots along profiles A1–A2 and B1–B2.	71
5.4	Evaluation of GMFlow on a representative M_w 7.0 finite-rupture scenario.	72

- 5.5 Station-level spectral evaluation across magnitude range. (A, C, E)** Spatial distribution of four randomly selected validation stations (triangles) and the epicenter (star) for M_w 4.4, 6.0, and 7.0 scenarios. **(B, D, F)** Comparison of the horizontal (power-mean) and vertical Fourier Amplitude Spectra (FAS) for the ground truth (blue) and synthetic realizations (orange/red). The solid lines for the synthetics represent the geometric mean of $N = 100$ realizations, while the shaded regions indicate the ± 1 geometric standard deviation. 73
- 5.6 Statistical distribution and magnitude scaling laws of generated ground motions. (A)** Pointwise distributions of $\log_{10}(\text{PGV})$ for M_w 4.4, 6.0, and 7.0 comparing held-out data to the same amount of synthetic realizations generated under identical conditions. **(B)** Pointwise distributions of $\log_{10}(\text{FAS})$ at 0.25, 0.5, 0.75, and 0.96 Hz. **(C)** PGV magnitude scaling from M_w 4.4–7.0 (0.1 magnitude increments), summarized by the spatial median per event over 100 realizations per bin. **(D)** Analogous magnitude scaling for FAS at the same frequencies; markers and error bars denote the ensemble geometric mean and standard deviation. 74
- 5.7 Frequency-dependent spectral residuals. (A–C)** Residual statistics of horizontal Fourier amplitude spectra (FAS) as a function of frequency for M_w 4.4, 6.0, and 7.0, computed over 300 test events using ensembles of 100 synthetic realizations per event; markers show the mean and median, and error bars indicate the 16th/84th percentiles. **(D–E)** Spatial maps of the mean FAS residual at 0.25, 0.50, 0.75, and 0.96 Hz for M_w 4.4 and M_w 6.0 events. Black dash and solid lines represent Hayward fault and coastline, respectively. 75
- A.1** Example of frequency content shift between acceleration, velocity, and displacement time series, depicted in the time and frequency domain. Example time series corresponds to event id ISKH020703250942 with $M_w = 6.9$ and $R_{rup} = 23\text{km}$ from the Kik-net dataset. The left-hand panels show the acceleration, velocity, and displacement time series from top to bottom, while the right-hand panels show the corresponding normalized frequency amplitude spectra. Each spectrum is normalized by its maximum amplitude. The dash lines show the FAS trend in 0.02Hz $\sim$ 10Hz frequency range. 96

A.2	An example of the location of the fault (black line) and stations (red dots) used in BBP simulation.	96
A.3	Time-domain and frequency-domain waveforms comparisons of BBP and GANO generated ground motions. Subfigure (a) corresponds to a three-component waveform comparison for $M_w = 7$, $R_{rup} = 10km$, and strike-slip style of faulting. Subfigure (b) corresponds to a three-component waveform comparison for $M_w = 7$, $R_{rup} = 10km$, and reverse style of faulting.	97
A.4	Inter-frequency correlation of synthetically generated ground motions through cGM-GANO based on the BBP training dataset.	97
A.5	Residual plots of the kik-net trained model based on the validation kik-net dataset (10 % of the total kik-net dataset) . Subfigures (a) and (b) show the horizontal EAS residuals for subduction and shallow crustal events, respectively. Subfigures (c) and (d) show the vertical EAS residuals for subduction and shallow crustal events, respectively. Subfigures (e) and (f) show the horizontal RotD50 PSA residuals for subduction and shallow crustal events, respectively. Subfigures (g) and (h) show the vertical PSA residuals for subduction and shallow crustal events, respectively.	98
A.6	Velocity waveforms comparisons between data and synthetic ground motions from cGM-GANO trained on the Kik-net dataset, unit in m/s	99
A.7	Displacement waveforms comparisons between data and synthetic ground motions from cGM-GANO trained on the Kik-net dataset, unit in m	100
B.1	OFM regression on 2D GP data with resolution 32×32 . (a) 32 random observations. (b) Predicted mean from OFM. (c) Ground-truth mean from GP regression. (d) Misfit of the predicted mean. (e) Misfit of the predicted standard deviation. (f) Posterior samples from OFM. (g) Posterior samples from GP regression.	117
B.2	OFM regression on Navier-Stokes functional data with resolution 64×64 . (a) 32 random observations. (b) Ground truth sample (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from best fitted GP.	117

B.3	OFM regression on MNIST-SDF with resolution 64×64 . (a) 64 random observations. (b) Ground-truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.	118
B.4	OFM regression on NS data. (left) Posterior samples from OFM. (right) Posterior samples from GP regression.	118
B.5	OFM regression on black hole data. (left) Posterior samples from OFM. (right) Posterior samples from GP regression.	118
B.6	OFM regression on MNIST-SDF data. (left) Posterior samples from OFM. (right) Posterior samples from GP regression.	118
B.7	OFM regression on co-domain GP data with resolution 32×32 . (a) 32 random observations at co-locations. (b) Predicted mean from OFM. (c) Ground truth mean from GP regression. (d) Misfit of the predicted mean. (e) Misfit of predicted standard deviation.	119
B.8	OFM for 1D GP prior learning, evaluated at resolution=128. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	122
B.9	OFM for 1D GP prior learning, evaluated at resolution=256. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	122
B.10	OFM for 1D GP prior learning, evaluated at resolution=512. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	122
B.11	OFM for 1D TGP prior learning, evaluated at resolution=128. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	123
B.12	OFM for 1D TGP prior learning, evaluated at resolution=256. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	123
B.13	OFM for 1D TGP prior learning, evaluated at resolution=512. (left two) Random samples from ground truth and generated by OFM. (right two) Autocovariance and histogram comparison	123
B.14	OFM for 2D Navier–Stokes prior learning, evaluated at resolution 64×64 . (a) Random samples from the ground truth. (b) Random samples generated by OFM. (c) Autocovariance comparison.	124

B.15	OFM for 2D N-S prior learning, evaluated at 128×128 resolution (zero-shot super-resolution)	124
B.16	OFM for 2D black hole prior learning, evaluated at resolution 64×64 . (a) Random samples from the ground truth. (b) Random samples generated by OFM. (c) Autocovariance comparison.	125
B.17	OFM for 2D black hole prior learning, evaluated at 128×128 resolution (zero-shot super-resolution)	125
B.18	OFM for 2D MNIST-SDF prior learning, evaluated at 64×64 resolution. (top) Random samples from ground truth. (bottom) Random samples generated by OFM.	126
B.19	OFM for 2D MNIST-SDF prior learning, evaluated at 128×128 resolution. (top) Random samples from ground truth. (bottom) Random samples generated by OFM.	126
B.20	Log likelihood by the integration of divergence, (left) plot for first GP sample. (right) plot for second GP sample	129
C.1	Consistency of SWD and MMD with respect to the number of discretization points. The metrics remain stable as the number of observation points is varied for three comparison scenarios: two synthetic GPs (left), a GP vs. a regular grid dataset (center), and a GP vs. an irregular mesh dataset (right).	139
C.2	Generation of Navier-Stokes samples under different baselines	142
C.3	Generation of Darcy-Flow samples	142
C.4	Generation of Shallow Water samples	143
C.5	Generation of Mesh GP samples	143
D.1	Computational domain adopted for the SFBA and model parameters .	149
D.2	(a) Map view of the free surface shear-wave velocity in the simulation domain, extracted from the United States Geological Survey velocity model, with the E-W red lines indicating the location of the velocity model cross-sections shown in (b); (b) Vertical cross-sections of the velocity model with the Hayward fault at 0 normal distance. (adopted from Pinilla-Ramos et al. 2025 with permission)	150

D.3	(a) Small-magnitude seismic source characterization as a double-couple point-source function defined by its focal mechanism and source time function; (b) Large-magnitude seismic source characterization following the Graves-Pitarka rupture model; (c) Example of a pure stochastic rupture realization for the Hayward fault plane, shown through its total slip, rise time and slip rate ; (d) slip rate of points at various depths within the fault-plane as a function of time, normalized to show a unit area of total slip, developed within a normalized rise time of 1 second.	151
D.4	Envelope of calculated Green's functions, rupture plane dimensions and positions for Magnitude 7 and Magnitude 6 earthquake events . .	151
D.5	Fault-plane discretization with point-sources at 200-m grid, repositioned in the corresponding centroids of 1x1km sub-faults	152
D.6	NCC analysis using randomly selected reference stations for the representative M_w 4.4 event.	156
D.7	Validation of GMFlow on a representative M_w 6.0 finite rupture source scenario.	157
D.8	Zero-shot super-resolution. Ground-truth wavefield snapshots are provided at spatial resolution 256×128 , whereas the synthetic samples are super-resolved to spatial resolution 512×256 . (A, C, E) Ground-truth snapshots for representative M_w 4.4, M_w 6, and M_w 7 events, respectively. (B, D, F) Corresponding zero-shot super-resolved synthetic snapshots for the same events.	158
D.9	Frequency-dependent spectral residuals for different cutoff frequency f_c	161
D.10	Evaluation of spectral-calibrated GMFlow on a representative M_w 7.0 point-source scenario.	163
D.11	NCC analysis using randomly selected reference stations for the spectral-calibrated representative M_w 7 event.	164
D.12	Statistical distribution and magnitude scaling laws of spectral-calibrated generated ground motions	165
D.13	Frequency-dependent spectral residual with calibrated synthetics. . .	165

LIST OF TABLES

<i>Number</i>	<i>Page</i>
3.1 Comparison of OFM with baseline models: GP regression; OpFlow (Shi et al., 2024a); Conditional models: NP (Garnelo et al., 2018); Attentive NP (Kim et al., 2019, ANP); Convolutional Conditional NP (Gordon et al., 2019, ConvCNP); Deep variational GP (Salimbeni et al., 2017, DGP); Deep Sigma Point Process (Jankowiak et al., 2020, DSSP); Metrics SMSE and MSLL used for 1D and 2D GP example. Mean squared error for the predicted mean (μ) and standard deviation (σ) are used for TGP example. Performance of GP regression for 1D and 2D GP are removed (marked with '-'), which are taken as the ground truth. Best performance in bold.	50
4.1 Comparison of MINO with other architectures. "local $\rightarrow$ global" receptive field gathers global context through stacked local operations. "local + global" receptive field combines local and global information directly. A detailed discussion is provided in Appendix C.3	58
4.2 Summary of experimental benchmarks, covering diverse domains with both regular and irregular grids	61
4.3 Generation performance on benchmarks with irregular grids. Best performance in bold	64
4.4 Generation performance on benchmarks with regular grids. Evaluation metrics include the Sliced Wasserstein Distance (SWD), Maximum Mean Discrepancy (MMD), and Mean Squared Error (MSE) for spectra, autocovariance, and point-wise density. Best performance is indicated in bold.	65
4.5 Computational efficiency and performance comparison between MINO variants and Transolver during training and inference on Navier-Stokes benchmark.	65
B.1 Comparison of OFM with baseline models on 1D GP with Gibbs kernel and RQK. Best performance in bold.	124
B.2 Parameters used in experiments of prior learning	127
B.3 Parameters used in regression experiments - Part A	127
B.4 Parameters used in regression experiments - Part B	127
B.5 Parameters used in regression experiment - Part C	127

B.6	Scaling study for the size of mini-batch given optimal transport plan, best performance in bold	128
B.7	Scaling study for the number of noise samples of Hutchinson trace estimator	129
B.8	High-level comparison of OFM and NP.	131
C.1	Hyperparameter settings for all baseline models.	134
C.2	Variance analysis of the averaged-SWD metric. We report the mean and standard deviation over 20 trials for different numbers of averaging runs (n_{run}).	138
C.3	Architectural Comparison: MINO vs. GINO / UPT	145
C.4	Ablation study on key architectural components of MINO across four benchmarks. We compare our standard MINO-T against two variants: one replacing the encoder’s cross-attention with self-attention, and another using a simpler position-only decoder query similar to prior work like UPT. Lower SWD/MMD scores are better; best results are in bold	146
C.5	Performance comparison over eight random seeds. We report the mean $\pm$ standard deviation in parentheses for each metric. Lower is better.	146
C.6	Performance of MINO-U when trained on a subset of the Navier-Stokes dataset.	146
C.7	Performance comparison with Transolver under identical settings. Best Performance in bold	146
D.1	Hyperparameters and training details of the SNO, AENO, and Flow Matching models.	155
D.2	Model-size ablation for latent flow matching. Quantitative comparison of the 16M and 128M CFM models using PGV, FAS-based Wasserstein distance at multiple frequencies, and RMSE of the mean residual for M_w 4.4, M_w 6.0, and M_w 7.0 events. The lower the better.	160
D.3	Cutoff-frequency ablation for the Physics-Aligned Subspace. Quantitative comparison of models using different cutoff frequencies $f_c \in \{0.5, 0.6, 0.75, 1.0\}$ Hz across representative M_w 4.4, M_w 6.0, and M_w 7.0 events. Metrics include PGV, FAS-based Wasserstein distance at multiple frequencies, and RMSE of the mean residual. The best and second-best results in each column are highlighted in black and brown, respectively.	160

D.4	Effect of post-processing spectral calibration. Quantitative comparison of GMFlow and GMFlow-calibrated for representative M_w 4.4, M_w 6.0, and M_w 7.0 events using PGV, FAS-based Wasserstein distance at selected frequencies, and RMSE of the mean residual. The lower, the better.	163
-----	---	-----

Chapter 1

INTRODUCTION

For the built environment, earthquake risk is ultimately mediated by ground motion: the transient shaking generated by fault rupture and seismic-wave propagation. Earthquakes are produced by the sudden release of accumulated tectonic strain on faults, which radiates seismic waves through the Earth and drives shaking at the surface (Aki et al., 2002; Scholz, 2019). Although the rupture process occurs underground, its consequences are expressed at the surface through ground shaking, permanent deformation, liquefaction, landslides, tsunamis, fires, and cascading failures of infrastructure systems. Large events such as the 2011 Tohoku earthquake illustrate how seismic hazards can cascade through interconnected engineering, energy, and social systems (National Research Council, 2014).

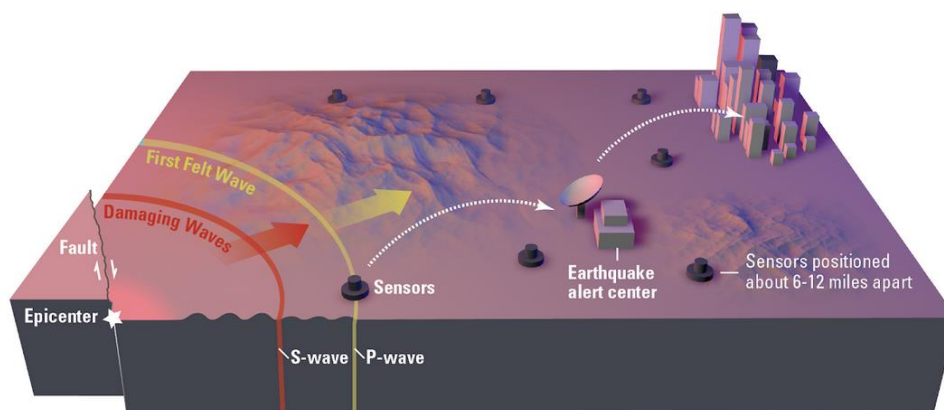


Figure 1.1: Schematic illustration of earthquake rupture, seismic-wave propagation, and earthquake early warning. Seismic energy released by fault slip propagates through the Earth and is recorded by sensor networks. Adapted from (State of California, n.d.).

Understanding and mitigating earthquake risk require reliable characterization of future ground shaking. Seismic hazard analysis quantifies the intensity, exceedance frequency, and variability of shaking that a site or region may experience from future earthquakes (Baker et al., 2021). This information supports building codes, performance-based earthquake engineering, emergency planning, insurance, and the

design of critical facilities such as hospitals, bridges, dams, pipelines, and nuclear power plants. Because future earthquakes cannot be predicted deterministically, hazard analysis must account for uncertainty in earthquake occurrence, rupture properties, wave propagation, and site effects. Risk analysis further combines these shaking estimates with the exposure and vulnerability of infrastructure systems.

Earthquake ground motions provide the primary link between seismic sources and their consequences for the built environment. In engineering practice, they are commonly represented as three-component acceleration, velocity, or displacement time histories. These time histories encode the amplitude, duration, frequency content and nonstationary characteristics of shaking, all of which influence structural and geotechnical response. Modeling ground motion is therefore central not only to estimating scalar intensity measures, but also to generating realistic time histories and spatially coherent wavefields for structural design, seismic hazard analysis, and regional risk assessment.

1.1 Modeling Earthquake Ground Motions

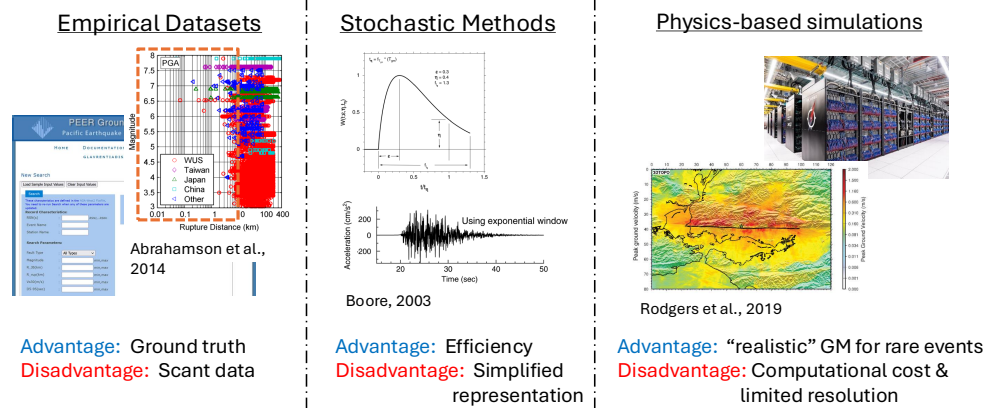


Figure 1.2: Common approaches for characterizing and generating earthquake ground motions: empirical datasets and models, stochastic methods, and physics-based simulations. Modified from (Abrahamson et al., 2014; Boore, 2003; Rodgers et al., 2020).

Ground-motion modeling aims to characterize the amplitude, variability, and time-dependent structure of shaking produced by future earthquakes. Existing approaches can be broadly grouped into empirical datasets and models, stochastic methods, and physics-based simulations, as summarized in Fig. 1.2.

Empirical datasets provide direct observations of earthquake shaking and form the basis of empirical ground-motion models, which relate source, path, and site

parameters to intensity measures. However, observed records remain limited for large earthquakes, near-fault conditions, rare rupture scenarios, and soft-soil sites. Stochastic methods efficiently generate synthetic motions by matching prescribed spectral and temporal properties, but they often rely on simplified representations of source, path, and site effects. Physics-based simulations solve the elastodynamic wave equation in heterogeneous media and provide a more direct representation of seismic-wave propagation, but they remain computationally expensive and are often limited to simulating reliable high-frequency ground motions.

Together, these limitations motivate data-driven generative models that can learn from observed records, physics-based simulations, or hybrid datasets. Such models aim to combine the statistical flexibility of empirical approaches with the waveform-level richness of simulations, enabling rapid sampling of scenario-conditioned ground-motion time histories and regional wavefields.

1.2 Generative Modeling for Ground-Motion Synthesis

Ground-motion synthesis is inherently probabilistic because earthquake shaking is controlled by both prescribed scenario variables and unresolved source, path, and site effects. Even for a fixed scenario, many physically plausible motions may differ in amplitude, duration, phase, frequency content, and spatial structure. Thus, a synthesis model should generate ensembles rather than only a median intensity measure or a single deterministic waveform.

In this thesis, ground-motion synthesis is formulated as conditional generation,

$$u \sim p_{\theta}(u \mid c), \quad (1.1)$$

where u denotes a three-component ground-motion time history or regional wavefield, c denotes the available conditioning variables defining the earthquake scenario, and p_{θ} approximates the corresponding conditional distribution. Sampling from $p_{\theta}(u \mid c)$ produces scenario-consistent ensembles that represent variability arising from rupture complexity, unresolved heterogeneity, attenuation, and local site response.

1.3 Functional Generative Models

Ground motions are naturally functions: a single-station record (waveform) is a function of time, while a regional wavefield is a function of space and time. This viewpoint is important because the same physical process may be sampled at different temporal resolutions, spatial grids or station layouts. Functional generative

models therefore aim to learn probability measures over fields rather than distributions over fixed-dimensional arrays.

Neural operators (Li et al., 2020b) provide a natural architecture for this setting because they approximate mappings between functions, making them suitable for scientific data observed on different resolutions, geometries, and sensor layouts. This thesis develops two methodological components for functional generative modeling: operator flow matching (Shi et al., 2026b), which learns stochastic-process priors and enables sampling and posterior inference in function space, and the Mesh-Informed Neural Operator (Shi et al., 2025), which extends generative modeling to irregular domains and sensor layouts.

1.4 From Single Stations to Regional Synthesis

Single-station synthesis treats each ground motion as an independent time history for an earthquake–station pair. This is useful for site-specific structural analysis, but regional risk assessment requires spatially coherent shaking across many locations. A regional model must therefore generate spatiotemporal wavefields that preserve key physical features of ground motion, including wavefront propagation, spatial amplification, frequency-dependent structure, and cross-site correlations.

The final part of this thesis addresses this challenge through Ground-Motion Flow, a physics-inspired latent operator flow-matching framework (Shi et al., 2026a). The model generates the dominant low-frequency structure of the regional wavefield in a physics-aligned latent space and then lifts this representation to the target spatiotemporal wavefield using a super-resolution neural operator. This avoids direct generation in the full high-dimensional physical space while enabling fast, scenario-conditioned regional ground-motion synthesis.

1.5 Structure and Scope of This Thesis

The remainder of this thesis follows a progression from single-station waveform generation to functional generative modeling and regional wavefield synthesis. Chapter 2 presents a conditional generative adversarial neural operator for broadband three-component ground-motion synthesis. Chapter 3 develops operator flow matching for stochastic-process learning and functional regression. Chapter 4 introduces the Mesh-Informed Neural Operator as a domain- and discretization-agnostic backbone for functional generative modeling. Chapter 5 presents Ground-Motion Flow for coherent regional ground-motion synthesis. Chapter 6 summarizes the main findings and future directions.

The scope of this thesis is generative modeling for earthquake ground-motion synthesis. The goal is not to replace empirical ground-motion models or physics-based simulations, but to complement them with efficient probabilistic surrogates that generate scenario-conditioned ensembles. The single-station work focuses on broadband three-component time histories, while the regional work focuses on simulation-trained coherent wavefield generation in the San Francisco Bay Area. Extending these models to broader regions, recorded datasets, higher-frequency motions, irregular sensor layouts, and sparse observation-conditioned posterior sampling remains an important direction for future work.

Chapter 2

BROADBAND GROUND MOTION SYNTHESIS VIA GENERATIVE ADVERSARIAL NEURAL OPERATORS: DEVELOPMENT AND VALIDATION

This chapter presents a conditional generative adversarial neural operator (cGM-GANO) for broadband three-component ground-motion synthesis. The model generates acceleration time histories conditioned on earthquake and site parameters, including magnitude, rupture distance, site condition, and faulting or tectonic environment.

We verify the method using simulated ground motions from the Southern California Earthquake Center Broadband Platform and validate it using recorded strong-motion data from KiK-net. The results show that cGM-GANO can reproduce the magnitude, distance, and site-condition scaling trends, while also revealing limitations associated with sparse data, near-fault motions, soft-soil sites, and high-frequency stochastic components.

The contents of this chapter are adapted from our publication (Shi et al., 2024b):

Shi, Yaozhong, Grigorios Lavrentiadis, Domniki Asimaki, Zachary E Ross, and Kamyar Azizzadenesheli (2024). “Broadband ground-motion synthesis via generative adversarial neural operators: Development and validation”. In: *Bulletin of the Seismological Society of America* 114.4, pp. 2151–2171. issn: 0037-1106.

Additional details for this chapter are provided in Appendix A. The code associated with this work is available at <https://github.com/yzshi5/GM-GANO>

2.1 Introduction

The demand for reliable ground motion time histories for seismic analysis is growing, yet there are still not enough records of large earthquakes covering the desired range of source, path, and site characteristics (Katsanos et al., 2010), in particular at near-source ground motions or on very soft site conditions. In order to resolve the limitations caused by data scarcity, ground motion records are usually selected and scaled in the time domain or adjusted in the frequency domain to satisfy specified intensity measurements or frequency content (Goulet et al., 2008; Rezaeian et al., 2012). However, scaling or modification of records that is not based on physics

may only scale some intensities or frequency measures while leaving other elements unaltered, leading to unrealistic ground motions (Naeim et al., 1995; Luco et al., 2007; Baker et al., 2021). Additionally, most of the scaling and spectrum matching techniques only modify the two horizontal components making them unsuitable for structural models that are sensitive to all three components (Bhosale et al., 2017).

Methods for synthesizing ground motions offer an alternative solution by directly generating ground motions for the scenarios of interest. These methods can be broadly grouped into “physics-based” numerical simulations and “stochastic-based” approaches, depending on whether the process of generating synthetic waveforms involves physics (Douglas et al., 2008). In general, stochastic-based methods for ground motion generation involve modifying white noise to emulate specific ground motion characteristics (Dabaghi et al., 2017; Baglio et al., 2023; Sabetta et al., 2021; Burks et al., 2014), while physics-based numerical simulations rely on modeling the physics that govern earthquakes, such as wave propagation and seismic source radiation to generate ground motions for a scenario of interest (McCallen et al., 2021a; Paolucci et al., 2021; Touhami et al., 2022).

Physics-based methods have the advantage of generating complex path and site effects, but they are still far from being incorporated in routine engineering applications due to the following reasons: (1) Such approaches demand in-depth knowledge of sources, paths, and site effects, which are usually unavailable for most engineering purposes. (Causse et al., 2008; Frankel, 2009; Rezaeian et al., 2012). (2) they require expensive computations to provide sufficient high-resolution frequency. Nowadays, with the assistance of supercomputers, it is possible to synthesize broadband ground motions up to 10Hz (Rodgers et al., 2020); however, such computational resources are not available for most engineering projects. (3) Even with earthquake simulations capable of generating high-frequency ground motions, the accuracy and spatiotemporal variability will continue to lag behind the input model accuracy for source physics, crustal complexity, nonlinear site response, and other constituent elements of simulating earthquakes end-to-end. (4) To achieve the desired frequency resolution, the generated ground motions may be applicable only to a limited range of site conditions due to the application of minimum V_S threshold for wave propagation or the omission of non-linear effects. For instance, to achieve computational efficiency, some physics-based methods impose a V_S lower limit at 500 m/s , which means such methods cannot accurately simulate strong motions at soil sites (Pitarka et al., 2013).

Stochastic methods take a different approach to generating ground motions for engineering applications. The main advantages are that: (1) the source, site, and path scaling terms can be determined from empirical data and can simply be combined to generate the desired ground motions (Hanks et al., 1981; Boore, 1983; Boore, 2003), and (2) they are computationally efficient. At the same time, the main limitations of stochastic methods are (1) Such methods are based on the reconstruction of particular features, such as pseudo-spectral accelerations (PSA) (Tavakoli et al., 2005) or pseudo-spectral velocity (PSV) (Boore, 2003) without taking physics into account to fully characterize the ground motion. (2) Due to their inability to properly represent P, S, surface-wave modes, and polarization, stochastic methods cannot be used to produce coherent 3-component waveforms with multiple arrivals and coda that are representative of real ground motions (Graves et al., 2010). (3) They tend to underestimate the correlation between amplitudes at different frequencies compared to observed motions, which underestimates the variability in the building response and can have an impact on the calculated risk (Stafford, 2017; Bayless et al., 2019).

In both stochastic and physics-based models, a source of model uncertainty arises from the approximate representation of unresolved processes through parameterized functional forms. Model errors can be reduced to a certain degree through targeted data collection, leading –for example– to an improved spatial resolution of 3D velocity models in physics-based simulations, but they cannot be eliminated because there is no resolution separation between resolved and unresolved processes. At the same time, functional parameterization is often necessary for robust model extrapolation, especially outside the range of available data. However, for applications with dense data coverage and emphasis on interpolation (i.e., surrogate modeling of hybrid empirical-numerical strong motion datasets), machine learning models offer a fundamentally different approach to reduce model uncertainty: they parameterize the complex, nonlinear sub-resolution processes directly from the data that describe those processes using well-defined optimization rules (Gagne et al., 2020; Kong et al., 2019). With the growth of available high-quality strong ground motion data and the explosion of modern machine learning algorithms, generating realistic ground motions that span the broadband range of engineering interest ($\sim 0.1 - 25\text{Hz}$) is no longer a far-reaching goal.

The types of stochastic methods mentioned previously are not data-driven, (i.e., they are not generative models learned from data). However, major advances in machine learning have been made in the last decade on the topic of deep generative

models, in which the goal is to learn an unknown data distribution given a large dataset for training. Generative adversarial networks, or GANs (Goodfellow et al., 2014), are a class of generative models well suited for time-series synthesis. GANs draw samples from random distributions to generate new data that mimic an existing sample. Carefully designed GANs have uncovered and replicated the properties of time series in various fields such as finance, medicine, and seismology, including ground motion synthesis for earthquake early warning. Among others, Wang et al., 2019 and Wang et al. (2021b) generated realistic seismic signals for training of early warning systems; (Li et al., 2020a) also proposed a GAN-based method to artificially generate time series that resemble realistic seismic signals for early warning system training; Gatti et al., 2020 used GANs to extract high frequency features from ground motion records and use them to enhance ground motion simulated time series; Matinfar et al., 2023 used GANs to synthesize ground motion acceleration time series compatible with a given target design spectrum; Florez et al., 2022 used GAN to synthesize realistic three-component accelerograms conditioned on a set of continuous physical variables (magnitude, distance, and shear wave velocity) and statistically evaluated the quality of the signals using common engineering ground-motion intensity measures; while Esfahani et al., 2023, inspired by Florez et al., 2022, simulated non-stationary ground-motion recordings leveraging information contained in the frequency domain and incorporating physics-based knowledge such magnitude and distance. For a detailed overview of the use of GANs in earthquake engineering fields, the reader is referred to Marano et al., 2024.

In this paper, we use a new class of generative models – a Generative Adversarial Neural Operator (GANO) (Rahman et al., 2022) with the ultimate goal of building a framework that synthesizes ground motions for engineering applications. We first present the ground-motion synthesis algorithm and discuss its advantages compared to previous work. We next evaluate the GANO framework using noiseless ground motions simulated by the Southern California Earthquake Center (SCEC) Broadband Platform (BBP) (Maechling et al., 2015), and finally validate it by training it on three-component acceleration time histories from a strong-motion dataset processed from the Kik-Net network in Japan. In both applications, the generated ground motions are conditioned on moment magnitude (M), closest-point-on-the-rupture to site distance (R_{rup}). Additionally, for the BBP dataset the style of faulting (strike-slip or reverse event) was included as a conditional parameter, while for the Kik-Net dataset, the time average shear wave velocity at the top 30m (V_{S30}) and tectonic environment type (subduction or shallow crustal event) were the conditional variables. Lastly,

we assess the performance of our model through residual analysis and through a comparison with published GMMs. While still in progress, quantitative analyses show that our GANO-generated ground motion model can overall reproduce realistic ground motion time histories with a broad range of frequencies (0.1~30Hz) for horizontal and vertical components and can approximate the variability in the empirical Kik-net dataset.

2.2 Methods

Neural Operators

The classical development of neural networks has primarily focused on learning mappings between finite-dimensional Euclidean spaces or finite sets. Neural operators were introduced as a generalization of neural networks, intended to learn mappings between continuous function spaces from finite input-output pairs (Li et al., 2020b; Li et al., 2020c). Learning mappings between function spaces has shown great success at efficiently solving complex partial differential equations (PDEs). For instance, Pathak et al., 2022; Wen et al., 2022; Yang et al., 2021, have shown that neural operators can efficiently reproduce simulations governed by PDEs in problems of weather forecasting, CO2 injection, and wave propagation in solids to name a few.

Neural operators can be understood as mappings between infinite-dimensional function spaces. Suppose, for instance, that there is a non-linear mapping $\mathcal{F}:\mathcal{A} \rightarrow \mathcal{U}$, where $\mathcal{A}$ is the input and $\mathcal{U}$ is the output function space. The neural operator approximates the solution operator $\mathcal{F}$ using a fixed set of parameters ($\mathcal{F}_\theta \approx \mathcal{F}$) that provide a continuous representation of $\mathcal{U}$ in the function space. Kovachki et al., 2023 formally established that neural operator models with a fixed number of parameters satisfy discretization invariance; and further showed that they are universal approximators of continuous operators, namely, can uniformly approximate any continuous operator defined on a compact set of normed vector spaces. According to Kovachki et al., 2023, neural operators are the only known class of models that guarantee both discretization-invariance and universal approximation.

A computationally efficient variant of Neural Operators particularly suited for wave propagation (and synthesis) are Fourier Neural Operators (FNO) (Li et al., 2020b), whereby the representation and parameterization of the neural operator kernel are formulated in the Fourier space. Despite the advantages in terms of efficiency and accuracy, FNOs, which according to Kovachki et al., 2023 *achieved the best numer-*

ical performance of all the numerical experiments, are fully connected networks, and as such, the memory requirements associated with their training can be significant. Inspired by the U-net architecture (Ronneberger et al., 2015), Rahman et al., 2022 proposed an architecture referred to as the U-shaped Neural Operator (U-NO), which relies on gradually compressing and decoding information to alleviate issues associated with excess memory requirements. U-NO has the advantage of faster training and less memory requirements compared to standard FNO.

Generative Adversarial Neural Operators (GANO)

Generative Adversarial Neural Operators (GANO) generalize the Generative Adversarial Network (GAN) framework (Rahman et al., 2022), which is one of the most prominent paradigms in machine learning for analyzing unsupervised data. While GANs are successful generative models with a wide range of theoretical developments (Arjovsky et al., 2017), their empirical success has been only in finite-dimensional data regimes. In fact, there has been little progress in developing generative models for infinite-dimensional spaces and function spaces, which limited the use of GANs for learning functional data (i.e., data on irregular grids in space and time), which is often the case in the fields of science and engineering.

A GANO framework consists of a neural operator generator and a discriminator functional (i.e., a neural operator followed by an integral function). The generator uses Gaussian Random Fields (GRFs) or any predefined probability function as input and outputs a function sample. The discriminator accepts as input real or synthetic data and returns a scalar (score). The objective function, namely the score generated by the discriminator, is used for the loss function. Rahman et al., 2022 devised the GANO framework by i) generalizing the Wasserstein GAN setting (Arjovsky et al., 2017) to infinite dimension space, and ii) using the generalized Wasserstein distance and a gradient penalty as regularization terms to make the models more stable and less sensitive to the tuning of their hyperparameters.

Conditional Ground Motion GANO (cGM-GANO)

In this paper, we present a conditional GANO architecture designed to generate realistic 3-component acceleration time histories; we will refer to the architecture in the following sections as cGM-GANO. The architecture presented here has been trained on four conditional variables: the moment magnitude ($\mathbf{M}$), the closest-point-on-the-rupture to site distance (R_{rup}), the time-averaged shear wave velocity in the top 30m (V_{S30}), and the style of faulting or tectonic environment (f_{type}). f_{type} is

used to differentiate between strike-slip ($f_{type} = 1$) and reverse ($f_{type} = 0$) events for the BPP data set, or shallow crustal ($f_{type} = 1$) and subduction ($f_{type} = 0$) events for the Kik-net data set. Given a vector of the aforementioned parameters, it generates 3-component acceleration time series (Fig 2.1). We should note here that the selection of four variables is a modeling decision and could be replaced with any number of conditional variables, provided that there is a sufficiently large dataset for GANO to learn the effect of the different parameters.

To design cGM-GANO, we implemented the U-NO architecture in a GANO framework, as outlined in the previous section. cGM-GANO consists of a generator and discriminator trained in a min-max adversarial game (Fig. 2.2). We should note here that the training database of ground motions spans 3-4 orders of magnitude of PGA amplitude; as such, acceleration time series needed to be normalized to a common amplitude scale to ensure that there was no loss of information (Goodfellow, 2016; Wu et al., 2018) so that small events are not interpreted as noise. The normalization constants for each channel of the three component time series are also *learned* and recovered when the trained generator is called to synthesize a conditional ground motion scenario. The input to the trained generator consists of two sets of one-dimensional (1D) discretized functions: (i) a 1D Gaussian Random Field (Gaussian Process), which is stochastic (ii) and a set of constant functions that are used to convert the input parameters ($\mathbf{M}$, R_{rup} , V_{S30} , f_{type}) to a measurable function. The output includes the 3-component normalized ground motion waveforms and three normalization constants, in this case, the peak amplitudes associated with the normalized waveforms.

Compared to the previous work by our research group (Florez et al., 2022), the important advantage of GANOs that pertains to the problem of ground motion synthesis is their resolution and discretization invariance (Kovachki et al., 2023): the input function to the generator can be expressed with an arbitrary discretization (i.e., a dataset of time series sampled at various frequencies), yet the generated output is a function, which can be queried at any (i.e., frequency) resolution. These properties follow the recent advancements in operator learning that generalize neural networks (and thus GANs) that only operate on a fixed resolution (Kovachki et al., 2023); specifically in Florez et al., 2022, the input to the GAN was a sample from a finite-dimensional multivariate random variable and the output was a finite-dimensional object that could be queried only at a predetermined time-frequency resolution.

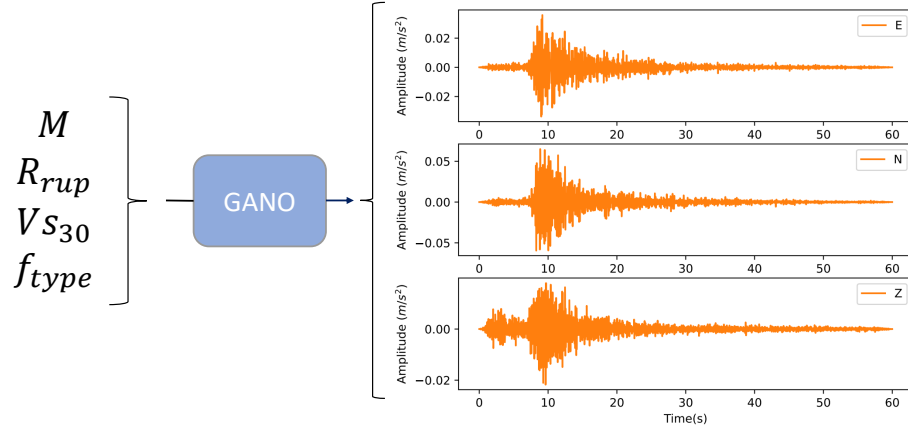


Figure 2.1: Schematic of GANO input-output structure used in this work: given a set of conditional variables (left), the trained GANO (middle) produces three-component acceleration time series (right).

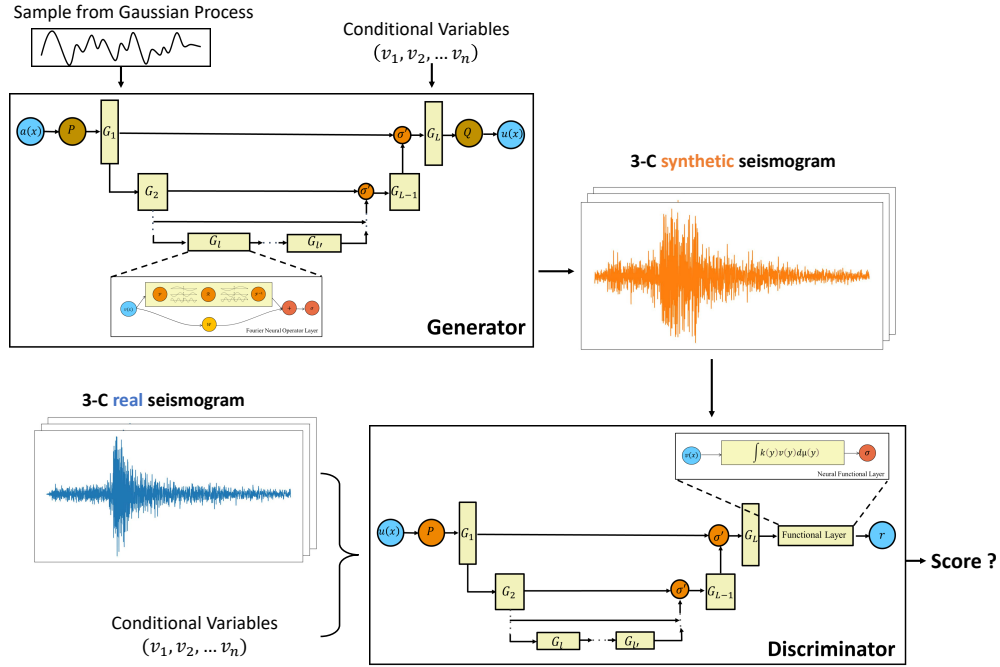


Figure 2.2: Schematic of cGM-GANO architecture, and generator - discriminator interaction. The input conditional variables are $\mathbf{M}$, R_{rup} , V_{S30} and f_{type} . The input function for the generator is finite duration Gaussian random field sample while the output is the normalized three-component acceleration time series and three PGAs. The input function of the discriminator are the real or synthetic three-component acceleration time series while the output is score evaluating the realism of the input ground motions.

Additionally, GANOs can learn probability measures on function spaces, in contrast to GANs, a difference that has been shown to avoid the mode collapse issues seen in GANs when the training data is generated from a mixture of GRFs (GANO reliably recovers the modes included in the mixture). For more information related to mode collapse and the strategies of hyperparameter tuning that have been developed to avoid it, as well as issues associated with training stability and mode entanglement of GANs, the reader is referred to Rahman et al., 2022; Wang et al., 2021a; Abdal et al., 2021.

In what follows, we present the cGM-GANO training strategy, as well as the verification framework using simulated (noise-free) ground motions, and the validation framework using strong motion recordings from the Japanese strong motion network Kik-Net and empirical ground motion models. By contrast to Florez et al., 2022, we here train GANO using velocity time series and then differentiate the generated ground motions to compute the target 3-component acceleration time series. Learning the acceleration time integral is equivalent to learning a linear filter of the target time series that balances the low and high frequencies. This effect is better described with (2.1) where for non-zero frequencies, the integral amplifies low-frequency components (for $\omega \leq 1$) and suppresses the high-frequency components (for $\omega \geq 1$). An example of the frequency content comparison between acceleration, velocity, and displacement is presented in the Appendix Fig A.1.

$$\int_0^t f(t) \overset{FT}{\leftrightarrow} \frac{F(\omega)}{j\omega} \quad (2.1)$$

Residual Analysis and Comparison to Empirical Ground Motion Models

The statistical evaluation of the generated ground motions was performed (i) by comparing the PSA and FAS, for BBP dataset, or EAS, for Kik-net dataset, ordinates of the observed and synthetically generated ground motions for specific scenario bins and (ii) by performing residual analysis of the PSA and FAS or EAS of the training and testing datasets, for both the horizontal and vertical directions. For the horizontal component, PSA was calculated as the rotation-independent 5%-damped response spectra (RotD50) for periods in the 0.01 – 10sec range (Boore et al., 2006; Boore, 2010; Boore, 2010) using a modified GPU parallel computation version of the Pyrotd (Kottke et al., n.d.) python library, while in the vertical direction PSA was calculated as 5%-damped response spectrum of the vertical component. The horizontal FAS was calculated as the power mean spectrum of the two horizontal

components:

$$A_h(f) = \sqrt{\frac{1}{2} (A_{h1}(f)^2 + A_{h2}(f)^2)} \quad (2.2)$$

where, $A_{h1}(f)$ and $A_{h2}(f)$ are the FAS of the two horizontal components at frequency f . The calculation of the horizontal EAS followed the recommendations of Kottke et al., 2021, smoothing the power-mean FAS through a Konno et al., 1998 smoothing window (Equation 2.3) with bandwidth, $b_w = 0.0333$. The vertical FAS was calculated directly from the vertical component, and the vertical EAS was smoothed with the Konno-Ohmachi filter.

$$b = 2\pi/b_w \quad (2.3)$$

$$W(f) = \left(\frac{\sin(b \ln(f/f_c))}{b \ln(f/f_c)} \right)^4 \quad (2.4)$$

The residual analysis was performed by comparing the observed intensity measures (IM) with the distribution of the synthetic ground motion IM , within the useable frequency range of the observed ground motions. The comparison is performed by: (i) generating 100 synthetic 3-component realizations with the same ($\mathbf{M}$, R_{rup} , V_{S30} , f_{type}) conditional variable values of the observed ground motion, (ii) computing the relevant IM from the observed and synthetic ground motions (IM^{observed} , $IM^{\text{cGM-GANO}}$, respectively), (iii) calculating the mean and standard deviation of $IM^{\text{cGM-GANO}}$ in log space, and (iv) computing the normalized model residual, ϵ , for each observation as:

$$\epsilon = \frac{\ln(IM^{\text{observed}}) - \text{mean}(\ln(IM^{\text{cGM-GANO}}))}{\text{std}(\ln(IM^{\text{cGM-GANO}}))} \quad (2.5)$$

this way the mean of ϵ quantifies bias and the standard deviation of ϵ the variability of the predictive model with respect to the data. A zero mean ϵ indicates that the ensemble of synthetic ground motions is unbiased with respect to the observations, and a unit ϵ standard deviation, or a two-unit width between the 16th and 84th ϵ percentile, suggests that the variability of the synthetic ground motion ensemble is consistent with the variability of the empirical dataset.

Since the BBP waveforms are considered noiseless, the entire frequency content of the ground motion was part of the useable frequency range. Kik-net ground motions, on the other hand, were filtered based on the noise-to-signal ratio (Bahrampour et

al., 2021). In this case, the useable frequency range is defined as:

$$\text{Lowest usable frequency (LUF)} = 1.25\max(HPF_{H1}, HPF_{H2}, HPF_V) \quad (2.6)$$

$$\text{Highest usable frequency (HUF)} = 0.8\min(LPF_{H1}, LPF_{H2}, LPF_V) \quad (2.7)$$

where HPF_{H1} , HPF_{H2} , and HPF_V are the corner frequencies of the high pass filter for the two horizontal and vertical component, while LPF_{H1} , LPF_{H2} , LPF_V are the corner frequencies for the low pass filter of the same components. The calculation of residuals was performed only within the usable frequency range to remove the effect of filtering

Lastly, we compared the scaling predicted by cGM-GANO against representative GMMs for the KiK-net dataset as part of our ongoing effort to develop a tool for efficient ground motion synthesis for engineering applications. Comparisons were performed against published PSA and EAS models such as the NGA-West2 GMMs (Abrahamson et al., 2014) and BA18 (Bayless et al., 2019) for the shallow crustal events and BChydro (Abrahamson et al., 2016) for the subduction events.

2.3 Training with Simulated Ground Motions: Verification

We first verified our cGM-GANO architecture by training it on simulated ground motion time series generated using the SCEC BBP, a collection of open-source scientific software modules that can simulate broadband (0 ~ 20Hz) ground motions for earthquakes at regional scales (Maechling et al., 2015).

The goal of this exercise was to verify that cGM-GANO is able to assimilate noiseless ground motion training data to reproduce the median scaling and aleatory variability of the simulations. Verifying the reasonableness of the simulated ground motions, however, was outside the scope of this study. This exercise was conducted first, before testing cGM-GANO on observed ground motions, as it offered a simpler learning environment. The BBP-generated ground motions do not require filtering and baseline correction, reducing the number of factors affecting the waveforms, and the uniform site conditions across the ensemble of simulated scenarios reduce the number of conditional variables that cGM-GANO was called to learn.

For this study, we used a dataset developed on BBP v17.3.0. following the Graves et al., 2015 simulation module (BBP-GP). This module is a hybrid approach that combines low-frequency (often referred to as deterministic) and high-frequency (often referred to as stochastic) motions computed with different methods into a single broadband response. The database was developed by Shi, 2019 in collaboration

with the BBP-GP developer, Robert Graves (pers. communication), and it contains 113 events ranging from M 5.5 to 7.2 on strike-slip and from M 7.0 to 7.2 on reverse faults, recorded at stations on reference site conditions ($V_{s30} = 760$ m/s), at rupture distances between 0 to 200 km. The database contains 6704 acceleration time histories: each record has 3-component 40s waveforms sampled at 100Hz. The magnitude-distance-PGA distribution of the data is shown in Fig A.2. A representative illustration of the fault and stations' locations is shown in Appendix Fig A.2

In this case, the cGM-GANO was trained conditioned on the moment magnitude (M), on the closest-point-on-the-rupture to site distance (R_{rup}), and on the fault type (f_{type} , strike-slip or reverse).

Fig. 2.4 compares the FAS and PSA spectra of BBP simulated and GANO-generated 3-component waveforms for M 7.0 strike-slip and reverse events at 10 km rupture distance. The comparison of the FAS and PSA ordinates is encouraging, especially in the long period, short frequency regime, where both the mean and standard deviation are in agreement. In particular, one can observe that the standard deviation of the two data sets converges for frequencies smaller than $f < 1Hz$ ($T > 1sec$), while for higher frequencies, the cGM-GANO-generated time series have systematically higher variability. Our working hypothesis is that the discontinuity occurs at $f = 1Hz$, the cross-over frequency between the low- and high-frequency methods in the BBP-GP hybrid module, because cGM-GANO cannot learn the stochastic high-frequency method and resorts to carrying over the learned physics-based solution of $f < 1Hz$ into the higher frequency range. One reason for this might be that GANO can only learn distributions over functions that are square-integrable, and the stochastic generation method used in the BBP does not necessarily satisfy this property. Representative waveforms from both the BPP and cGM-GANO are also depicted in the Appendix (Fig A.3), showing that the synthetic time series manifest realistic features, such as P-, S-, and coda wave arrivals, as well as time history envelopes consistent with the simulated ground motions.

Next, we test the capabilities of cGM-GANO to learn the magnitude and distance scaling in sparsely sampled regions of the training dataset. Fig. 2.5 shows that the BBP training database by Shi, 2019 has no reverse event records (here, depicted by the dot markers) for small magnitude at distances 20 – 40km, or for magnitudes M 6.9–7.1 at large distances ($\geq 50km$). Still, the cGM-GANO was trained using data from both reverse and strike-slip events, each appropriately tagged. The example

in the figure suggests that the model has, in part, inferred the median magnitude and distance scaling of the reverse style faulting from the data ensemble training, transferring the information across faulting styles. At the same time, however, the variability of cGM-GANO is larger than the BBP data in the densely sampled regions of the training dataset (for example, $M6.9 - 7.1$ far-field strike-slip as well as near-field reverse) where one would expect the variability to be best captured.

Lastly, we examined the degree to which cGM-GANO can recover the frequency domain characteristics of the BBP simulated ground motions through a residual analysis. Results for the horizontal and vertical components of the EAS and RotD50 residuals are shown in Fig 2.6, separately for the strike-slip and the reverse simulated events. The cGM-GANO results for the horizontal EAS components for both types of ruptures show discontinuities at 1Hz. The mean and aleatory variability are captured reasonably well for both strike-slip and reverse events in the low-frequency range ($f \leq 1\text{Hz}$). In the high-frequency range ($f > 1\text{Hz}$), the mean is captured reasonably well, but the aleatory variability is underestimated for strike-slip events, and there is a noticeable bias in the mean and underestimation of the variability for reverse events. The RotD50 spectral accelerations show a slight constant positive bias across the period range for both styles of faulting. The difference between EAS and RotD50 residual distributions is attributed to the unsuitable inter-frequency correlation of the synthetic ground motions that is influenced by the stochastic component of the training dataset (Appendix Fig A.4); further investigation is needed, nonetheless, to find the exact cause of the mismatch between the bias and variability of EAS and PSA in the various frequency-period ranges.

Our working hypothesis is that verification of cGM-GANO using BBP simulations, which are synthesized by two separate processes, is compromised when the neural operator attempts to recover the entire frequency range as if it were the solution of the wave equation for propagation in a continuum. Whether the identified biases in the aforementioned figures stem from the hybrid BBP-GP method and the difficulties associated with training neural operators on stochastic data; or it is evidence that more training data is necessary to reliably recover the median and standard deviation scaling relationships across the entire dataset, is still being debated. More extensive validation is currently being carried out by the authors to confirm this claim, including the simulation of large-magnitude, near-field reverse faulting events to confirm the aforementioned recovered scaling. These results, along with the foregoing sections, however, point to the possibility of using cGM-GANO in the future

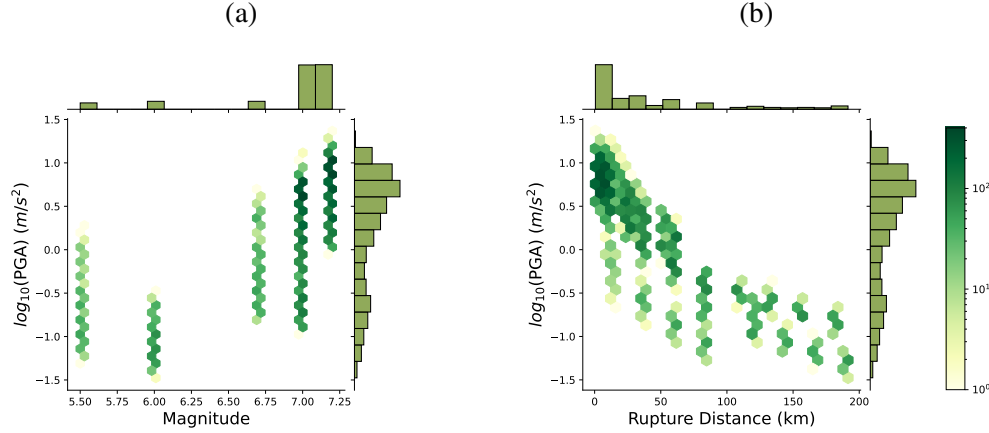


Figure 2.3: Data distribution of BBP dataset. Subfigures (a) and (b) show crossplot between $\log_{10}(\text{PGA})$ and magnitude and rupture distance, respectively

to fill sparsely sampled regions of observational datasets used to develop GMMs with defensible, realistic, 3-component ground motion time series appropriate for engineering applications.

2.4 Training with Recorded Ground Motions: Verification and Validation

In this section, we evaluated the performance of cGM-GANO by training it on a ground motion dataset compiled from the Japanese Strong Motion Network KiK-net (National Research Institute for Earth Science and Disaster Resilience, 2019). The compiled KiK-net dataset included 42481 strong ground motions sampled at 100Hz from 1985 events, 620 shallow crustal and 1365 subduction events, collected over a period of 20 years (from 1997 to 2017) at 613 stations. For the data preprocessing, we used the method described in Bahrampouri et al., 2021, which comprises of the following steps: (i) identify the filter corner frequencies for each record based on noise windows, and (ii) detect the P- and S-wave arrivals. We ensured that each time history contained the ground motion from a single event, and we used a fixed duration of 60sec for training, starting from the onset of the P wave. The choice of a fixed duration window stems from FNO requirements that we used in our architecture. Prior to training, we detrended and baseline-corrected the KiK-net ground motions to ensure that the displacement, velocity, and acceleration are zero at the end of the 60sec. The same baseline correction was applied on the cGM-GANO generated ground motion time series before the validation tests described below.

The conditional parameters for the KiK-net dataset ($\mathbf{M}$, R_{rup} , V_{S30} , and f_{type}) were

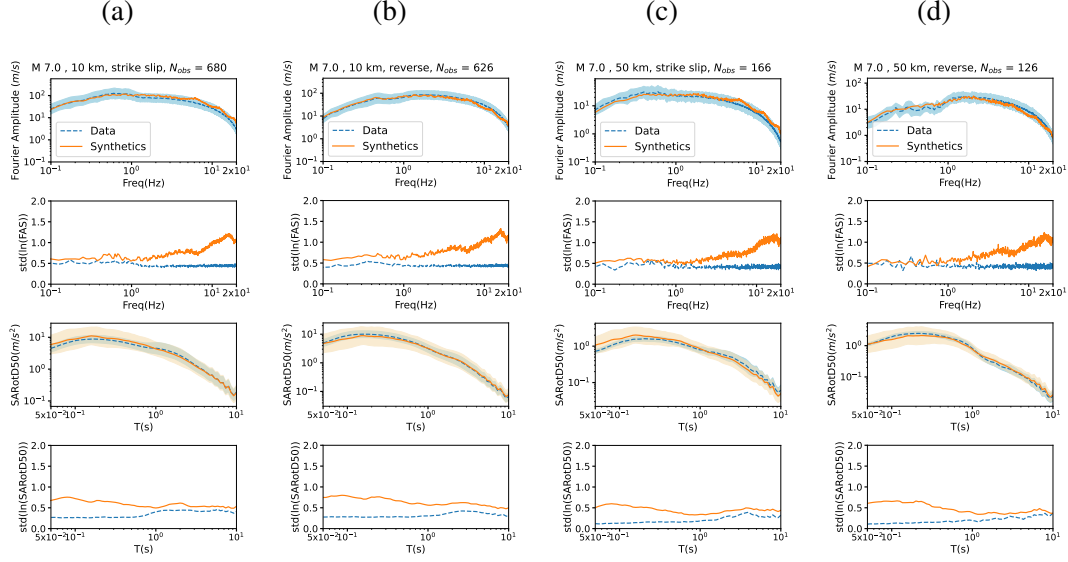


Figure 2.4: Scenario event comparisons for cGM-GANO trained on the BBP dataset. Each vertical sub-panel depicts the Fourier Amplitude Spectrum (FAS), aleatory standard deviation of FAS, Rotd50 Pseudo-spectral acceleration (PSA) spectrum, and aleatory standard deviation of Rotd50 PSA. The title of each plot corresponds to the midpoint of each bin, and N_{obs} is the number of observed records within each bin. The lines represent the mean in log space of the horizontal components of the synthetic (solid line) or BBP (dashed line) data, while the shaded zone represents the 16th to 84th percentile range.

obtained from Bahrampouri et al., 2021; M ranges from 4.5 to 8.0, R_{rup} from 0 to 300 km, V_{S30} from 100 to 1100 m/s, and the tectonic environment is subduction or shallow crustal. Figure 2.7 shows the magnitude, distance, and V_{S30} distribution of the compiled data, which is primarily comprised of small-to-moderate events recorded at distances $R_{rup} \geq 50$ km, on moderate stiffness sites. Only 1.6% of the dataset ground motions corresponded to events with $M \geq 7.0$, 2.8% of records have rupture distances less than $R_{rup} \leq 50$ km, and 2.7% of the ground round motions were recorded on stations with $V_{S30} < 200$ m/sec.

For training purposes, the KiK-net dataset was randomly divided into 90% training data and 10% testing data, not used in training. In what follows, we present verification results against the entire database, while validation results based on the testing dataset alone are provided in the Appendix (Fig A.5). Both the entire and testing datasets show similar trends, indicating no overfitting on the training dataset. It was decided to include the comparison with the entire dataset in the manuscript for more robust magnitude, rupture distance, V_{S30} scaling, and individual scenario

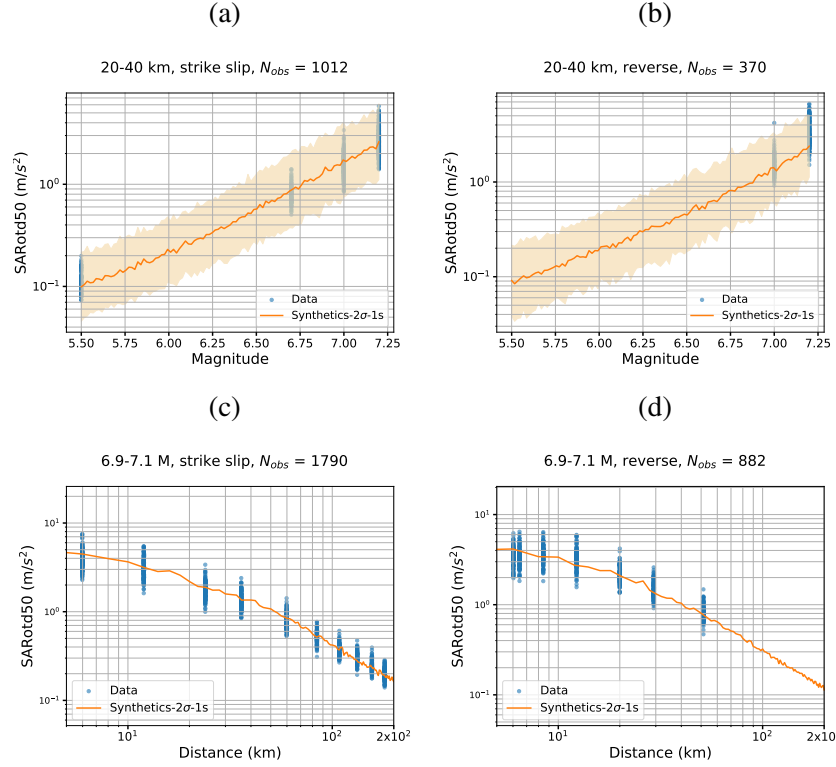


Figure 2.5: Comparison of cGM-GANO magnitude and distance scaling with BBP dataset for PSA RotD50($T=1$ s). Solid lines represent the mean scaling in log space. The shaded region shows the 2nd to 98th percentile of aleatory variability. Solid dots correspond to the training data.

comparisons.

Fig 2.8 shows a visual comparison between randomly sampled observed time series for different $\mathbf{M}$, R_{rup} , V_{S30} combinations and the corresponding cGM-GANO random realizations for the same scenarios. This comparison is intended to show that the generated waveforms are comparable in amplitude, duration and phase arrival with the recorded ground motions.

Fig. 2.9 compares the frequency domain statistics of the observed and cGM-GANO generated time-series belonging to the same $\mathbf{M}$, R_{rup} , V_{S30} bin. Although neural operators map function spaces and by definition have continuous input spaces (i.e., conditional variables), for the validation we binned the data in windows of increment $\mathbf{M} = 0.2$, $R_{rup} = 20\text{km}$ and $V_{S30} = 100$ m/s to ensure that the number of records in each bin would provide statistically significant results for comparison purposes. We successively generated cGM-GANO time-series for each scenario in each bin to create a database of synthetic ground motions for the same scenarios

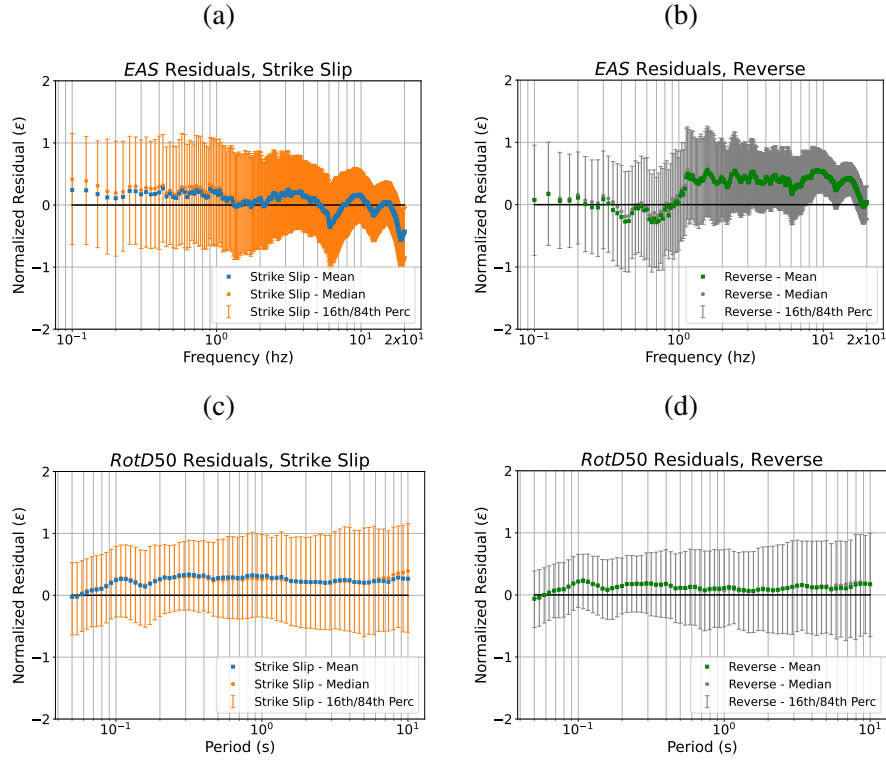


Figure 2.6: Residual plots of the BBP trained model. Subfigures (a) and (b) show the EAS residuals for strike-slip and reverse faults, respectively. Subfigures (c) and (d) show the RotD50 PSA residuals for strike-slip and reverse faults, respectively.

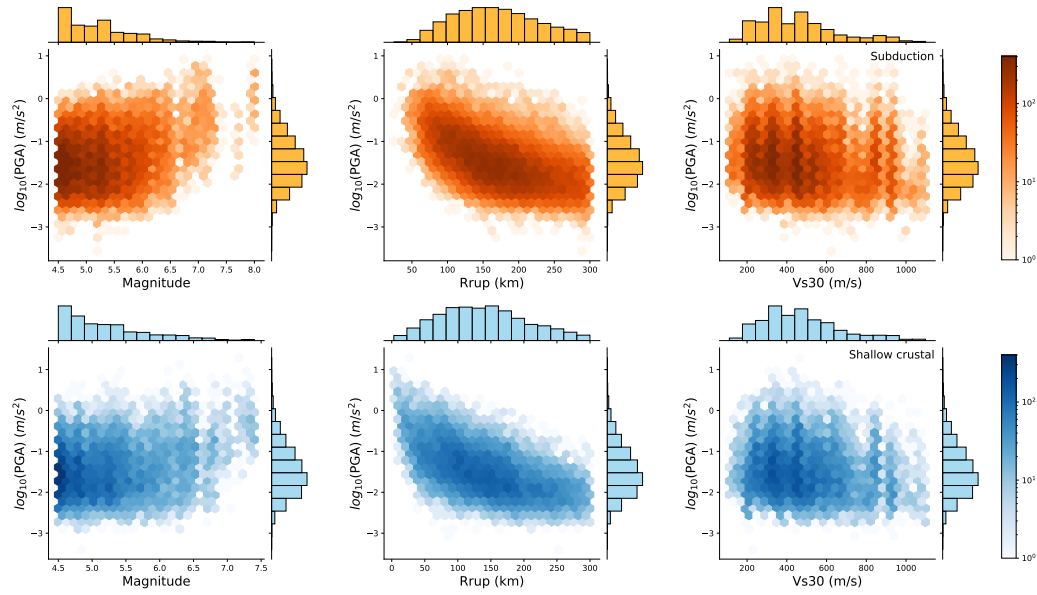


Figure 2.7: Distribution of earthquake magnitude, rupture distance, and V_{s30} versus $\log_{10}(\text{PGA})$ for the subduction (first row) and shallow crustal (second row) events.

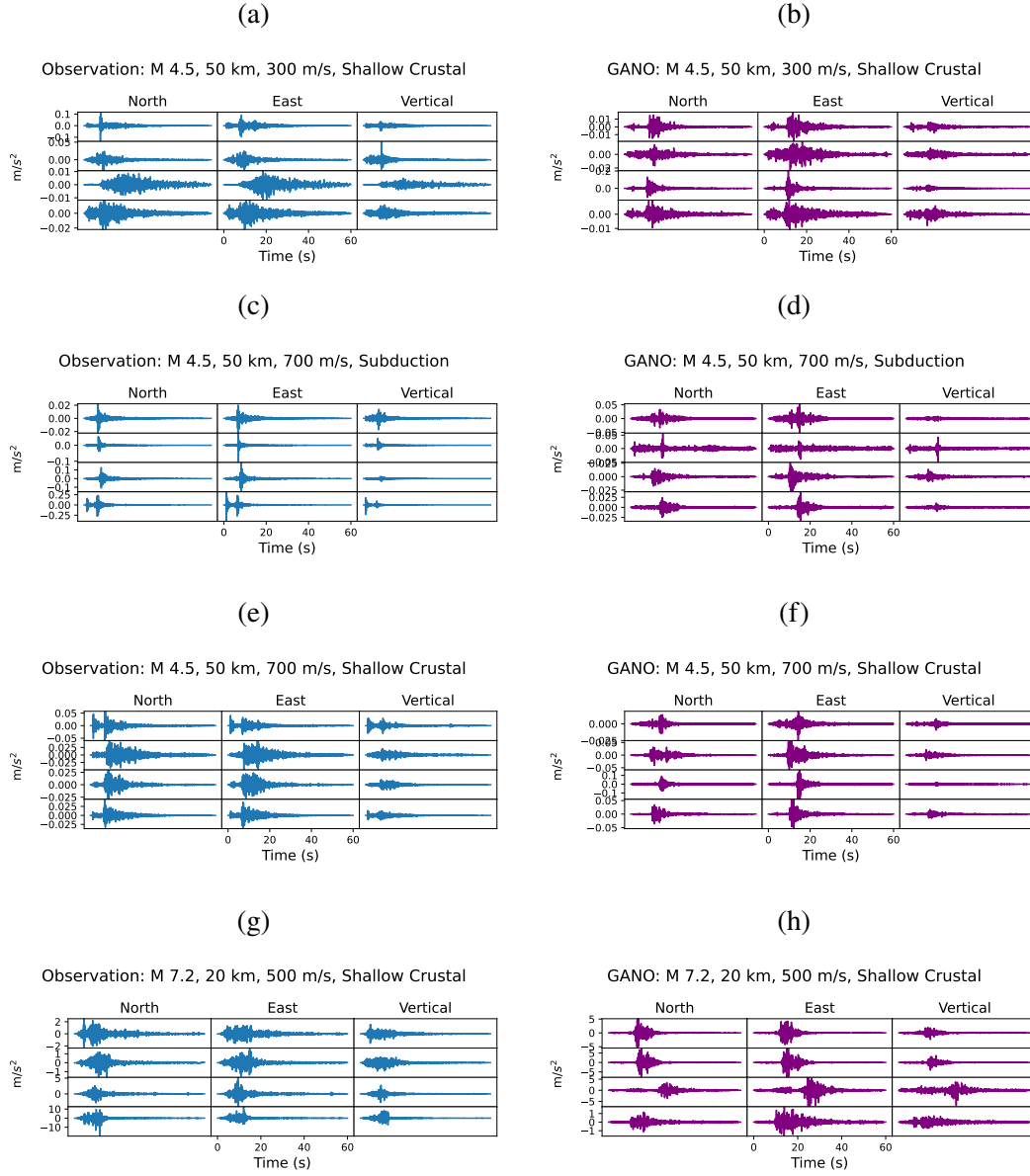


Figure 2.8: Accelerograms comparisons real and synthetically generated ground motions for cGM-GANO trained on the Kik-net dataset.

as the observations in each bin. The statistics of the two datasets are compared for representative bins in Fig. 2.9. For the examples shown, cGM-GANO recovers the mean and aleatory variability of the FAS in the frequency range between ~ 0.5 Hz and 30 Hz. Similarly, cGM-GANO recovers the mean and aleatory variability of PSA RotD50 in most cases; exceptions such as scenario (e) for periods shorter than $T = 0.2\text{sec}$, or scenario (f) for periods longer than $T = 1\text{sec}$ which may stem from the small number of observations available in those bins (21 and 17 observations, respectively) and will be further investigated by the authors.

We successively examined the extent to which cGM-GANO can recover the $\mathbf{M}$, R_{rup} , V_{S30} scaling of the ground motion observations for a given tectonic environment by means of residual analysis (for more information, see Section on Residual Analysis above). Results for the horizontal and vertical components of the EAS and RotD50 residuals are shown in Fig A.5, separately for the subduction and the shallow crustal events. The cGM-GANO results of the horizontal EAS components for both tectonic environments are unbiased in the frequency range $\sim 0.3 - 20\text{Hz}$. The vertical components show similar behavior with a small increasing bias starting at 0.5Hz . The horizontal spectral accelerations are well captured in the $\sim 1 - 10\text{sec}$ period range and show small, nearly constant positive bias for shorter periods $T \leq 1\text{sec}$. The vertical spectral accelerations for both tectonic environments are well captured in the $T \geq 2\text{sec}$ period range, while for shorter periods, cGM-GANO shows significant bias. While encouraging, these results leave open questions about the frequency range where cGM-GANO can reliably recover the mean and aleatory variability of recorded ground motions and the factors that affect the reliability of the GANO model, which are further investigated by the authors.

The extent to which cGM-GANO recovers the ground motion scaling versus $\mathbf{M}$, R_{rup} , V_{S30} is presented in Figs 2.11 and 2.12. The horizontal and vertical FAS residuals in subfigures 2.11 (a) and (c), and horizontal and vertical PSA residuals in subfigures 2.12 (a) and (c) for $f = 5\text{Hz}$ ($T = 0.2\text{s}$) suggest that cGM-GANO can recover the median magnitude scaling of the ground motions. By contrast, cGM-GANO shows systematic positive bias when called to evaluate the median scaling against rupture distance, for $R_{rup} \leq 20\text{ km}$ (see Figs 2.11 and 2.12 (b) and (e)). Our working hypothesis is that the bias reflects the very small number of near-field observations (0.4% of the entire training dataset recorded at distance less than 20km), which may not be enough for the model to learn the ground motion characteristics in this regime. The cGM-GANO generated ground motion scaling

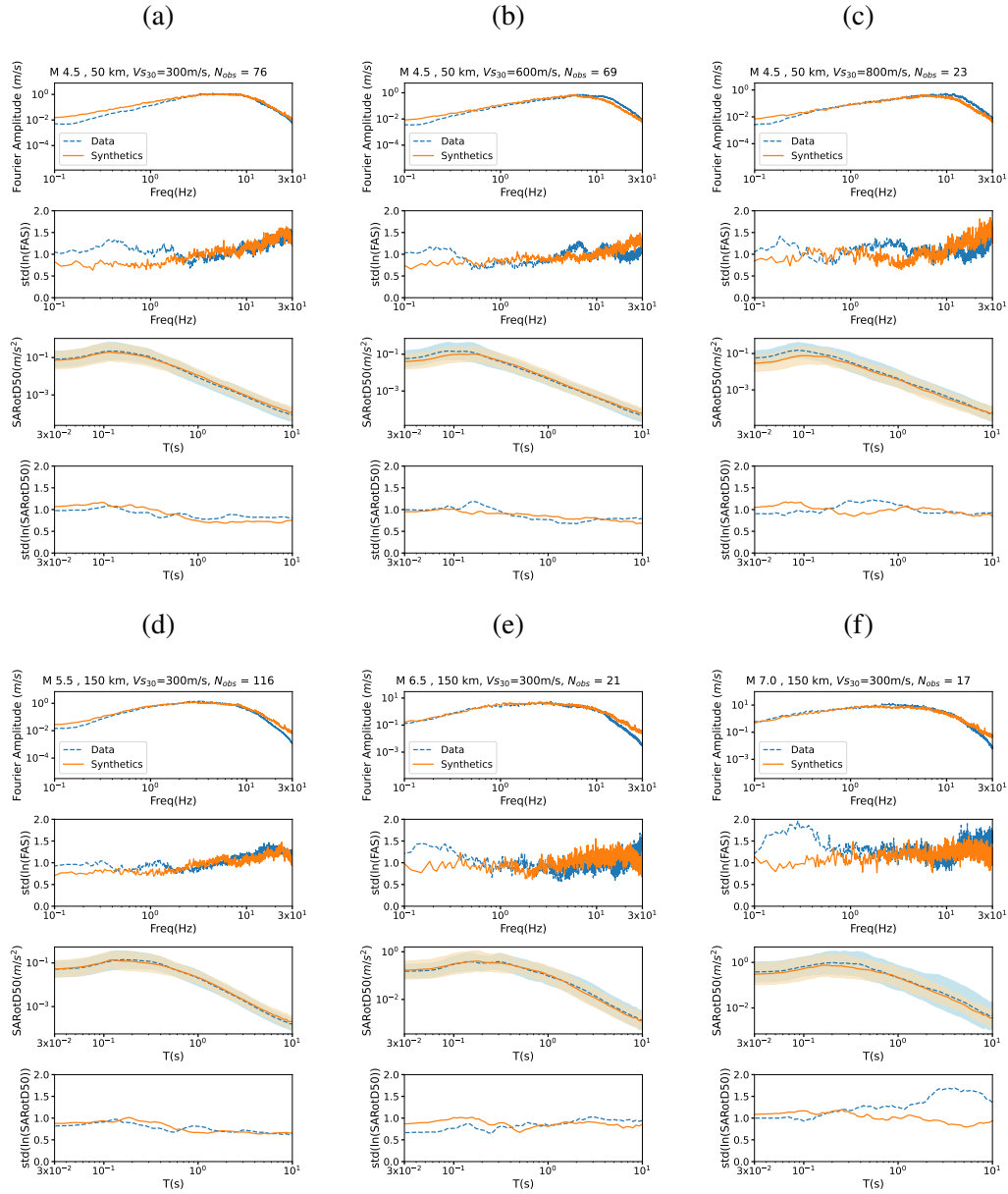


Figure 2.9: Scenario event comparisons for cGM-GANO trained on the kik-net dataset. Each sub-panel depicts the Fourier Amplitude Spectrum (FAS), aleatory standard deviation of FAS, Rotd50 Pseudo-spectral acceleration (PSA) spectrum, and aleatory standard deviation of Rotd50 PSA. The title of each plot corresponds to the midpoint of each bin, and N_{obs} is the number of observed records within each bin. The lines represent the mean in log space of the horizontal components of the synthetic (solid line) or kik-net (dashed line) data, while the shaded zone represents the 16th to 84th percentile range.

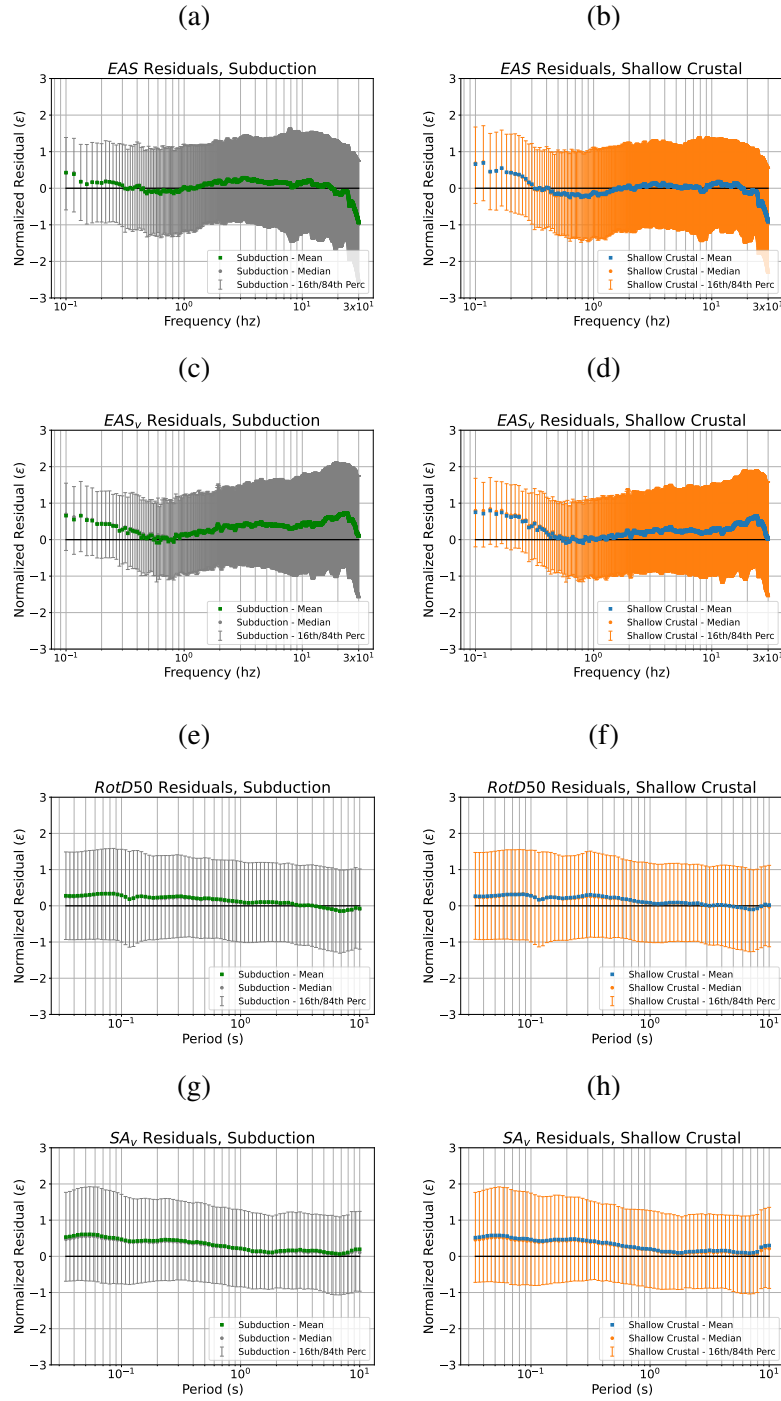


Figure 2.10: Residual plots of the kik-net trained model. Subfigures (a) and (b) show the horizontal EAS residuals for subduction and shallow crustal events, respectively. Subfigures (c) and (d) show the vertical EAS residuals for subduction and shallow crustal events, respectively. Subfigures (e) and (f) show the horizontal RotD50 PSA residuals for subduction and shallow crustal events, respectively. Subfigures (g) and (h) show the vertical PSA residuals for subduction and shallow crustal events, respectively.

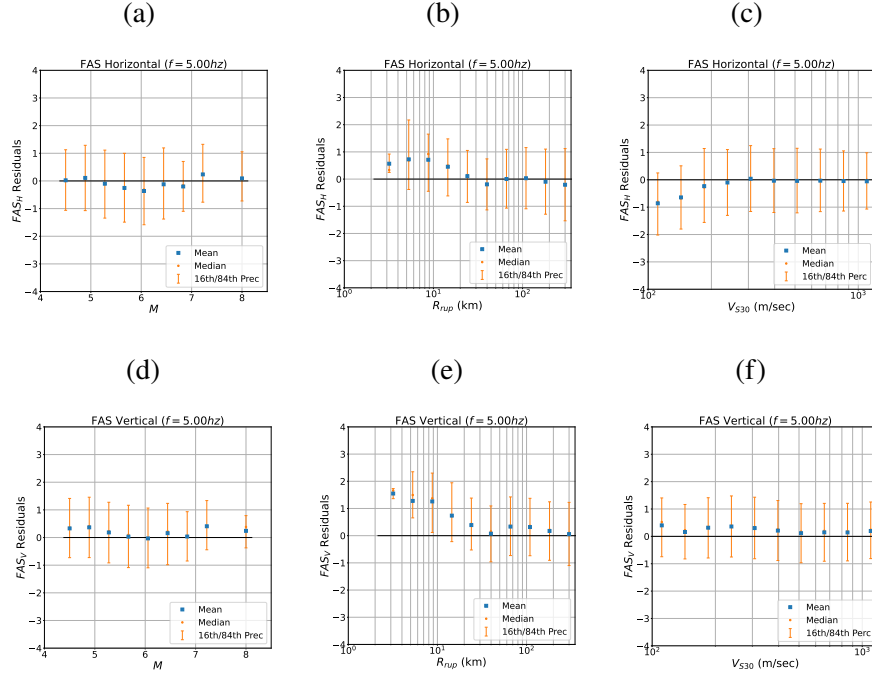


Figure 2.11: Comparison of horizontal and vertical FAS residuals for $f=5.0$ Hz against magnitude (subfigure (a) and (d)), R_{rup} (subfigure (b) and (e)), and V_{S30} (subfigure (c) and (f))

with V_{S30} recovered in Figs 2.11 and 2.12 (c) and (f) shows negative bias for both FAS and RotD50 versus V_{S30} for sites with $V_{S30} \leq 200$ m/s. The bias in this V_{S30} range could be associated with the limited number of ground motion observations for soft soil sites within the KiK-net strong motion network (2.7% of the entire training dataset was recorded at sites with $V_{S30} \leq 200$ m/s.) Although this is not a negligible dataset of training ground motions, we hypothesize that training cGM-GANO to generate strong motions at soft sites would be associated with learning ground motion characteristics controlled by nonlinear effects, which are likely less pronounced for stiffer sites and thus less likely to be learned from the remainder of the dataset. This hypothesis is supported by the fact that the vertical component scaling with V_{S30} shows a small, nearly constant positive bias across the same V_{S30} range. The capability of cGM-GANO to learn the effects of nonlinear site response, given the appropriate size of training data and conditional variables, is the focus of a separate study undertaken by the authors. Considered strategies for this objective include augmenting the dataset with synthetic datasets that include non-linear effects and employing a weighting sampling, increasing the frequency of soft sites being sampled during training.

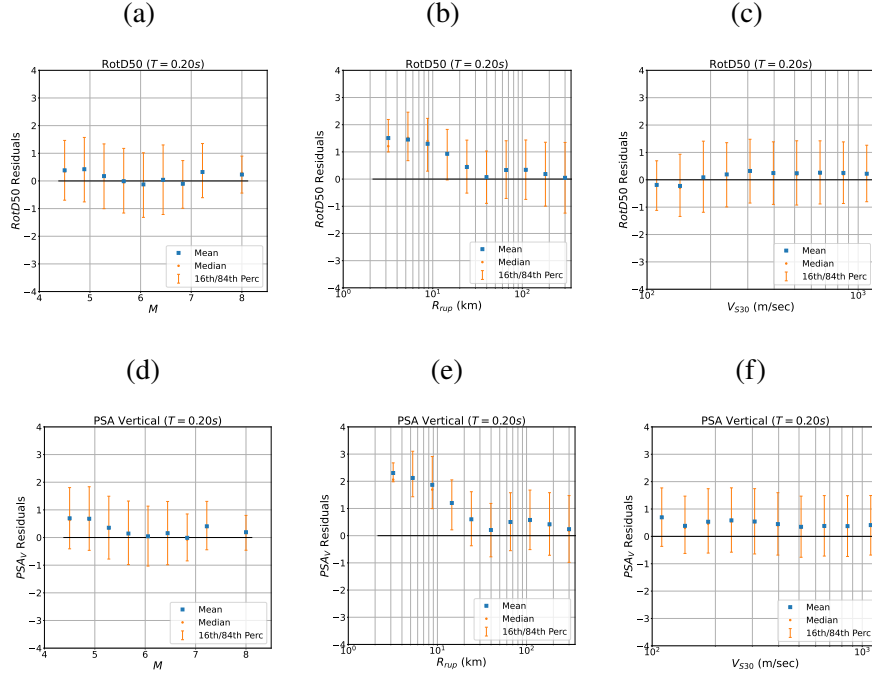


Figure 2.12: Comparison of horizontal and vertical PSA residuals for $T=0.2$ s ($f=5.0$ Hz) against magnitude (subfigure (a) and (d)), R_{rup} (subfigure (b) and (e)), and V_{S30} (subfigure (c) and (f))

We also evaluated the inter-frequency correlation of the cGM-GANO-generated ground motion time series. Fig 2.13 shows the correlation of the RotD50 residuals between $T=1$ s and $T=0.2$ s or $T=2.0$ s for the entire KiK-net dataset. The Pearson correlation coefficient of residuals (ρ) at the period pairs T_1 and T_2 is also calculated as follows:

$$\rho_{\epsilon_{T_1}, \epsilon_{T_2}} = \frac{cov(\epsilon_{T_1}, \epsilon_{T_2})}{\sigma_{\epsilon_{T_1}} \sigma_{\epsilon_{T_2}}} \quad (2.8)$$

where, $cov(\epsilon_{T_1}, \epsilon_{T_2})$ is the covariance between the two periods, and $\sigma_{\epsilon_{T_1}}$ and $\sigma_{\epsilon_{T_2}}$ are the standard deviations at T_1 and T_2 , respectively, as determined empirically from the synthetic ground motions. In this example, we see a stronger correlation between the $T=1.0$ s and $T=2.0$ s residuals ($\rho=0.78$) than the $T=1.0$ s and $T=0.2$ s residuals ($\rho=0.29$), which are further apart in log space. This observation is in agreement with Baker, 2011, who found the correlation coefficient between these pairs to be 0.75 and 0.44, respectively, for shallow crustal events. This example suggests that cGM-GANO is capable of learning the intricate characteristics of ground motion databases, such as the RotD50 inter-period correlation, that can be important in the

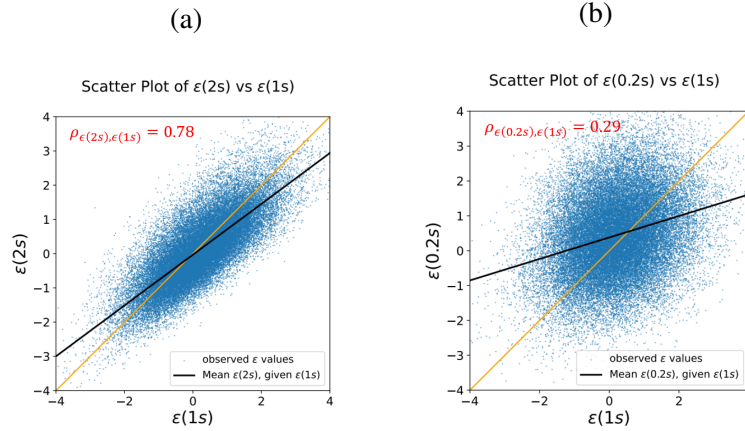


Figure 2.13: Inter-period correlation of horizontal PSA residuals. Between $T = 1$ s and $T = 2$ s in subfigure (a), and between $T = 1$ s and $T = 0.2$ s in subfigure (b).

seismic response of multi-modal structures.

Next, as we did with the verification SCEC BBP dataset, we tested the capability of cGM-GANO to learn the magnitude and distance scaling in sparsely sampled regions of the training dataset. In Fig 2.14, we show scenarios of RotD50 scaling versus magnitude, distance, and V_{S30} generated by cGM-GANO. The cGM-GANO-generated ground motions generally follow the mean and aleatory variability of the observations. In particular, they are able to capture the non-linear distance scaling at periods $T = 0.1, 1.0, 10.0s$, and the magnitude and V_{S30} scaling at periods $T = 1.0s$.

Validation against empirical GMMs

Lastly, we performed a series of comparisons of the cGM-GANO generated ground motions, trained on the KiK-net data set, to selected published ground motion models (GMMs) described in Section Residual Analysis and Comparison to Empirical Ground Motion Models. Fig. 2.15 compares the PSA magnitude and distance scaling of the cGM-GANO generated ground motions for shallow crustal events with the NGA-West 2 GMMs. Overall, cGM-GANO magnitude and distance scaling show similar trends with the published models in terms of median ground motion and aleatory variability. The biggest difference is at short distances which as explained above is believed to be a result of sparse data to constrain the model over that distance range. Similar comparisons are made in Fig. 2.16 against the BA18 EAS GMM. Here, a systematic misfit is observed at $f < 1$ Hz, which is introduced by the misfit between BA18 and the training data set; cGM-GANO scaling is consistent with the training dataset. Otherwise, similar conclusions are drawn for the median

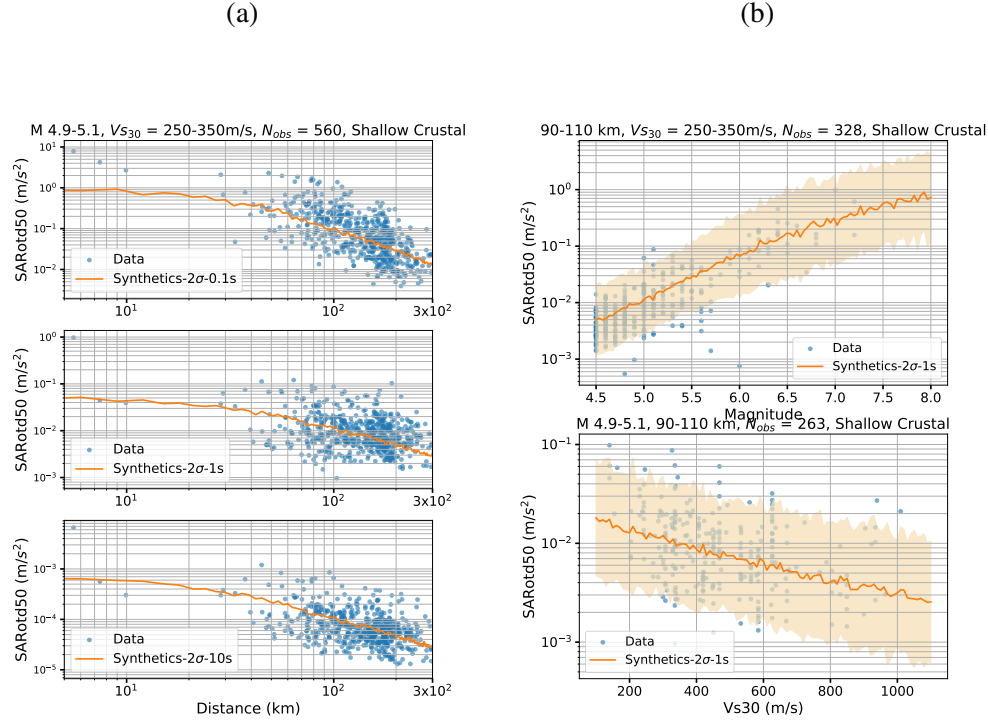


Figure 2.14: Comparison PSA distance, magnitude, and V_{S30} scaling of cmGM-GANO with Kik-net dataset. Left panels, from top to bottom, show the distance scaling for $T = 0.1, 1.0$, and 10 s. Top right panel shows the magnitude scaling for $T = 1.0$ s. Bottom right panel shows the V_{S30} scaling for $T = 1.0$ s. Solid lines represent the mean scaling in log space. The shaded region shows the 2^{nd} to 98^{th} percentile of aleatory variability. Solid dots correspond to the kik-net data.

magnitude and distance scaling. The underestimation of EAS aleatory variability by cGM-GANO compared to BA18 and the cause of the misfit between the BA18 and the data will be further investigated by the authors.

For the subduction events, PSA scaling is compared in Fig. 2.17. The synthetic ground motions have consistent scaling with BCHydro both in terms of median ground motion and aleatory variability over the entire frequency range. The main misfit is observed at short distances. cGM-GANO is believed to have better performance for subduction events compared to shallow crustal due to the larger dataset size.

2.5 Summary and Discussion

We presented a data-driven model for conditional ground motion synthesis that can be used to generate 3-component ground motion time series covering a broadband

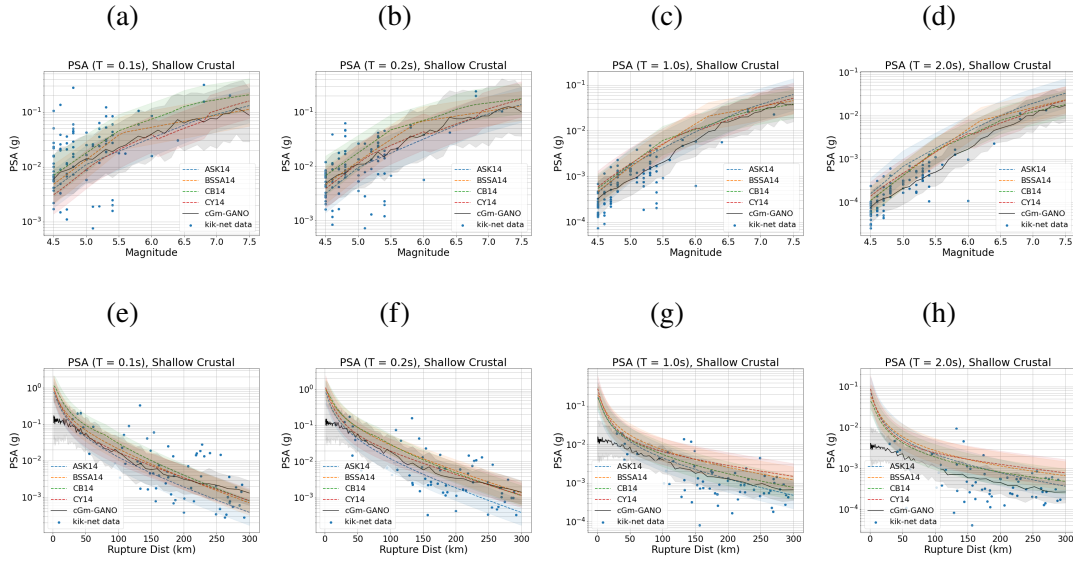


Figure 2.15: Comparison of cGM-GANO PSA scaling for shallow crustal events against the NGA-West2 GMMs. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $T = 0.1, 0.2, 1, 2$ s ($f = 10, 5, 1, 0.5$ Hz), respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16th to 84th percentile.

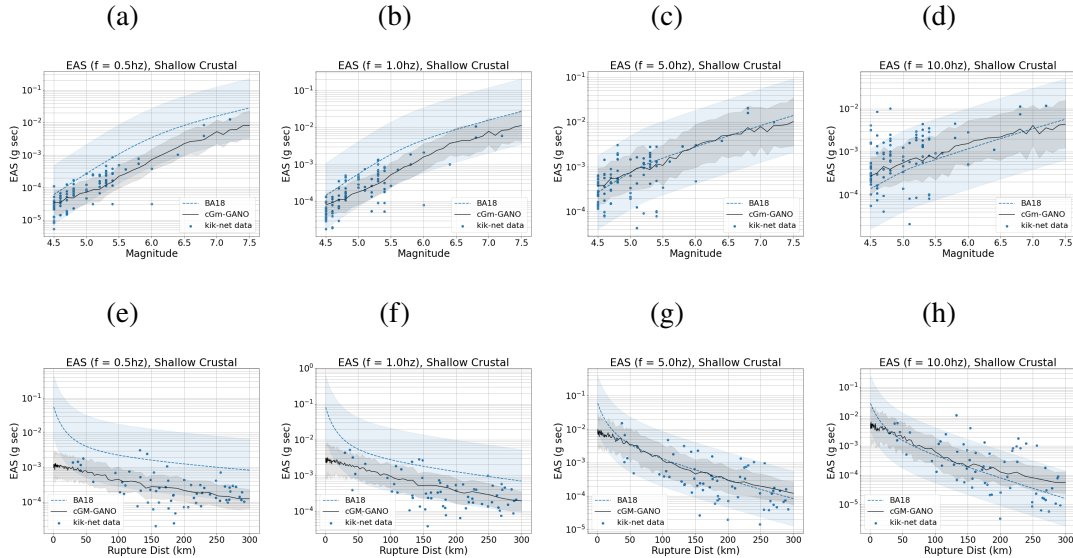


Figure 2.16: Comparison of cGM-GANO EAS scaling for shallow crustal events against the BA18 EAS GMMs. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $f = 0.5, 1, 5$, and 10 Hz, respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16th to 84th percentile.

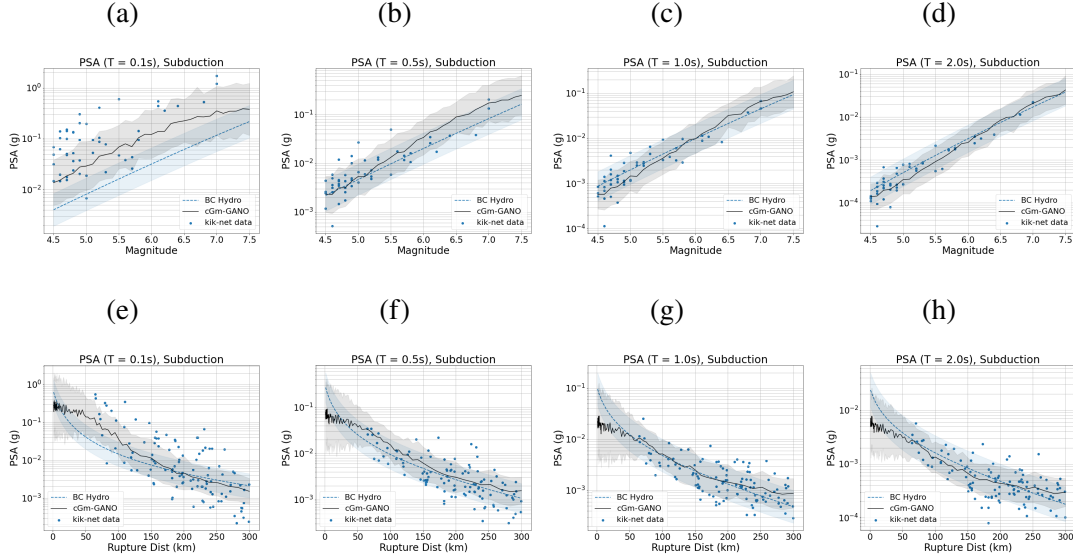


Figure 2.17: Comparison of cGM-GANO PSA scaling for subduction events against BC-Hydro GMM. Top panel shows the magnitude comparison for $R_{rup} = 60$ km and $V_{s30} = 700$ m/s, at $T = 0.1, 0.5, 1,$ and 2 s ($f = 10, 2, 1,$ and 0.5 Hz), respectively. Bottom panel shows the distance comparison for $M = 6$ km and $V_{s30} = 700$ m/s, km, for the same frequencies. Solid lines represent the median scaling, the shaded region corresponds to the 16th to 84th percentile.

range of frequencies. The model is based on the architecture of Neural Operators, a resolution-invariant generalization of Neural Networks. The ground motion synthesis is conditioned on moment magnitude (M), rupture distance (R_{rup}), time-average shear-wave velocity at the top 30m of the crust (V_{s30}), and tectonic environment or style of faulting (f_{type}).

Technical advancements of this work compared to previous efforts, including of the authors', are the use of Neural Operators and training of the model on velocity time series. Using Neural Operators guarantees both discretization-invariance of the model from the resolution of the training data set and universal approximation. We specifically employed the U-shaped Neural Operator (U-NO) architecture, which alleviates some issues associated with the memory requirements of fully connected Neural Operators. As well as, the training on velocity time series (as opposed to the target acceleration ground motions) also led to performance improvements due to the flatter Fourier spectral shape of the training signals.

The model was verified against a dataset of simulated ground motion time series from the SCEC BBP. The comparison of scenarios in terms of Fourier amplitude in the low-frequency regime $f < 1\text{Hz}$ ($T > 1$ sec) systematically showed the mean and

standard deviation of the simulated and cGM-GANO-generated time series to be in agreement. At the same time, for higher frequencies, the cGM-GANO-generated time series of the same scenarios have systematically higher variability and a positive bias for reverse events. Our working hypothesis for the discontinuity that occurs at $f = 1\text{Hz}$, the cross-over period between the low- and high-frequency methods in the BBP hybrid GP module, is that cGM-GANO cannot learn the stochastic high-frequency method, and resorts to carrying over the learned PDE solution of $f < 1\text{Hz}$ into the higher frequency range. The comparison of the response spectral ordinates showed a small constant positive bias across the period range for both styles of faulting. This bias is attributed to an unsuitable inter-period correlation of the synthetic ground motions that are influenced by the stochastic component of the training dataset; however, further investigation is needed.

The model was next trained and compared to a dataset curated and processed from observations recorded by the Japanese strong motion network KiK-net, which contains both shallow crustal and subduction events. Bias was observed in the distance scaling of near field ground motions and V_{S30} scaling for soft soil sites, which we attributed to a lack of adequate training data in the corresponding regimes: in the first case, the training dataset contains less than 0.5% of ground motions recorded at stations with $R_{rup} \leq 20\text{ km}$; in the second case, the training dataset was larger but we hypothesize that cGM-GANO is also called to learn the scaling that includes nonlinear site response, and as such, requires an even larger amount of soft site data. This last hypothesis is being tested by the authors as part of a separate project.

Comparisons with traditional ground motion models (GMMs) led to similar conclusions: the cGM-GANO generated ground motions showed consistent median scaling with the GMMs for the corresponding tectonic environments. The largest misfit was observed at short distances due to the scarcity of training data, as mentioned above. A systematic misfit in EAS in the low-frequency regime ($f < 1\text{Hz}$) between cGM-GANO and BA18 is likely caused by the inconsistency of the available data and the GMM. The EAS variability, on the other hand, is underestimated across the entire frequency range. The source of the variability inconsistency between data, cGM-GANO, and GMMs is currently under investigation by the authors. The median scaling of PSA is better captured by cGM-GANO, except at short distances, and the aleatory variability is more accurately recovered, especially for subduction events, due to the availability of the data.

Ongoing work by the authors focuses on addressing the limitations of the model training and on expanding the overall framework. Until recently, strong motions recorded by the National Research Institute for Earth Science and Disaster Resilience (NIED) have been the only available open-access datasets that could be used in the training of cGM-GANO. As publicly available, larger and higher quality recorded datasets from regions such as California are slowly emerging, we plan to use them to improve the distance scaling over short distances and the V_{S30} scaling at soft soils of cGM-GANO, as well as to resolve the bias observed in the low-frequency range of the empirical dataset presented here. Further validations with traditional GMMs will also be performed, including duration and intensity of ground shaking intensity measures. For conditions not well constrained in the empirical data, we plan to employ additional techniques, such as reinforcement learning with physics-based simulations.

Framework enhancements that we are investigating include the effects of ground motion filtering and instrument direction, as well as the partitioning of the stochastic component into within and between event variability. More specifically, in its current architecture, cGM-GANO does not recognize the effect of filtering in recorded ground motions, and thus, any differences in the filtering of ground motions are mapped onto the stochastic component of cGM-GANO output. Similarly, the horizontal components are currently being provided in the North and South direction, irrespective of the fault's azimuth, but, as past studies (e.g., Wald et al., 1994; Archuleta, 1984) have demonstrated, radiation patterns and thus the fault's azimuth can have a significant impact in the ground motion amplitude. Last, in this study's framework, each ground motion realization is produced for a random event, which may be adequate for characterizing the seismic risk of individual structures, but it is not suitable for evaluating the seismic risk at a system's level. These improvements are developed as part of several ongoing projects led by the authors, and the results thereof will be submitted for publication consideration in forthcoming manuscripts.

Data-driven models for ground motion synthesis may still be at the pilot stage, but with further refinements, they have the potential to bridge the gap between physics-based simulations and empirical models. Already in its current form, cGM-GANO can generate hundreds of site-specific realistic 3-component coherent ground motions with clear P-S arrivals and surface waves –currently only feasible with physics-based simulations– with the simplicity of empirical methods in less than a second. One should also consider this: data-driven generated ground motions can

span a broad range of frequencies (up to 30 Hz in this study), all of which are learned from real recorded ground motion data. By contrast, physics-based simulations may have broken the computational barrier of high-frequency deterministic wave propagation with EXASCALE computing, but they will continue to lag behind the empirical ground motion models in the high-frequency regime until the propagation medium reaches the input resolution necessary for these frequencies to realistically propagate through and be modified appropriately by the heterogeneous, nonlinear shallow crust.

Chapter 3

STOCHASTIC PROCESS LEARNING VIA OPERATOR FLOW MATCHING

This chapter presents Operator Flow Matching (OFM), a framework for learning stochastic-process priors in function space. OFM extends flow matching to stochastic processes by transporting a reference Gaussian process to a target data process using a neural-operator velocity field.

The goal of this chapter is to develop a generative framework that supports not only function sampling, but also density evaluation and Bayesian functional regression at arbitrary query points. After learning the prior, OFM enables posterior inference from partial and noisy observations through likelihood estimation and stochastic-gradient Langevin dynamics. The method is evaluated on Gaussian and non-Gaussian stochastic processes, as well as scientific datasets including Navier–Stokes examples and black-hole simulations.

The results show that OFM provides accurate functional regression and posterior sampling, outperforming Gaussian-process, deep-Gaussian-process, neural-process, and operator-flow baselines across a range of stochastic-process learning tasks.

The contents of this chapter are adapted from our publication (Shi et al., 2026b):

Shi, Yaozhong, Zachary Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2026b). “Stochastic process learning via operator flow matching”. In: *Advances in Neural Information Processing Systems* 38, pp. 38186–38226.

Additional details for this chapter are provided in Appendix B. The code associated with this work is available at https://github.com/yzshi5/SPL_OFM.

3.1 Introduction

Stochastic processes are foundational to many domains, from functional regression and physics-based data assimilation, to financial markets, geophysics, and black box optimization. Stochastic processes can serve as prior distributions over functions and can provide the density of any finite collection of points. Conventionally, these priors are designed by hand from predefined Gaussian processes (GP) and

their variants, and therefore assume that they adequately describe the phenomena of interest. However, many processes modeled in the natural world are not well described by GP, Fig 3.2. Such models limit the flexibility and generalizability of these stochastic processes in real-world applications, leaving behind significant challenges for more general stochastic process learning (SPL).

In SPL, the prior over the stochastic process is learned from data, i.e., a set of historical point evaluations in past experiments. Learning the prior over the process is crucial for universal functional regression (UFR), which is a recently proposed Bayesian scheme for functional regression and takes GP-regression as a special case when the prior is Gaussian (Shi et al., 2024a). UFR is important to scientific and engineering domains, including reanalysis, data completion-assimilation, uncertainty quantification, and black box optimization.

In this paper, we introduce a novel operator learning framework termed operator flow matching (OFM) for learning priors over stochastic processes through the joint distribution of any collection of points. To achieve this, we theoretically and empirically generalize marginal optimal transport flow matching (Tong et al., 2024) to infinite-dimensional function spaces where we map a GP into a prior over function spaces through a flow differential equation. We then derive SPL from the function space derivation and learn a prior over arbitrary sets of points. For SPL, we map any collection of pointwise evaluations of a GP to pointwise evaluations of target functions. This allows us to learn prior distributions over more general stochastic processes, hence enabling sampling values of any collection of points with their associated density and facilitating efficient UFR. We leverage this capability by extending neural operators Azizzadenesheli et al., 2024—designed initially to map functions between infinite-dimensional spaces—to maps between collections of points deploying their functional convergence properties. This serves as the essential architecture block in OFM.

After learning the prior and having access to the densities, OFM can be used for UFR, where given any collection of points of the underlying function, we estimate the posterior mean value of any new collection of points and efficiently sample from their posterior values using stochastic gradient Langevin dynamics (SGLD) (Welling et al., 2011), i.e., a Gaussian sampling on the input GP space (Fig 3.1). We show that OFM significantly outperforms state-of-the-art (SOTA) methods, including deep GPs, conditional models, and operator flows (OpFlow) (Kim et al., 2019; Gordon et al., 2019; Garnelo et al., 2018; Williams et al., 2006; Salimbeni et al., 2017;

Jankowiak et al., 2020; Shi et al., 2024a).

Generalizing GP-regression to regression over general stochastic process in a practical and implementable way demands a unified framework that spans many key fields. Because readers will have diverse backgrounds and expertise, we provide a set of potential questions and answers in Appendix B.1 to further clarify the development and highlight our contributions.

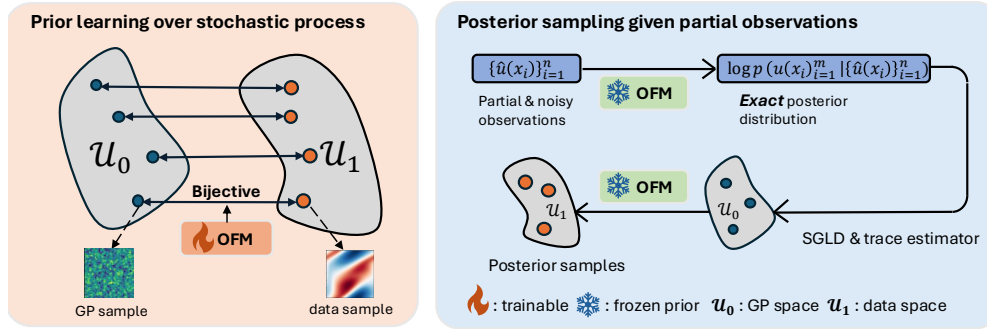


Figure 3.1: Two-phase strategy for prior learning and posterior sampling. In the prior-learning phase, OFM leverages the marginal optimal-transport path to learn a bijective mapping between a predefined GP and the unknown stochastic process that generates the training data. In the posterior-sampling phase, the learned prior is frozen; given noisy, partial observations, the exact posterior is obtained via Bayes’ theorem. SGLD, aided by the Hutchinson trace estimator, then enables efficient and robust sampling.

Lastly, our contributions are summarized as follows:

- **First** work to extend *flow matching / stochastic interpolants / rectified flow* (Lipman et al., 2022; Albergo et al., 2022; Liu et al., 2022) to stochastic processes via operator learning. The formulation enables likelihood estimation for function values at any collection of points for the target stochastic process. We also contribute to the development of marginal (dynamic) optimal transport in infinite-dimensional flow matching through optimal coupling and dynamic Kantorovich formulations.
- **First** integration of flow matching with functional regression yields a unified framework for prior and posterior sampling that is applicable to both generation and regression tasks.
- A practical generalization of GP regression that provides the **exact** prior and posterior density over an unknown stochastic process (whether GP or non-GP).

In contrast, previous methods such as deep GPs and conditional models work with approximate posteriors. Our method achieves **SOTA** performance in all challenging functional regression tasks.

- This work provides a unified perspective bridging several important fields, which opens new research directions for problems in science and engineering. Additionally, we present extensive ablation and scaling studies that demonstrate the effectiveness of each component in our framework.

3.2 Related Work

Neural operators. Neural operators constitute a paradigm in machine learning for learning maps between function spaces, a generalization of conventional neural networks that map between finite dimensional spaces (Li et al., 2020b; Kovachki et al., 2023). Among neural operator architectures, Fourier neural operators (FNO) (Li et al., 2020b) enable convolution in the spectral domain and are effective for operator learning (Yang et al., 2021; Pathak et al., 2022; Wen et al., 2022; Yang et al., 2023; Sun et al., 2023; Li et al., 2023). In this work, we use this as our neural operator architecture.

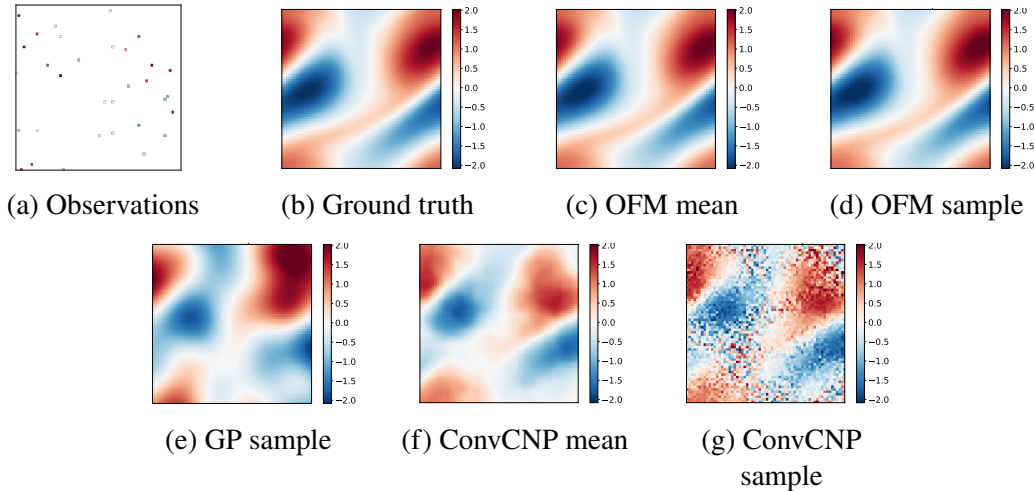


Figure 3.2: Operator Flow Matching (OFM) regression on Navier–Stokes functional data with resolution 64×64 . (a) 32 random observations (only 0.7%). (b) Ground truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.

Direct function samples. There is a body of work on generative models dedicated to learning distributions over functions, such that direct sampling on the function space

is possible. For example, generative adversarial neural operators (GANO) generalize generative adversarial nets on finite dimensional spaces to function spaces (Rahman et al., 2022; Shi et al., 2024b), yielding a neural operator generative model that maps GP to data functions (Azizzadenesheli et al., 2024). Other works in this area have followed the success of diffusion models (Song et al., 2020; Ho et al., 2020) in finite dimensional spaces, e.g., denoising diffusion operators generalize diffusion models to function spaces by using GP as a means to add noise and use neural operators to learn the score operator on function-valued data (Lim et al., 2025; Pidstrigach et al., 2024; Kerrigan et al., 2022). Moreover, the same principle has been deployed to generalize flow matching (Lipman et al., 2022) to functional spaces (Kerrigan et al., 2023), an approach closely related to our work. However, these works on learning generative models on function spaces do not support UFR the way GP-regression does because they (i) focus solely on generating function samples, (ii) do not clarify how to model a stochastic process on point value sequential generation, and (iii) do not provide point evaluation of probability density. In contrast, OFM enables exact density calculation and conditional posterior sampling.

Stochastic processes. Earlier works on SPL have focused on hand-tuned methods in the style of GP-regression. In these cases, an expert tunes the GP parameters given a set of experimental samples. More advanced methods rely on deep GPs, in which a network of GPs is stacked on top of each other. The parameters of deep GPs are commonly optimized by minimizing the variational free energy, which serves as a bound on the negative log marginal likelihood. (Damianou et al., 2013; Liu et al., 2020). Deep GPs have limitations in terms of learnability, expressivity, and computational complexity. Warped GPs (Kou et al., 2013) and transforming GP (Maroñas et al., 2021) methods use historical data to learn a pointwise transformation of GP values and achieve on par performance compared to deep GP type methods. The pointwise nature of such approaches limits their generality.

Another line of work proposes learning the conditional distribution of point values, inspired by variational inference and designed for sampling from function spaces; we refer to this line of models as conditional models¹ (Garnelo et al., 2018). This method trains a model to map any collection of points and their values to a vector, used as an input to a decoder that maps any collection of points to their values. The architectures used in these models (including the decoder) are not mathematically consistent as the number of points grows, limiting the approach to finite dimensions.

¹Neural process (NP) is a prominent example of conditional models

In particular, Convolutional Conditional Neural Processes (ConvCNPs) (Gordon et al., 2019) use convolutional architectures to capture local, translation-invariant structure, but they assume stationarity and struggle with long-range dependencies or highly non-uniform data. The diffusion based variants (Dutordoir et al., 2023) also use uncorrelated Gaussian noise, and the results do not exist in function spaces (Rahman et al., 2022; Lim et al., 2025). Furthermore, methods based on conditional models are unable to provide density estimation for collections of points, as needed for UFR and SPL in general.

Separately, OpFlow introduced invertible neural operators that are trained to map any collection of points sampled from a GP to a new collection of points in the data space (Shi et al., 2024a), using the maximum likelihood principle. This method is mathematically consistent as the resolution grows, captures the likelihood of any collection of points, and allows for UFR using SGLD. However, similar to normalizing flow (Papamakarios et al., 2021) methods in finite dimensional domains, the use of invertible deep learning models makes their training a challenge, particularly with regard to expressiveness, as also described in the original OpFlow work. Finally, we strongly encourage readers to consult Appendix B.1 and B.17 for an in-depth comparison of stochastic process learning and other generative frameworks.

3.3 Operator Flow Matching

Here, we introduce the problem setting and notations used for OFM in function space. We recommend that readers consult Appendix B.2–B.5 for a foundational overview of SPL, UFR and related background. Subsequently, we present the framework of OFM, which extends marginal optimal transport flow matching (Tong et al., 2024) to infinite-dimensional spaces. We further demonstrate the generalization of flow matching to stochastic processes as it is induced from OFM on function spaces. Finally, we illustrate how to evaluate exact and tractable likelihoods for any point evaluation of functions using OFM, making it applicable in the UFR setting.

For a real separable Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle, \|\cdot\|)$, equipped with the Borel σ -algebra of measurable sets denoted by $\mathcal{B}(\mathcal{H})$, we introduce two measures on $\mathcal{B}(\mathcal{H})$, ν_0 as the reference measure and ν_1 as the data measure. Consider a function h_0 sampled from ν_0 , such that $h_0 \sim \nu_0$. For a smooth time-varying infinite dimensional vector field $\mathcal{G}_t : \mathcal{H} \rightarrow \mathcal{H}$, we define an ordinary differential equation (ODE)

$$\frac{\partial \Phi_t(h_0)}{\partial t} = \mathcal{G}_t(\Phi_t(h_0)) \quad (3.1)$$

with initial condition $\Phi_0(h_0) = h_0$, where $h = h_t = \Phi_t(h_0)$ represents a function h_0 transported along a vector field from time 0 to time t . The diffeomorphism Φ_t induces a pushforward measure $\mu_t := [\Phi_t]_{\#}(\mu_0)$, with $\mu_0 = \nu_0$, and we refer to μ_t as the path of probability measure. The goal is to construct a path of probability measure such that at $t = 1$, $\mu_1 \approx \nu_1$. The dynamic relationship between the time varying measure μ_t and vector field $\mathcal{G}_t$ can be characterized by the continuity equation:

$$\frac{\partial \mu_t}{\partial t} = -\nabla \cdot (\mu_t \mathcal{G}_t) \quad (3.2)$$

In practice, we use Eq. 3.2 in its weak form (Ambrosio et al., 2005; Kerrigan et al., 2023) to check whether a given vector field $\mathcal{G}_t$ generates the target μ_t :

$$\int_0^1 \int_{\mathcal{H}} \frac{\partial \varphi(h, t)}{\partial t} + \langle \mathcal{G}_t(h), \nabla_h \varphi(h, t) \rangle d\mu_t(h) dt = 0 \quad (3.3)$$

Where $\varphi \in \text{Cyl}(\mathcal{H} \times [0, 1])$, and $\text{Cyl}(\mathcal{H} \times [0, 1])$ is the space of smooth cylindrical test functions. Suppose that the time-varying vector field $\mathcal{G}_t$ and the induced μ_t satisfying Eq. 3.3 are known. We parameterize $\mathcal{G}_t$ with a neural operator $\mathcal{G}_\theta : [0, 1] \times \mathcal{H} \rightarrow \mathcal{H}$ and regress $\mathcal{G}_\theta$ to target $\mathcal{G}_t$ through flow matching objective.

$$\mathcal{L}_{\text{FM}}^\dagger = \mathbb{E}_{t \sim \mathcal{U}[0, 1], h \sim \mu_t} \|\mathcal{G}_\theta(t, h) - \mathcal{G}_t(h)\|^2 \quad (3.4)$$

However, similar to its finite-dimensional counterpart, $\mathcal{G}_t$ is typically unknown. Moreover, there are infinitely many paths of probability measures that satisfy Eq. 3.3 and ensure $\mu_1 \approx \nu_1$. Therefore, it is necessary to specify a path of probability measures to effectively guide the learning of $\mathcal{G}_\theta$.

By constructing the appropriate Gaussian and conditional probability measures and leveraging optimal coupling together with the dynamic Kantorovich formulation (Chizat et al., 2018), we propose marginal (dynamic) optimal-transport flow matching in function space; detailed theory development and proofs are provided in Appendix B.6 and B.7. In the next subsection, we show how to generalize flow matching to stochastic processes, which is induced by the function space derivation.

Generalizing Flow Matching to Stochastic Processes

Stochastic processes are inherently infinite-dimensional and define distributions over any collection of points (Brémaud (2020), Chapter 5.1). We generalize the above marginal optimal transport flow matching on function spaces to stochastic processes by defining the transport map on any collection of points. We then show that, as

the collection of points covers the space in the limit, this generalization recovers infinite-dimensional flow matching implemented with neural operators.

For any n and points $\{x_1, x_2, \dots, x_n\}$, consider an ODE system in which a vector of random variables $u_0 \in \mathbb{R}^n$ is gradually transformed into $u_1 \in \mathbb{R}^n$, for which, the i th entry is equal to $u(x_i)$, via a smooth, time-varying vector field, denoted by $\mathcal{G}_t$ with abuse of notation.

$$u_t := \Phi_t(u_0) = u_0 + \int_0^t \mathcal{G}_s(u_s) ds \quad (3.5)$$

Here, the neural operator is applied to a collection of point evaluations. Given the set of points and the density of $p_0 := p_0(\{u_0(x_1), u_0(x_2), \dots, u_0(x_n)\}) = \mathcal{N}(\mathbf{0}, K(\{x_1, x_2, \dots, x_n\}))$, where K is a $n \times n$ covariance matrix with entries described by kernel function $k(x_i, x_j)$ and $u_0 \sim p_0$, the time-varying density p_t induced by the diffeomorphism Φ_t or $\mathcal{G}_t$ can be computed, extending the transport equation Eq. 3.2 to collections of points,

$$\frac{\partial p_t(u_t)}{\partial t} = -(\nabla \cdot (\mathcal{G}_t p_t))(u_t) \quad (3.6)$$

Eq. 3.6 shows that constructing p_t is equivalent to constructing $\mathcal{G}_t$ for finite entries for which the analysis carries to finite collections of random variables. In the following, we refer to p_t as the marginal probability path induced by $\mathcal{G}_t$ for the given collection of points. From Eq. 3.6, the log density can be computed through integration,

$$\log p_t(u_t) = \log p_0(u_0) - \int_0^t (\nabla \cdot \mathcal{G}_s)(u_s) ds \quad (3.7)$$

In this formulation, we are seeking a specific vector field that transports density q_0 to target density q_1 for any n and any collection of points $\{x_1, x_2, \dots, x_n\}$ with boundary conditions $p_0 = q_0, p_1 = q_1$. Extending optimal transport flow matching to stochastic processes, we parameterize the vector field $\mathcal{G}_t$ with a neural operator $\mathcal{G}_\theta$, which is optimized through the flow matching objective for SPL,

$$\mathcal{L}_{\text{FM}} := \sup_n \sup_{\{x_1, \dots, x_n\}} \mathbb{E}_{t \sim \mathcal{U}(0,1), u_t \sim p_t} \|\mathcal{G}_\theta(t, u_t) - \mathcal{G}_t(u_t)\|^2 \quad (3.8)$$

Note that p_t and u_t depend on the point collocations. In the above equation, the suprema are intractable and we replace them with an expectation as a soft approximation (see Appendix B.1. Q5 for a detailed discussion of the approximation). Moreover, the true $\mathcal{G}_t$ is usually unknown and to address it, we derive a probability path conditioned on latent variable z of the same alphabet size as the collection.

Consequently, the marginal probability path $p_t(u_t)$ is a mixture of conditional probability paths $p_t(u_t|z)$,

$$p_t(u_t) = \int p_t(u_t|z)q(z)dz \quad (3.9)$$

$$\mathcal{G}_t(u_t) = \mathbb{E}_{q(z)}\left[\frac{\mathcal{G}_t(u_t|z)p_t(u_t|z)}{p_t(u_t)}\right]. \quad (3.10)$$

Given Eq. 3.10, the conditional flow matching (CFM) objective is defined as,

$$\mathcal{L}_{\text{CFM}} := \mathbb{E}_n \mathbb{E}_{x_1, \dots, x_n} \mathbb{E}_{t, q(z), p_t(u_t|z)} \|\mathcal{G}_\theta(t, u_t) - \mathcal{G}_t(u_t|z)\|^2. \quad (3.11)$$

Eqs. 3.11 and 3.10 have an identical gradient for θ , i.e. $\nabla_\theta \mathcal{L}_{\text{FM}}(\theta) = \nabla_\theta \mathcal{L}_{\text{CFM}}(\theta)$. Inspired by the finite dimensional developments (Tong et al., 2024), the variable z is chosen as a couple (u_0, u_1) from the coupling $\pi(u_0, u_1) = q(z)$, which is achieved by minimizing the dynamic 2-Wasserstein distance,

$$W_{\text{dyn}}(q_0, q_1)_2^2 = \inf_{p_t, \mathcal{G}_t} \int_0^1 \int_{\mathbb{R}^n} p_t(u_t) \|\mathcal{G}_t(u_t)\|^2 du_t dt \quad (3.12)$$

Under mild conditions on $\mathbb{R}^n$, this is equivalent to the static 2-Wasserstein distance,

$$W_{\text{sta}}(q_0, q_1)_2^2 = \inf_{\pi \in \Pi} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|u_1 - u_0\|^2 d\pi(u_0, u_1). \quad (3.13)$$

Considering the class of Gaussian conditional probability paths

$$p_t(u_t|z) = \mathcal{N}(u_t|m_t(z), \sigma_t(z)^2 K(\{x_1, x_2, \dots, x_n\}))$$

with conditional flow $\Phi_t(u_0|z) = \sigma_t u_0 + m_t$. Specially, we choose $m_t = tu_1 + (1-t)u_0$ and $\sigma_t = \sigma$, where $\sigma > 0$ is a small constant. Then we have the following closed-form expression for the corresponding vector field (full derivation provided in Appendix B.8)

$$\mathcal{G}_t(u_t|z) = u_1 - u_0 \quad (3.14)$$

With the aforementioned developments, for any collection of points, we transport a Gaussian distribution to a target distribution. The Gaussian distribution is drawn from a GP, with its covariance matrix $K(x_1, \dots, x_n)$ determined by the kernel function $k(x_i, x_j)$ of the GP. According to Kolmogorov extension theorem (KET) (Kolmogorov, 2018), there exists a valid stochastic process whose finite-dimensional marginal is the pushforward distribution under $\mathcal{G}_\theta$. This demonstrates that the generalization of flow matching to infinite-dimensional spaces with neural operators naturally induces the generalization of flow matching to stochastic processes. In the

scenario where the limit of points covers the space, these two become equivalent. For a detailed explanation and proof, please refer to Appendix B.3 and B.4.

Our framework extends seamlessly to alternative probability paths, such as those in *stochastic interpolants and rectified flow* (Albergo et al., 2022; Liu et al., 2022), because the generalization to stochastic processes is decoupled from any specific path. However, we focus on the OT path in this work for two reasons: (1) ablation and scaling studies show that it accelerates inference and enables more accurate prior learning compared to independent coupling (Table B.6, and Appendix B.16); (2) theoretically, the OT path (straight line) simplifies the Jacobian evaluation required for likelihood estimation (discussed in the next subsection), yielding greater stability and speed than arbitrary paths.

Likelihood Estimation and Bayesian Universal Functional Regression

We parameterize $\mathcal{G}_\theta$ with FNO (Li et al., 2020b) to ensure our model is resolution agnostic, and assume $\mathcal{G}_\theta$ learns a map from ν_0 to ν_1 , which serves as the prior. In practice, we deal with discretized evaluations of functions that may have different sampling rate and resolution. For instance, consider a function u sampled from ν_1 , observed on a collection of points $u_1 := \{u(x_1), u(x_2), \dots, u(x_m)\}$; thus we have a density function $\mathbb{P}(u_1)$ defined on collection of points $\{x_1, x_2, \dots, x_m\}$, where $\mathbb{P}(u_1)$ is derived from measure ν_1 . This is similar to how a multivariate Gaussian distribution can be derived from a Gaussian measure characterized by a Gaussian process. Therefore, we can rewrite Eq. 3.7 as:

$$\log \mathbb{P}(u_1) = \log \mathbb{P}(u_0) - \int_0^1 (\nabla \cdot \mathcal{G}_\theta)(u_t) dt \quad (3.15)$$

where u_0 is drawn from the reference Gaussian measure ν_0 , which is also defined on the collection of points $\{x_1, x_2, \dots, x_m\}$. Thus, $\mathbb{P}(u_0)$ is a multivariate Gaussian with a tractable density function. Furthermore, with the probability density function $\mathbb{P}(u_1)$, we can evaluate the precise likelihood of any u_1 from $\mathbb{P}(u_1)$ via Eq. 3.15. However, following a similar argument to Grathwohl et al. (2018), the computation of $\nabla \cdot \mathcal{G}_\theta(u)$ incurs a cost of $\mathcal{O}(m^2)$ where m is the cardinality of $\{x_1, x_2, \dots, x_m\}$. This quadratic time complexity renders the likelihood calculation prohibitively expensive. To address this issue, we adopt the strategy proposed in Grathwohl et al. (2018), using the unbiased Skilling-Hutchinson trace estimator (Hutchinson, 1989; Skilling, 1989) to approximate the divergence term. This technique reduces the computation cost to $\mathcal{O}(m)$, which is the same as the cost of inference, thereby streamlining the

evaluation process. The estimator is implemented as follows:

$$\nabla \cdot \mathcal{G}_\theta(u_t) = \mathbb{E}_{p(\varepsilon)}[\varepsilon^T \frac{\partial \mathcal{G}_\theta(u, t)}{\partial u} \varepsilon] \quad (3.16)$$

In the unbiased trace estimator, the random variable ε is characterized by $\mathbb{E}(\varepsilon) = 0$ and $\text{Cov}(\varepsilon) = I$. This estimator benefits from the optimal transport nature of the map which gives rise to a direct line. The gradient computation in Eq. 3.16 can be efficiently handled with reverse-mode automatic differentiation, allowing for precise estimation with arbitrary error by averaging over a sufficient number of runs, which can benefit from parallel computing of GPUs.

With the efficient tool established for estimating the likelihood of any discretized function samples, we now turn our attention to UFR, i.e., Bayesian functional regression with a learned prior. Consider a collection of pointwise observations of the underlying unknown function drawn from ν_1 , corrupted with Gaussian noise, denoted as $\{\widehat{u}(x_1), \widehat{u}(x_2), \dots, \widehat{u}(x_n)\}$ or $\{\widehat{u}(x_i)\}_{i=1}^n$. We specifically focus on Gaussian white noise characterized by $\epsilon \sim \mathcal{N}(0, \sigma^2)$, such that $\widehat{u}(x_i) = u(x_i) + \epsilon_i$ for $i \in \{1, \dots, n\}$ (depending on the nature of the problem, the noise may also be considered as a correlated GP noise). In UFR setting, we are interested in the posterior distribution over new $m \geq n$ points that include the n observation points.

Proposition 3.3.1. *Given noisy observations $\{\widehat{u}(x_i)\}_{i=1}^n$, the posterior distribution is*

$$\log \mathbb{P} \left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n \right) = -\frac{\sum_{i=1}^n \|\widehat{u}(x_i) - u(x_i)\|^2}{2\sigma^2} + \log \mathbb{P}(\{u(x_i)\}_{i=1}^m) + C \quad (3.17)$$

Where the constant $C = -\frac{n}{2} \log(2\pi\sigma^2) - \log \mathbb{P}(\{\widehat{u}(x_i)\}_{i=1}^n)$.

Proof. This is derived from Bayes' theorem, along with the translation invariance property of the Gaussian distribution, see the full proof in Appendix B.9 $\square$

Given this form of the posterior distribution, we adopt SGLD (Welling et al., 2011) to efficiently sample from it, and then derive statistical features of interest, e.g. mean, maximum a posteriori, and posterior uncertainty, i.e., variance, from the posterior samples. More specifically, we follow the posterior sampling strategy developed by Shi et al. (2024a), which, given an invertible framework, suggests SGLD sampling within the input GP space where the Gaussian measure ν_0 is defined and Langevin dynamics is native to, and then mapping to the data function space

(where data measure ν_1 is defined). We also present a scaling study in Appendix B.16 analyzing the variance introduced by the Hutchinson trace estimator, demonstrating its robustness and effectiveness when paired with SGLD sampling. Last, Eq 3.17 offers a unified view of prior and posterior sampling with flow matching models by showing that when no observations are present, the posterior collapses to the prior (up to a constant), making the posterior sampling process identical to that of the prior.

Finally, we provide a plain-language summary of the framework to help the reader better understand the paper (with a detailed discussion available in Appendix B.17),

- OFM is an *expressive, flow-based generative model that learns a prior over functions (stochastic process)*: a neural operator parameterizes a continuous probability flow that transports samples from a simple reference Gaussian process to data-like functions, yielding an explicit prior with a tractable density.
- It excels at functional regression by treating functions as *first-class objects* rather than mere pointwise values (unlike NPs), yielding predictions that are consistent across resolutions and arbitrary query sets.
- The learned flow is invertible, enabling change-of-variables for likelihoods and principled Bayesian regression, yielding calibrated uncertainty from few observations.

3.4 Experiments

In this section, we demonstrate the superior regression performance compared to several baselines across a variety of function datasets, including both Gaussian and highly non-Gaussian Processes. As baselines, we employ standard GP Regression (Williams et al., 2006), Deep GPs (Salimbeni et al., 2017; Jankowiak et al., 2020), Conditional models (Garnelo et al., 2018; Kim et al., 2019; Gordon et al., 2019), and OpFlow (Shi et al., 2024a).

For our function datasets, we analyze: (1) Gaussian and non-Gaussian with known posterior, including 1D GPs, 2D GPs, and 1D Truncated GPs, (TGP). (2) Highly non-GPs, datasets with unknown posterior, such as those derived from Navier-Stokes equations (Li et al., 2020b), black hole dataset from expensive Monte Carlo simulation, and 2D Signed Distance Functions extracted from MNIST digits (MNIST-SDF) (Sitzmann et al., 2020). During regression, we assume that the prior $\mathcal{G}_\theta$ is always

successfully trained and remains frozen. Details about the learning process for priors and experimental setup for regression are provided in the Appendix B.13, B.15.

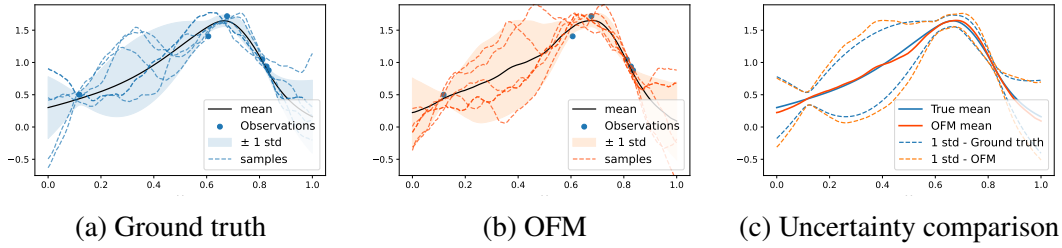


Figure 3.3: OFM regression on GP data. (a) Ground-truth GP regression with observed data and posterior samples. (b) OFM regression with observed data and posterior samples. (c) Standard-deviation comparison between the true GP posterior and OFM predictions.

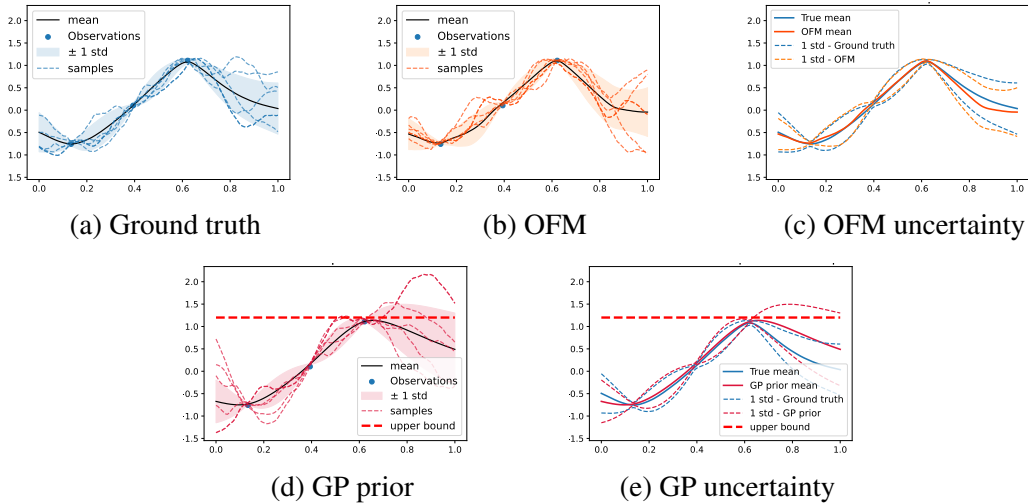


Figure 3.4: OFM regression on transformed Gaussian process data.

1D GP data. This experiment replicates the results of classical GP regression, wherein the posterior distributions are precisely known in a closed form. The process involves generating a single new realization from the data measure ν_1 . We then select observations at $n = 6$ randomly chosen positions, incorporating a predefined noise level. The posterior is inferred across $m = 128$ positions, which includes estimating noise-free values at the observation points. We evaluate our results with two commonly used quantities in the GP literature (1) Standardized Mean Squared Error (SMSE) that normalizes the mean squared error by the variance of the ground truth; and (2) Mean Standardized Log Loss (MSLL), originally introduced

by Williams et al. (2006), defined as:

$$-\log p(\{u(x_i)\}_{i=1}^m | \{\hat{u}(x_i)\}_{i=1}^n) = \frac{1}{2} \log(2\pi\sigma^2) + \frac{(\{u(x_i)\}_{i=1}^m - \{\bar{u}(x_i)\}_{i=1}^m)^2}{2\sigma^2} \quad (3.18)$$

where $\{\hat{u}(x_i)\}_{i=1}^n$ represents observations, $\{x_i\}_{i=1}^m$, $\{u(x_i)\}_{i=1}^m$, indicate the new positions queried, and the test data (true posterior samples). Meanwhile, $\{\bar{u}(x_i)\}_{i=1}^m$ and σ^2 are predicted mean and variances from the model. We average out SMSE and MSLL over a test dataset containing 1000 true GP posterior samples for all models. The performance of each model is detailed in Table 3.1. From Fig. 3.3, the regression with OFM matches the analytical solution well and provides realistic posterior samples. We further include GP regression tasks using more complex kernels (Gibbs and Rational Quadratic kernel), as shown in Appendix B.14, OFM consistently outperforms all comparative methods across all metrics.

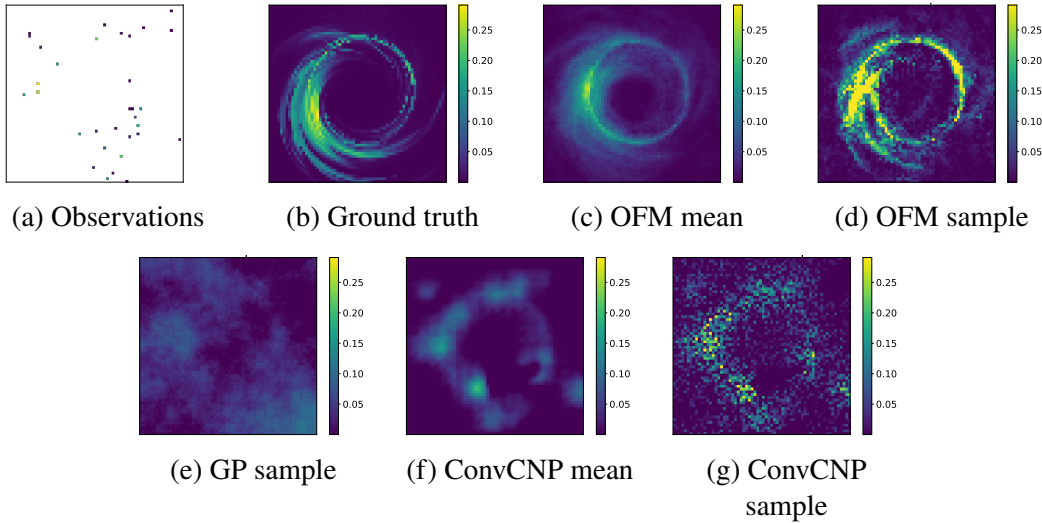


Figure 3.5: OFM regression on black hole data with resolution 64×64 . (a) 32 random observations. (b) Ground-truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.

Truncated GP data. In this experiment, we analyze the regression performance of OFM for tractable non-GP. Specifically, we work on truncated GP (Swiler et al., 2020; Shi et al., 2024a), which constrains the function amplitude within a specified range. This is achieved by applying a sampling-rejection strategy on samples from the GP prior. We set the bounds of our TGP to $[-1.2, 1.2]$ and perform regression using observations only at three points, while estimating the posterior across $m = 128$ points. Subsequently, we sample 1000 true TGP posteriors from the GP prior to

Table 3.1: Comparison of OFM with baseline models: GP regression; OpFlow (Shi et al., 2024a); Conditional models: NP (Garnelo et al., 2018); Attentive NP (Kim et al., 2019, ANP); Convolutional Conditional NP (Gordon et al., 2019, ConvCNP); Deep variational GP (Salimbeni et al., 2017, DGP); Deep Sigma Point Process (Jankowiak et al., 2020, DSSP); Metrics SMSE and MSLL used for 1D and 2D GP example. Mean squared error for the predicted mean (μ) and standard deviation (σ) are used for TGP example. Performance of GP regression for 1D and 2D GP are removed (marked with '-'), which are taken as the ground truth. Best performance in bold.

Dataset →	1D GP		2D GP		1D TGP	
Algorithm ↓ Metric →	SMSE	MSLL	SMSE	MSLL	μ	σ
GP prior	-	-	-	-	$6.4 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$
NP	$6.1 \cdot 10^{-1}$	$4.5 \cdot 10^0$	$1.7 \cdot 10^{-1}$	$2.1 \cdot 10^0$	$1.0 \cdot 10^{-1}$	$1.9 \cdot 10^{-2}$
ANP	$5.1 \cdot 10^{-1}$	$9.8 \cdot 10^{-1}$	$1.6 \cdot 10^{-1}$	$1.1 \cdot 10^0$	$1.4 \cdot 10^{-1}$	$1.7 \cdot 10^{-2}$
ConvCNP	$5.6 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$4.5 \cdot 10^{-1}$	$1.6 \cdot 10^{-2}$	$2.1 \cdot 10^{-3}$
DGP	$4.1 \cdot 10^{-1}$	$6.8 \cdot 10^{-2}$	$1.8 \cdot 10^0$	$4.2 \cdot 10^0$	$4.9 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$
DSPP	$4.7 \cdot 10^{-1}$	$6.5 \cdot 10^0$	$1.9 \cdot 10^{-1}$	$6.6 \cdot 10^0$	$1.1 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$
OpFlow	$5.0 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$	$3.9 \cdot 10^{-3}$
OFM(Ours)	$4.1 \cdot 10^{-1}$	$5.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$1.6 \cdot 10^{-1}$	$5.2 \cdot 10^{-3}$	$9.5 \cdot 10^{-4}$

calculate the mean and standard deviation. Traditional metrics like MSLL and SMSE, which assume a Gaussian posterior, are not suitable for TGP. We evaluate the performance using the mean squared error for both the predicted mean and standard deviation. The results are reported in Table. 3.1, and illustrated in Fig. 3.4. OFM accurately learns the specified bounds and provides accurate estimations of mean and standard deviation, along with realistic posterior samples. In contrast, directly applying GP regression exceeds the bounds and yields unrealistic posterior samples.

2D GP data. Similar to the 1D GP example, we extend our regression analysis to 2D GP data. As shown in Fig. B.1 and detailed in Table 3.1, OFM provide accurate posterior estimation. The relative error shown in Fig. B.1 is the absolute error normalized by the maximum absolute value of the mean prediction derived from the ground truth GP regression.

Navier-Stokes, Black hole and MNIST-SDF datasets. We collected a 2D Navier-Stokes dataset and applied OFM for the regression. Unlike the GP experiments, where MSLL and SMSE score serve as standard benchmarks, evaluating the performance of models on general non-GPs presents a significant challenge due to the difficulty of determining the true posterior and lack of benchmarks. Moreover, the evidence term in the posterior (Eq. B.23) is intractable, so the likelihood cannot serve as a

meaningful evaluation metric in our setting. A detailed discussion is provided in Appendix B.1.

We present the predicted mean and a posterior sample in Fig 3.2 for visual comparison with the ground truth. The predicted mean and the posterior sample are closely aligned with the ground truth. In contrast, traditional GP regression and NP models failed to accurately capture the dynamics of the Navier-Stokes data. In Fig. 3.5, we conduct a similar analysis using a simulated black hole dataset. Here, OFM provides a much more realistic mean and posterior sample that capture the density and swirling patterns of the black hole. Once again, GP regression and NP fail to capture these key statistics. We observe similar outcomes when applying OFM to the MNIST-SDF example (Fig B.3), where OFM correctly recognizes the number "7" while GP regression does not.

3.5 Conclusion

In this paper, we proposed Operator Flow Matching (OFM) for stochastic process learning, which generalizes flow matching models to infinite-dimensional space and stochastic process with optimal transport path. OFM efficiently computes the probability density for any finite collection of points and supports mathematically tractable functional regression. We extensively tested OFM across a diverse range of datasets, including those with closed-form GP and non-GP data, as well as highly non-GP such as Navier-Stokes and black hole data. In comparative evaluations, OFM consistently outperformed all baseline models, establishing new standards in stochastic process learning and regression.

Despite SOTA accuracy, our method is presently limited to low-dimensional domains and demands larger datasets and more compute than GP-based baselines. see Appendix B.1, B.15 and B.17 for details.

Chapter 4

MESH-INFORMED NEURAL OPERATOR: A TRANSFORMER GENERATIVE APPROACH

This chapter presents the Mesh-Informed Neural Operator (MINO), a domain- and discretization-agnostic backbone for functional generative modeling. The goal is to extend generative models in function spaces beyond regular grids and rectangular domains, which are common limitations of Fourier-neural-operator-based implementations.

MINO combines graph neural operators with cross-attention mechanisms. The graph neural operator acts as a geometry-aware tokenizer for functions sampled on arbitrary meshes, while the cross-attention encoder and decoder map between irregular input locations and latent representations. This design enables generative modeling on both regular grids and irregular domains, while retaining the function-space perspective of neural operators.

The chapter evaluates MINO on a suite of regular- and irregular-grid benchmarks, including Navier–Stokes, shallow-water, Darcy-flow, cylinder-flow, Mesh-GP, and global-climate datasets. The results show that MINO achieves strong generative performance and supports zero-shot generation at different spatial resolutions. The chapter also introduces Sliced Wasserstein Distance and Maximum Mean Discrepancy as general-purpose metrics for evaluating functional generative models.

The contents of this chapter are adapted from our publication (Shi et al., 2025):

Shi, Yaozhong, Zachary E. Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2025). “Mesh-Informed Neural Operator : A Transformer Generative Approach”. In: *Transactions on Machine Learning Research*. issn: 2835-8856.

Additional details for this chapter are provided in Appendix C. The code associated with this chapter is available at <https://github.com/yzshi5/MINO>.

4.1 Introduction

Generative models are powerful tools for fields dealing with complex data distributions, with recent advances in diffusion and flow matching models demonstrating impressive capabilities for synthesizing high-fidelity images (Song et al., 2020; Ho

et al., 2020; Lipman et al., 2022), audio (Liu et al., 2023; Huang et al., 2023), and video (Jin et al., 2025). These models excel at learning highly complicated probability distributions in finite-dimensional spaces. However, numerous fields in science and engineering—such as seismology, biomechanics, astrophysics and atmospheric sciences—primarily deal with data that inherently reside in infinite-dimensional function spaces. Moreover, in many cases, the data collected in these fields are functions sampled on heterogeneous networks of sensors, or even on manifolds (e.g., global seismic networks or weather stations on Earth’s surface). Functional generative models are especially important for these fields because they routinely deal with several broad (sometimes overlapping) challenges: (i) uncertainty quantification in physical units, (ii) the inherent non-uniqueness present in, e.g. solutions to inverse problems, and (iii) the need to model stochastic or latent fields that are effectively unobservable, which are better handled probabilistically. Together, these factors necessitate a paradigm shift towards generative models that operate directly in function spaces.

Recent studies have generalized various generative paradigms to function spaces (Rahman et al., 2022; Shi et al., 2024a; Shi et al., 2024b; Kerrigan et al., 2022; Kerrigan et al., 2023; Lim et al., 2025; Seidman et al., 2023) by leveraging neural operators (Azzadenesheli et al., 2024; Kovachki et al., 2023). Despite these advancements, two critical bottlenecks hinder the broader adoption and rigorous evaluation of functional generative models. First, current implementations are predominantly based on FNO (Li et al., 2020b) as their backbone. However, FNO implementations are restricted to regular grids on rectangular domains, thereby preventing the realization of many key theoretical benefits. Second, prior studies rely on dataset-specific metrics, making it difficult to compare functional generative models across datasets and to draw robust conclusions about generative quality.

This study addresses the aforementioned limitations of functional generative models. Specifically, our contributions are summarized as follows:

- We introduce MINO, a domain-agnostic functional generative backbone. Our comprehensive experiments show that MINO achieves state-of-the-art (SOTA) performance on a diverse suite of benchmarks with regular and irregular grids.
- We demonstrate through analysis and experiments that Sliced Wasserstein Distance (SWD) and Maximum Mean Discrepancy (MMD) are efficient, ro-

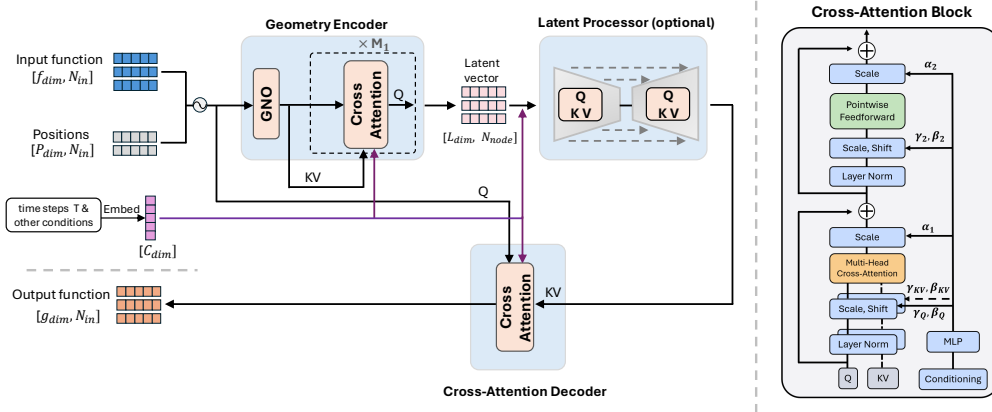


Figure 4.1: Overview of the MINO architecture. The geometry encoder uses a GNO as a domain-agnostic tokenizer, followed by several cross-attention blocks and an optional latent processor. The decoder then employs a distinct cross-attention mechanism to map the latent representation back to the target locations.

bust, dataset-independent metrics for evaluating the performance of functional generative models on regular and irregular grids.

- We propose a novel framework that integrates operator learning with modern deep learning practice. By using a graph neural operator as a domain-agnostic tokenizer and embedding tailored cross-attention modules in both the encoder and decoder, our design avoids information bottlenecks present in prior encoder-processor-decoder neural operators, allowing any powerful, finite-dimensional network (e.g., U-Net or Transformer) to serve as the latent processor.

4.2 Preliminaries

Neural operators. Neural operators learn mappings between function spaces, extending deep learning beyond the fixed-dimensional vectors handled by classical networks. This function space perspective is especially important for scientific computing that is governed by partial differential equations (PDEs) (Kovachki et al., 2023; Li et al., 2020b). A core theoretical property of neural operators is *discretization convergence (agnosticism)*: as the mesh of input function is refined, the prediction approaches the unique continuous solution (Kovachki et al., 2023). The Fourier Neural Operator (FNO) (Li et al., 2020b) achieves this property via spectral-domain convolutions, achieving quasi-linear complexity on regular grids. Its global mixing excels at capturing long-range dependencies but may miss fine-scale detail (required

for generative tasks) unless many Fourier modes are retained (Liu-Schiaffini et al., 2024).

For handling functions on irregular grids, Graph Neural Operator (GNO) (Gilmer et al., 2017; Li et al., 2020c) remains a powerful candidate. GNO shares the message-passing functionality of Graph Neural Network (GNN), but differs in that the search radius for neighbors is defined in physical coordinates and made consistent across resolutions to guarantee discretization convergence. This results in GNO having a local receptive field, however it is computationally inefficient for certain tasks (Li et al., 2023; Liu-Schiaffini et al., 2024) and often struggles to learn the high-frequency information required for many generative models. This difficulty arises from the standard message-passing mechanisms, which tend to act as low-pass filters (Li et al., 2018; Di Giovanni et al., 2022). More recent work has focused on improved scalability and performance aspects of Neural Operators on irregular grids and non-rectangular physical domains, e.g., Geometry-Informed Neural Operator (GINO), Universal Physical Transformer (UPT), Latent Neural Operator (LNO), and Transolver (Li et al., 2023; Alkin et al., 2024; Wang et al., 2024b; Wu et al., 2024). However, it remains unclear whether these architectures will also be effective for functional generative modeling on irregular grids (see Appendix C.3).

Functional generative models. Classical generative models learn a mapping from a simple base distribution (e.g., multi-variate Gaussian) to a target finite dimensional data distribution. Alternatively, the functional perspective frames this problem in infinite-dimensional spaces. A sample function $f_1 : D \rightarrow \mathbb{R}^{f_{dim}}$, with D denoting the domain of the function, is treated as a sample from a target probability measure μ_1 over a function space. The discrete samples are formed by evaluating f_1 on a set of grid points of locations. Functional generative models often aim to learn a mapping between function spaces by training a neural operator to transform a function f_0 sampled from a simple base measure μ_0 (typically a Gaussian random field measure $\mathcal{N}(0, C)$) into a function f_1 that follows the target measure μ_1 . The distribution of the discrete data is thus a finite-dimensional marginal of the true data measure μ_1 .

Adopting this perspective, functional generative models offer several key benefits not generally available in their traditional finite-dimensional counterparts:

- *Flexible domain geometries:* The function domain D is not required to be rectangular; it could be an irregular domain or even a manifold. Such flexibil-

ity is ideal for many real-world scenarios in science and engineering, such as modeling rainfall over a city with an irregular boundary, or modeling weather patterns on the Earth’s surface.

- *Discretization agnosticism (mesh invariance)*: By learning a probability measure over functions, functional generative models can be trained on samples with varying discretizations and can subsequently generate new function samples at any desired discretization in a zero-shot manner.
- *Inference-time control and guidance in function space*: Functional diffusion/flow models enable the incorporation of inference-time scaling rules and precise guidance signals directly within function spaces. For instance, this facilitates the development of probabilistic PDE solvers that can rigorously enforce hard boundary conditions or other external controls during generation (Cheng et al., 2025; Yao et al., 2026).
- *Universal functional regression via stochastic process learning*: By leveraging invertible trajectories derived from neural operators, it is possible to perform functional regression with learned (non-Gaussian) stochastic process priors (Shi et al., 2024a; Shi et al., 2026b). These learned distributions are capable of providing exact prior and posterior density estimation for a general stochastic process.

Flow matching in Hilbert space. Recent work has established a rigorous mathematical framework for extending flow matching to Hilbert spaces (Kerrigan et al., 2023), Functional Flow Matching (FFM), where a velocity field is learned to transport a base Gaussian measure to a target measure. This was further built upon by Operator Flow Matching (OFM) (Shi et al., 2026b), which extended the paradigm to stochastic processes and incorporated a dynamic optimal-transport path.

The core idea of FFM/OFM is to learn a continuous path of functions, f_t for $t \in [0, 1]$, that transforms samples f_0 from a base Gaussian measure μ_0 into samples f_1 from a target data measure μ_1 . This transformation is governed by an Ordinary Differential Equation (ODE) whose velocity field, v_θ , is parameterized by a neural operator with weights θ :

$$\frac{df_t}{dt} = v_\theta(f_t, t) \quad (4.1)$$

For training, OFM draws pairs of functions (f_0, f_1) from a mini-batch optimal coupling $\pi(\mu_0, \mu_1)$, which is achieved by minimizing the 2-Wasserstein distance

between μ_0 and μ_1 . The base measure, μ_0 , is typically a Gaussian measure $\mathcal{N}(0, C)$, where C is a trace-class covariance operator. The framework then defines the path f_t as a linear interpolation between f_0 and f_1 , i.e., $f_t = (1 - t)f_0 + tf_1$. f_t induces a time-dependent probability measure μ_t , such that at $t = 0, \mu_t = \mu_0$ and $t = 1, \mu_t = \mu_1$. This specific choice yields a simple ground truth expression for the velocity, $v_t = f_1 - f_0$. The training objective is then to minimize the Mean Squared Error (MSE) between the parameterized velocity field $v_\theta(f_t, t)$ and this target v_t . To generate a new sample, one draws $f_0 \sim \mu_0$ and solves the learned ODE numerically. While functional diffusion models offer a related framework, they typically rely on Stochastic Differential Equations (SDEs), which introduce an additional perturbation term with longer generation time and inferior performance compared to ODE-based flow matching (Lim et al., 2025; Kerrigan et al., 2022). A fundamental property of both paradigms is *domain alignment*: for a given trajectory, all evolving functions f_t must be defined on the same fixed spatial domain D . ($\mu_t \in \mathcal{P}(L^2(D; \mathbb{R}^{f_{dim}}))$). Crucially, the theory allows this domain D to be geometrically complex; it is not restricted to a rectangular shape and can be a domain with an irregular boundary or even a manifold. However, this theoretical flexibility has been underutilized in practice, as existing implementations predominantly rely on FNO, which restricts them to domains discretized by regular grids.

Grid-Based Architectures. The modern U-Net architecture used in diffusion models (Rombach et al., 2022) has become a de-facto standard for generative tasks on grid-like data. These advanced U-Nets are distinct from the variant used in some earlier neural operators (Rahman et al., 2023), the U-Net can effectively capture multi-scale information through their hierarchical structure, skip connections, and integration of attention and residual blocks. Nevertheless, they remain constrained to regular grids due to their reliance on standard convolutional neural network kernels. Another prominent line of work involves Transformer-based architectures. Standard Vision Transformers (ViTs) and Diffusion Transformers (DiTs) (Dosovitskiy et al., 2020; Peebles et al., 2022), first "patchify" the input into non-overlapping blocks and then apply full self-attention, incurring $O(N_{\text{patch}}^2)$ time complexity. For images, $N_{\text{patch}} \ll H \times W$ (Height and Width), yet the patchify operation creates discontinuities at block borders that can lead to checkerboard artifacts (Fang et al., 2023). Moreover, both ViT and DiT presuppose a fixed rectangular grid so that patchification is well defined.

Table 4.1: Comparison of MINO with other architectures. "local $\rightarrow$ global" receptive field gathers global context through stacked local operations. "local + global" receptive field combines local and global information directly. A detailed discussion is provided in Appendix C.3

Architecture	Efficient	Receptive field	Irregular grid	Discretization convergent	Frequency learnt
GNN	✗	local	✓	✗	low
GNO	✗	local	✓	✓	low
FNO	✓	global	✗	✓	low + high
U-Net	✓	local $\rightarrow$ global	✗	✗	low + high
ViT/DiT	✓	global	✗	✗	low + high
GINO	✓	local $\rightarrow$ global	✓	✓	low
UPT	✓	local $\rightarrow$ global	✓	✓	low
MINO [Ours]	✓	local + global	✓	✓	low + high

4.3 Methods

Our proposed architecture, MINO, provides an effective solution for functional generative models on non-rectangular domains with irregular grids. It integrates powerful deep learning backbones with neural operators in a manner that is tailored for improving functional generative performance. Specifically, this is achieved by combining GNO with cross-attention mechanisms (see Table 4.1). As illustrated in Figure 4.1, the GNO in the encoder can be viewed as a domain- and discretization-agnostic alternative to the "patchify" operation. By injecting the input function representation into both the encoder and the cross-attention decoder, our framework benefits from both local and global receptive fields.

In the following, we outline the MINO architecture. We specifically focus on its use in the flow matching paradigm to learn velocity fields, as illustrated in Fig. 4.2. Although our primary application is generative modeling, the MINO architecture is general-purpose and can be used for other operator learning tasks, such as solving PDEs, with minor adjustments.

Problem Formulation. Within the flow matching paradigm, MINO takes three primary inputs: a noisy function $\mathbf{f}_t$, the set of its observation positions $\mathbf{f}_{\text{pos}}$, and the corresponding time step $\mathbf{t}$, along with any optional conditioning variables. Specifically, the input function $\mathbf{f}_t$ is represented by its values at N_{in} discrete locations, $\mathbf{f}_{\text{pos}} = \{p_i\}_{i=1}^{N_{\text{in}}}$. These locations are a discretization of a continuous domain $D \subset \mathbb{R}^{P_{\text{dim}}}$. Note that N_{in} can be different per sample. The codomain (channel) of $\mathbf{f}_t$ has dimension f_{dim} , resulting in an input tensor of function values with shape $[f_{\text{dim}}, N_{\text{in}}]$. The output is another function (velocity field $\mathbf{v}_t$) at these same locations. In the context of flow matching or diffusion, the output dimension matches the input,

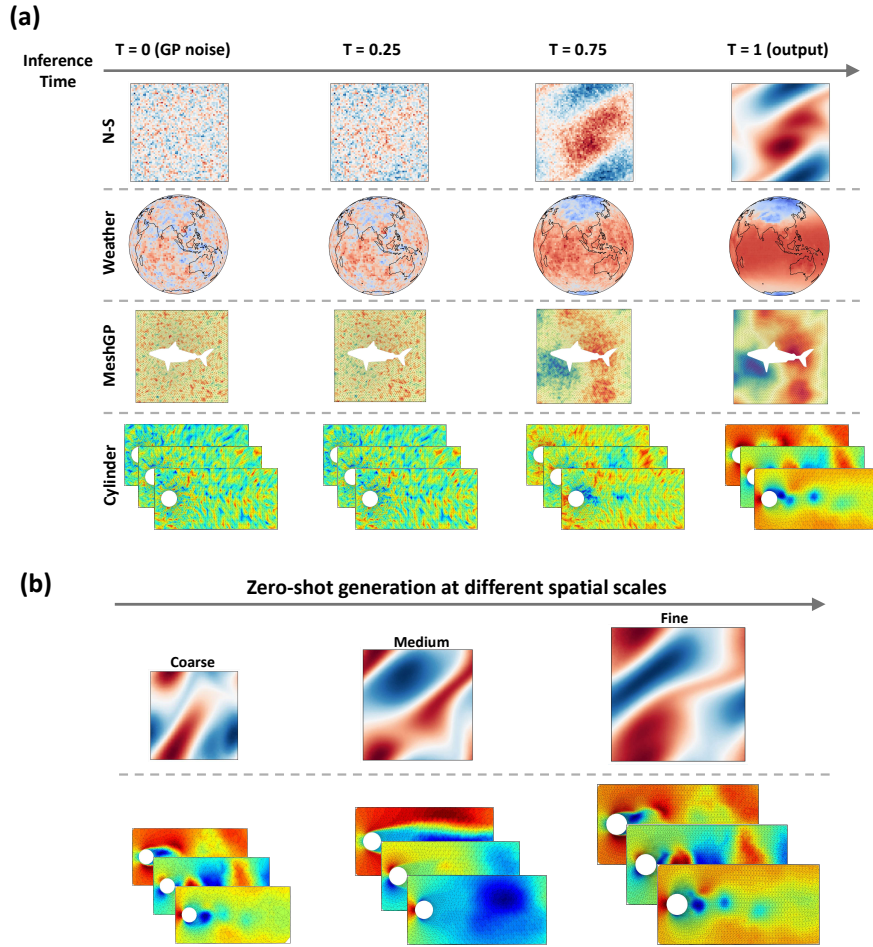


Figure 4.2: Inference and zero-shot Generation with MINO. (a) MINO gradually transforms a GP sample to the data sample under flow matching paradigm. (b) Zero-shot generation at varying spatial scales by directly transforming finer GP samples to finer data samples.

so the target output also has a shape of $[g_{\text{dim}}, N_{\text{in}}]$, where $g_{\text{dim}} = f_{\text{dim}}$.

The MINO framework is inspired by the encoder-processor-decoder structure of prior architectures like GINO and UPT. Our model consists of three main components: (i) a geometry encoder, (ii) an (optional) latent-space processor, and (iii) a cross-attention decoder. We introduce several key modifications to this design that significantly improve performance on functional generative tasks, as detailed in the following subsections.

Geometry Encoder. First, we apply sinusoidal embeddings (Vaswani et al., 2017; Peebles et al., 2022) to the time step $\mathbf{t}$ and the observation positions $\mathbf{f}_{\text{pos}}$, yielding $\mathbf{t}_{\text{emb}}$

and $\mathbf{p}_{\text{emb}}$ respectively, as shown in Eq. 4.2. The input function $\mathbf{f}_t$ is then concatenated with the embedded positions $\mathbf{p}_{\text{emb}}$ along the codomain before being passed to a GNO layer; this maps the function from its potentially irregular discretization $\{p_i\}_{i=1}^{N_{\text{in}}}$ to a latent representation on a predefined regular grid of query points $\{p_j^{\text{query}}\}_{j=1}^{N_{\text{node}}}$. The GNO mapping is based on neighbor-searching within a fixed radius r , where $N_{\text{node}} \ll N_{\text{in}}$. Following this GNO layer, a linear layer projects its output $\mathbf{f}_t^{\text{E}}$ into $\mathbf{h}_0^{\text{E}}$ of shape $[L_{\text{dim}}, N_{\text{node}}]$.

This initial latent representation $\mathbf{h}_0^{\text{E}}$ is then processed by M_1 blocks of Multi-Head Cross-Attention (MHCA). Importantly, the key-value (KV) pair for all attention blocks is fixed as $\mathbf{h}_0^{\text{E}}$, while the query (Q) is the output of the previous MHCA layer $\mathbf{h}_{j-1}^{\text{E}}$, and the time embedding $\mathbf{t}_{\text{emb}}$ is used for conditioning. We found this configuration yields better performance than the standard self-attention used in DiT, with comparisons provided in Appendix C.5. Thus, for this entire component of the architecture, we have the following,

$$\mathbf{p}_{\text{emb}} = \text{Emb}(\mathbf{f}_{\text{pos}}), \quad \mathbf{t}_{\text{emb}} = \text{Emb}(\mathbf{t}) \quad (4.2)$$

$$\mathbf{f}_t^{\text{E}} = \text{GNO}(\text{concat}(\text{MLP}(\mathbf{f}_t), \mathbf{p}_{\text{emb}})), \quad \mathbf{h}_0^{\text{E}} = \text{Linear}(\mathbf{f}_t^{\text{E}}) \quad (4.3)$$

$$\mathbf{h}_j^{\text{E}} = \text{MHCA}(Q = \mathbf{h}_{j-1}^{\text{E}}, KV = \mathbf{h}_0^{\text{E}}, C = \mathbf{t}_{\text{emb}}), \quad j = 1, \dots, M_1. \quad (4.4)$$

The output of the geometry encoder has a fixed size of $[L_{\text{dim}}, N_{\text{node}}]$, which is independent of the manner in which the input function $\mathbf{f}_t$ is discretized.

Latent-Space Processor. Given that the latent representation $\mathbf{h}_{M_1}^{\text{E}}$ has a fixed size, any suitable neural network (NN) can act as the processor. In our implementation, we use a Diffusion U-Net, known for being a strong backbone in generative modeling. It is important to note that this processor is optional; MINO remains a well-defined neural operator without it, although we found that its inclusion generally improves performance. The processor’s input is conditioned on $\mathbf{t}_{\text{emb}}$, whereas its output, $\mathbf{h}^{\text{P}}$, maintains the same shape as its input,

$$\mathbf{h}^{\text{P}} = \text{NN}(\mathbf{h}_{M_1}^{\text{E}}, \mathbf{t}_{\text{emb}}) \quad (4.5)$$

Cross-Attention Decoder. The decoder employs a different cross-attention mechanism from that of the encoder to enable mapping the latent representation back into the output velocity field at the original observation locations. Specifically, $\mathbf{f}_t$ is first processed by an MLP and then concatenated with $\mathbf{p}_{\text{emb}}$ to form $\mathbf{f}_t^{\text{D}}$. In a single cross-attention block, $\mathbf{f}_t^{\text{D}}$ serves as the query (Q), while the key-value (KV) pair is

taken from the output of the optional processor $\mathbf{h}^{\mathbf{P}}$ (or $\mathbf{h}_{\mathbf{M}_1}^{\mathbf{E}}$ if the processor is omitted). The time embedding $\mathbf{t}_{\text{emb}}$ provides conditioning. Finally, a LayerNorm and a linear transformation are applied to the output of this attention block to produce the estimated velocity field $\mathbf{v}_t$.

$$\mathbf{f}_t^{\mathbf{D}} = \text{MLP}(\text{concat}(\text{MLP}(\mathbf{f}_t), \mathbf{p}_{\text{emb}})) \quad (4.6)$$

$$\mathbf{h}_1^{\mathbf{D}} = \text{MHCA}(Q = \mathbf{f}_t^{\mathbf{D}}, KV = \mathbf{h}^{\mathbf{P}}, C = \mathbf{t}_{\text{emb}}) \quad (4.7)$$

$$\mathbf{v}_t = \text{Linear}(\text{LayerNorm})(\mathbf{h}_1^{\mathbf{D}}) \quad (4.8)$$

A detailed comparison of the MINO framework to prior neural operator architectures is provided in Appendix C.3, with an ablation study of our model’s components in Appendix C.5

4.4 Experiments

In this section we empirically evaluate MINO and other baselines on a suite of functional generative benchmarks under the OFM (Shi et al., 2026b) paradigm due to its concise formulation and SOTA performance among functional generative paradigms. For each task we parameterize the velocity field $v_\theta(f_t, t)$ with different neural-operator backbones to study how the choice of architecture affects performance. The benchmarks cover both regular and irregular grids, providing a comprehensive assessment. To the best of our knowledge, this is the first systematic comparison of modern neural-operator architectures on function-generation problems. We evaluate two variants of our proposed architecture: (i) MINO-U: a Diffusion U-Net as the latent space processor, and (ii) MINO-T: a pure Transformer-based variant without the latent space processor, where the encoder receives additional cross-attention blocks.

Table 4.2: Summary of experimental benchmarks, covering diverse domains with both regular and irregular grids

Geometry	Datasets	Domain	Co-domain	Mesh	Training Samples	Test Samples
Regular Grid	Navier Stokes	$D \subset \mathbb{R}^2$	1	4,096	30,000	5,000
	Shallow Water	$D \subset \mathbb{R}^2$	1	4,096	30,000	5,000
	Darcy Flow	$D \subset \mathbb{R}^2$	1	4,096	8,000	2,000
Irregular Grid	Cylinder Flow	$D \subset \mathbb{R}^2$	3	1,699	30,000	5,000
	MeshGP	$D \subset \mathbb{R}^2$	1	3,727	30,000	5,000
	Global Climate	$\mathbb{S}^2 \subset \mathbb{R}^3$	1	4,140	9,676	2,420

Datasets. We curated a benchmark of six challenging functional datasets to evaluate performance across different domain types: (i) Rectangular Domains and Regular

Grids: We use Navier-Stokes (Li et al., 2020b), Shallow Water equation (Takamoto et al., 2022), and Darcy Flow (Takamoto et al., 2022) datasets. (ii) Irregular Domains and Grids: We use the Cylinder Flow dataset (Han et al., 2022), a synthetic Mesh-GP dataset on irregular meshes (Zhao et al., 2022), and a real-world global climate dataset (Dupont et al., 2021). These datasets are summarized in Table 4.2. Detailed descriptions of the datasets and their associated preparation are provided in Appendix C.1.

Baselines. To ensure a thorough comparison, we selected several state-of-the-art neural operators tailored to different domains : (i) For irregular grids, we benchmark against leading architectures including the Universal Physical Transformer (UPT) (Alkin et al., 2024), Latent Neural Operator (LNO) (Wang et al., 2024b), Transolver (Wu et al., 2024), and Geometry-Informed Neural Operator (GINO) (Li et al., 2023). (ii) For regular grids, we further include the Fourier Neural Operator (FNO) as a strong, established baseline. To ensure a fair assessment, the parameter counts (or computational budgets) of all baseline models were carefully matched to our variants. It is important to note that, with the exception of FNO, all evaluated models (including our MINO variants) are designed to be domain-agnostic and process regular and irregular grids identically. A detailed description of the the baseline implementation is provided in Appendix C.1.

Metrics. A significant current challenge in the field of functional generative models is the lack of standardized evaluation protocols. Previous studies often rely on dataset-specific metrics, which prohibits fair and direct comparison between models (Rahman et al., 2022; Kerrigan et al., 2022; Shi et al., 2024a; Lim et al., 2025). To address this gap, our work validates two robust and general metrics suitable for estimating the distance between the learned probability measure ν (from which generated samples X are drawn) and the target probability measure μ (from which test samples (dataset) Y are drawn).

The first metric is the Sliced Wasserstein Distance (SWD) (Bonneel et al., 2015; Han, 2023). SWD provides a computationally efficient yet powerful alternative to the exact Wasserstein distance, which is often intractable in high dimensions. It works by projecting high-dimensional distributions onto random 1D lines and then averaging the simpler 1D Wasserstein distances. This approach converges to the true Wasserstein distance as the number of projections increases and provides a robust way to compare the geometric structure of two probability measures. For two measures μ and ν in Hilbert space, SWD is defined as (Definition 2.5 of Han

(2023)):

$$\text{SWD}_p^\gamma(\mu, \nu) = \left(\frac{1}{\gamma_S(S)} \int_{\|\theta\|=1} W_p^p(\theta_\# \mu, \theta_\# \nu) \gamma_S(d\theta) \right)^{\frac{1}{p}} \quad (4.9)$$

Where base measure γ_S is strictly positive Borel measure defined on a sphere S , W_p is the Wasserstein-p distance, and $\theta_\# \mu, \theta_\# \nu$ denote the pushforward measure of μ, ν under the projection θ separately. Further details are provided in Appendix C.2.

The second metric is the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012). Let $\mathcal{X} = L^2(D; \mathbb{R}^{f_{dim}})$ and choose a bounded, characteristic kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ with reproducing-kernel Hilbert space (RKHS) $(\mathcal{H}_k, \langle \cdot, \cdot \rangle_k)$. Denote the mean embeddings $m_\mu = \mathbb{E}[k(X, \cdot)]$, $m_\nu = \mathbb{E}[k(Y, \cdot)]$. Then MMD is defined as

$$\text{MMD}_k(X, Y) = \|m_\nu - m_\mu\|_{\mathcal{H}_k} \quad (4.10)$$

Because k is characteristic, $\text{MMD}_k = 0$ if and only if $\mu = \nu$. In Appendix C.2, we show how to calculate SWD and MMD given discretized function samples, and empirically verify that these metrics are valid, robust, and sample-efficient for functional generative models.

Results. A visualization of the performance of MINO is provided in Figure 4.3 and as shown in Table 4.4, on regular grid benchmarks, our MINO variants achieve SOTA performance: MINO-U achieves best performance on the Navier-Stokes and Shallow Water datasets, while MINO-T excels on the Darcy Flow benchmark. In contrast, several baseline neural operators (GINO, LNO, UPT) perform poorly. Further analysis in Appendix C.3, C.4 suggests this is because these models primarily learn low-frequency information and fail to capture the high-frequency components essential for these generative tasks. FNO and Transolver achieve comparable, albeit inferior, results to our models. Notably, beyond superior accuracy, our MINO variants are also significantly more efficient than Transolver. As detailed in Table 4.5, they require substantially less computation to train, are more GPU memory-efficient, and achieve up to a 3.0x speedup during inference, demonstrating the efficiency of the MINO architecture.

For tasks on irregular grids, our models continue to demonstrate top performance. MINO-U achieves the best results on the Cylinder Flow benchmark, while MINO-T is the top performer on the Mesh-GP and Global Climate benchmarks. Then, We perform an ablation study, which confirm the effectiveness the each component of our architecture, as detailed in Appendix C.5. Last, we provide additional experiments in Appendix C.6 and C.7 to further demonstrate the superior performance of MINO.

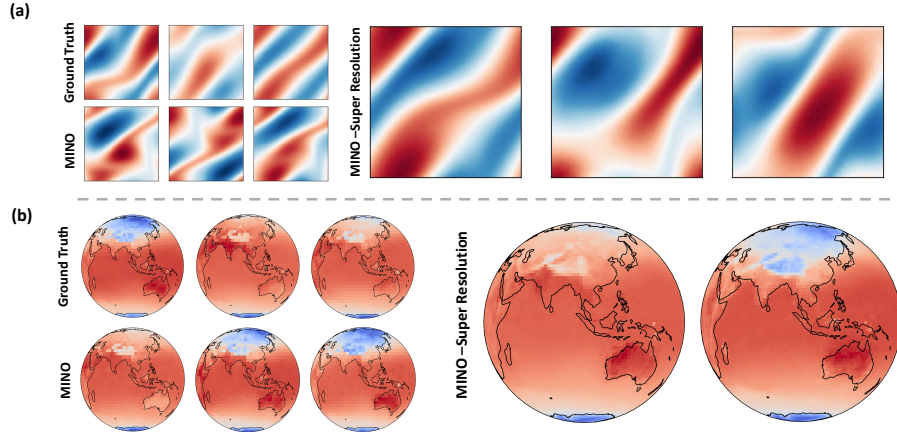


Figure 4.3: Visualization of generation and zero-shot super-resolution by MINO-U. (a) Navier-Stokes samples generated on the original mesh (4,096 nodes) and a finer mesh (25,600 nodes). (b) Global Climate sample generated on the original mesh (4,140 nodes) and a finer mesh (16,560 nodes).

Table 4.3: Generation performance on benchmarks with irregular grids. Best performance in bold

Dataset →		Cylinder Flow		Mesh-GP		Global Climate	
Model ↓	Metric →	SWD	MMD	SWD	MMD	SWD	MMD
GINO		$4.6 \cdot 10^{-1}$	$3.1 \cdot 10^{-1}$	$1.1 \cdot 10^0$	$3.6 \cdot 10^{-1}$	$6.9 \cdot 10^{-1}$	$5.1 \cdot 10^{-1}$
UPT		$7.3 \cdot 10^{-1}$	$5.4 \cdot 10^{-1}$	$4.2 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$7.4 \cdot 10^{-1}$	$5.4 \cdot 10^{-1}$
Transolver		$2.5 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$9.1 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$
LNO		$5.3 \cdot 10^{-1}$	$3.8 \cdot 10^{-1}$	$3.5 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$	$4.9 \cdot 10^{-1}$
MINO-T (ours)		$2.9 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$
MINO-U (ours)		$2.3 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$7.2 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$

4.5 Conclusions

In this paper, we introduce the Mesh-Informed Neural Operator (MINO), a novel backbone for functional generative models that leverages graph neural operators and cross-attention mechanisms. MINO operates directly on arbitrary meshes, addressing the reliance on grid-dependent architectures like FNO and thereby enabling high-fidelity generation on complex and irregular domains. To complement this architectural advance, we validate the Sliced Wasserstein Distance (SWD) and Maximum Mean Discrepancy (MMD) as general-purpose metrics for fair and standardized model comparison. Our comprehensive experiments confirm the success of this approach: MINO variants achieve state-of-the-art performance across a diverse suite of benchmarks while being substantially more computationally efficient than strong competitors.

Additionally, the MINO framework helps bridge the gap between math-driven op-

Table 4.4: Generation performance on benchmarks with regular grids. Evaluation metrics include the Sliced Wasserstein Distance (SWD), Maximum Mean Discrepancy (MMD), and Mean Squared Error (MSE) for spectra, autocovariance, and point-wise density. Best performance is indicated in bold.

Datasets	Model ↓ Metric →	SWD	MMD	Spectra-MSE	Autocov-MSE	Density-MSE
Navier-Stokes	GINO	$4.6 \cdot 10^{-1}$	$3.7 \cdot 10^{-1}$	$1.9 \cdot 10^2$	$4.6 \cdot 10^{-4}$	$1.9 \cdot 10^{-2}$
	UPT	$6.0 \cdot 10^{-1}$	$4.7 \cdot 10^{-1}$	$2.5 \cdot 10^3$	$1.5 \cdot 10^{-2}$	$4.0 \cdot 10^{-3}$
	Transolver	$5.3 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$1.1 \cdot 10^2$	$6.7 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
	LNO	$5.2 \cdot 10^{-1}$	$3.8 \cdot 10^{-1}$	$2.0 \cdot 10^4$	$1.1 \cdot 10^{-1}$	$5.0 \cdot 10^{-4}$
	FNO	$3.2 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$5.1 \cdot 10^1$	$2.2 \cdot 10^{-4}$	$6.9 \cdot 10^{-5}$
	MINO-T (Ours)	$4.0 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	$1.9 \cdot 10^1$	$6.8 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$
	MINO-U (Ours)	$2.8 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$1.3 \cdot 10^1$	$9.2 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$
Shallow Water	GINO	$7.3 \cdot 10^{-1}$	$5.1 \cdot 10^{-1}$	$1.5 \cdot 10^2$	$1.8 \cdot 10^{-1}$	$9.9 \cdot 10^{-1}$
	UPT	$8.6 \cdot 10^{-1}$	$5.9 \cdot 10^{-1}$	$3.1 \cdot 10^2$	$4.0 \cdot 10^{-3}$	$1.0 \cdot 10^0$
	Transolver	$1.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$1.2 \cdot 10^{-6}$	$9.0 \cdot 10^{-4}$
	LNO	$8.7 \cdot 10^{-1}$	$6.5 \cdot 10^{-1}$	$2.1 \cdot 10^2$	$1.4 \cdot 10^0$	$1.0 \cdot 10^0$
	FNO	$1.0 \cdot 10^{-2}$	$9.4 \cdot 10^{-3}$	$7.9 \cdot 10^{-3}$	$2.8 \cdot 10^{-6}$	$1.0 \cdot 10^{-4}$
	MINO-T (Ours)	$9.8 \cdot 10^{-3}$	$8.7 \cdot 10^{-3}$	$1.6 \cdot 10^{-2}$	$5.2 \cdot 10^{-6}$	$2.0 \cdot 10^{-5}$
	MINO-U (Ours)	$4.0 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$4.3 \cdot 10^{-3}$	$5.1 \cdot 10^{-7}$	$1.7 \cdot 10^{-5}$
Darcy Flow	GINO	$4.0 \cdot 10^{-1}$	$3.0 \cdot 10^{-1}$	$5.5 \cdot 10^3$	$5.4 \cdot 10^{-2}$	$1.8 \cdot 10^{-3}$
	UPT	$8.7 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$	$6.0 \cdot 10^4$	$4.0 \cdot 10^0$	$7.5 \cdot 10^{-3}$
	Transolver	$2.2 \cdot 10^{-1}$	$8.9 \cdot 10^{-2}$	$4.8 \cdot 10^3$	$3.4 \cdot 10^{-1}$	$3.0 \cdot 10^{-4}$
	LNO	$5.5 \cdot 10^{-1}$	$3.8 \cdot 10^{-1}$	$2.7 \cdot 10^3$	$2.4 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$
	FNO	$1.1 \cdot 10^{-1}$	$3.6 \cdot 10^{-2}$	$1.2 \cdot 10^3$	$8.4 \cdot 10^{-2}$	$8.9 \cdot 10^{-5}$
	MINO-T (Ours)	$8.9 \cdot 10^{-2}$	$3.4 \cdot 10^{-2}$	$9.1 \cdot 10^2$	$4.4 \cdot 10^{-2}$	$9.4 \cdot 10^{-5}$
	MINO-U (Ours)	$9.5 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$	$1.0 \cdot 10^3$	$7.7 \cdot 10^{-2}$	$3.8 \cdot 10^{-5}$

Table 4.5: Computational efficiency and performance comparison between MINO variants and Transolver during training and inference on Navier-Stokes benchmark.

Training Phase →	Params	GPU mem. (per sample)	Training time (per epoch)	Speedup
Transolver	15.0 M	0.913 GB	734 s	1×
MINO-T (ours)	21.5 M	0.429 GB	277 s	2.6×
MINO-U (ours)	19.2 M	0.419 GB	279 s	2.6×
Inference Phase →	SWD	MMD	Generation time (per sample)	Speedup
Transolver	0.053	0.051	1.49 s	1×
MINO-T (ours)	0.040	0.036	0.51 s	2.9×
MINO-U (ours)	0.028	0.019	0.49 s	3.0×

erator learning and modern deep learning practice, unlocking the potential to apply high-fidelity generative modeling to a broader range of complex scientific problems. We believe our contributions will foster more rigorous and rapid advancement in the field of functional generative modeling.

Chapter 5

SCALABLE PHYSICS-INSPIRED GENERATIVE MODELING OF COHERENT REGIONAL EARTHQUAKE WAVEFIELDS

This chapter presents Ground-Motion Flow (GMFlow), a physics-inspired functional generative model for regional earthquake ground-motion synthesis. GMFlow aims to generate scenario-conditioned, three-component spatiotemporal wavefields with realistic frequency content, spatial variability, and coherent wave-propagation structure.

GMFlow follows a two-stage synthesis pipeline. It first generates the dominant low-frequency structure of the regional wavefield in a physics-aligned latent space using conditional flow matching. A super-resolution neural operator then lifts this generated low-frequency realization to the full spatiotemporal wavefield. This design avoids direct generation in the full high-dimensional physical space while preserving physically meaningful wavefield coherence.

The framework is evaluated using large-scale physics-based simulations in the San Francisco Bay Area, including both point-source and finite-rupture scenarios. The results show that GMFlow reproduces key ground-motion features, including peak ground velocity, Fourier amplitude spectra, magnitude scaling, spatial variability, and normalized cross-correlation structure. Once trained, the model generates coherent regional wavefield ensembles in seconds, providing orders-of-magnitude acceleration relative to direct numerical simulation.

The contents of this chapter are adapted from our work (Shi et al., 2026a):

Shi, Yaozhong, Grigorios Lavrentiadis, Konstantinos Tsalouchidis, Zachary E Ross, David McCallen, Caifeng Zou, Kamyar Azizzadenesheli, and Domniki Asimaki (2026a). “Large-Scale 3D Ground-Motion Synthesis with Physics-Inspired Latent Operator Flow Matching”. In: arXiv preprint arXiv:2603.17403.

Additional details for this chapter are provided in Appendix D. The code associated with this chapter is available at <https://github.com/yzshi5/GMFlow>.

5.1 Introduction

Managing and operating modern infrastructure systems, such as electric power grids, nuclear power plants, and other critical lifeline networks, requires rigorous evaluation of the performance and risk of systems composed of multiple, interdependent components (Guidotti et al., 2016). In seismic-prone regions, designing these assets for resilient performance under strong shaking is crucial, as their failure can trigger cascading disruptions, endanger lives, and impose major economic losses. Consequently, the design process requires scenario-based hazard characterization and the quantification of within-event and between-event uncertainties to calculate the joint exceedance of performance thresholds with ground-motion time histories (Miller et al., 2015; Baker et al., 2021). In principle, event-level regional ground-motion time histories can be obtained either from empirical ground-motion databases or from physics-based simulations (Fig 5.1). In practice, however, both approaches face fundamental limitations. Observational records of large-magnitude earthquakes remain sparse and unevenly distributed, and the application of scenario-specific filtering criteria (e.g., magnitude, distance, mechanism, site conditions) further reduces the number of usable time histories. Physics-based rupture and wave-propagation simulations partially alleviate data scarcity by enabling direct modeling of scenarios of interest at prescribed spatial resolution. Nevertheless, solving the elastodynamic wave equation in heterogeneous media to generate broadband ground motions at regional scale remains computationally intensive. Moreover, the resolution and fidelity of simulated ground motions are inherently constrained by the quality and resolution of the input models (e.g., three-dimensional shear-wave velocity structure), which often lack sufficient detail to accurately propagate high-frequency content critical for civil infrastructure applications (Rodgers et al., 2020; Pitarka et al., 2013; Lavrentiadis et al., 2025). These limitations motivate a new paradigm for generating scenario-consistent broadband waveforms, namely machine-learning-based surrogate approaches capable of rapidly producing ensembles of scenario-conditioned, spatially coherent ground-motion time histories while preserving key physical characteristics. At the same time, such approaches provide a pathway toward hybrid frameworks in which physics-based simulations offer spatial completeness and scenario control, while observational data from both small and large events can be seamlessly incorporated to improve realism, particularly for high-frequency behavior and nonlinear site response.

Recent studies have explored conditional generative models capable of synthesizing seismic waveforms across broadband frequencies at specific sites (Florez et

al., 2022; Shi et al., 2024b; Palgunadi et al., 2024). Although promising, these approaches typically treat ground motions as independent one-dimensional time series at isolated sites, thereby failing to capture the within-event spatial correlations necessary for analyzing distributed systems; Extending the generative framework to synthesize coherent regional wavefields over large and continuous spatial domains requires a fundamental shift in problem formulation, introducing significant challenges in learning algorithm design, model architecture, and computational scalability during both training and inference, all of which must be addressed to yield a tractable and physically consistent solution.

To address these challenges, we propose Ground-Motion Flow (GMFlow), building on the machine-learning framework introduced in (Liu et al., 2022; Rombach et al., 2022; Shi et al., 2026b). To the best of our knowledge, GMFlow is the first large-scale spatiotemporal ground-motion generative model formulated directly in function space. The method leverages neural operators (Li et al., 2020b) and follows a simple but effective two-stage design (Fig. 5.2): it first generates a low-frequency regional wavefield that captures the dominant large-scale spatiotemporal structure and coherence of earthquake shaking, and then reconstructs a full-band wavefield at the target resolution from that low-frequency realization. In the first stage, an autoencoding operator maps the low-frequency wavefield into a compact latent representation, and a conditional flow-matching model efficiently samples from this latent space to generate spatially coherent regional realizations. In the second stage, a discretization-agnostic super-resolution neural operator reconstructs the full-band wavefield conditioned on the low-frequency field, recovering higher-frequency effects associated with scattering, heterogeneity, and localized amplification. Together, these components form an end-to-end resolution-agnostic generative pipeline that bypasses the computational bottleneck of direct high-dimensional generation while still producing physically consistent wavefields across different scales. We demonstrate that GMFlow achieves unprecedented computational scalability, providing approximately a 10,000-fold acceleration in inference relative to physics-based simulations, enabling the training of full regional ground-motion generative models for the San Francisco Bay Area on a single workstation GPU, while also appropriately capturing the complex multiscale physics of large-magnitude earthquakes; this enables large ensembles of scenario-consistent ground motions, which are important for uncertainty quantification and infrastructure risk assessment.

A central feature of GMFlow is the introduction of a *Physics-Aligned Subspace*,

in which the intermediate generation step is performed in a physically meaningful low-frequency representation. This design greatly reduces the computational cost of training and inference. In addition, because GMFlow is formulated as a functional generative model, it supports zero-shot super-resolution, allowing the generated wavefields to be evaluated at spatial resolutions finer than those used during training. Such resolution flexibility is especially important in scientific machine learning, where predictions often need to be queried on refined grids for downstream analysis. More broadly, we envision GMFlow serving as a foundational framework for the generative modeling of diverse large-scale spatiotemporal natural-hazard phenomena, including hurricanes, wildfires, tsunamis, and volcanic eruptions (Schwerdtner et al., 2024; Yu et al., 2025; Price et al., 2025), thereby advancing uncertainty quantification, risk mitigation, and resilient infrastructure planning.

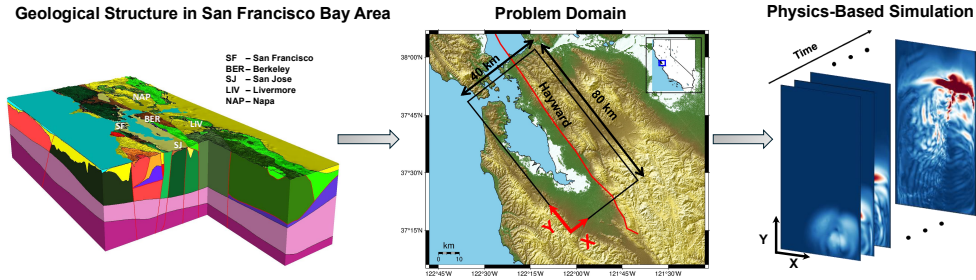


Figure 5.1: **Workflow for physics-based ground-motion simulation in the San Francisco Bay Area.** (left) A 3D representation of the heterogeneous geological structure beneath the Bay Area, adopted from <https://www.usgs.gov/media/images/3d-geologic-model> with permission. (center) The problem (computational) domain used for the numerical simulations. (right) An example of finite-fault event simulation.

5.2 Results

Simulation Dataset and Evaluation Framework

We evaluate the performance of GMFlow using a high-fidelity simulation dataset generated over the high-resolution domain of the U.S. Geological Survey San Francisco Bay Area three-dimensional seismic velocity model (SFVM). The computational domain spans 80×40 km (Figure 5.1). The dataset comprises 5,300 simulated events, including 2,100 M_w 4.4 point-source earthquakes distributed across the entire domain, and 3,200 finite-rupture events with magnitudes M_w 6.0 and M_w 7.0 located on the Hayward Fault. The double-couple mechanism for the point-source events is based on the M_w 4.4 2018 Berkeley earthquake. Model performance is

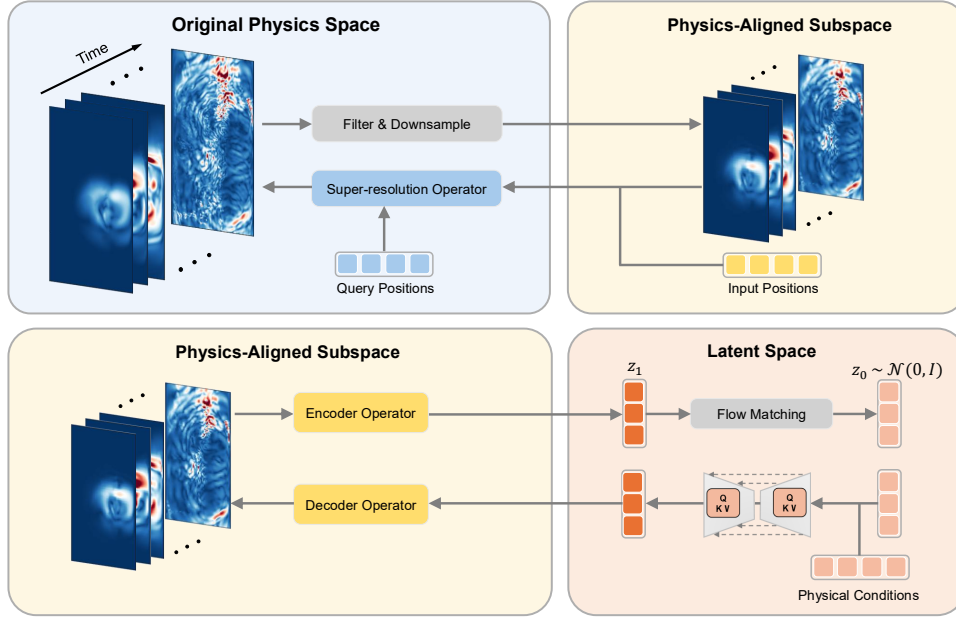


Figure 5.2: **GMFlow model architecture and two-stage synthesis pipeline.** **(top)** The original surface wavefield u is filtered and downsampled to a low-frequency representation u_f ; the reverse mapping is performed by a super-resolution neural operator. **(bottom)** An encoder-decoder neural operator encodes u_f to a latent code z_1 , and decodes it back to u_f ; conditional flow matching transports $z_0 \sim \mathcal{N}(0, I)$ to z_1 given physical conditions c .

assessed using a held-out test set of 300 events, stratified by magnitude class with 100 samples per class. All events were simulated using the SW4 platform (McCallen et al., 2021b; McCallen et al., 2024) using USGS v21.1 SFBA velocity model (Aagaard et al., 2021), resolved to a maximum frequency of 1 Hz. For each scenario, the surface velocity field was recorded over grid spacing, recorded on a 256×128 surface grid. Time histories were sampled at 4 Hz over a duration of 24 s, corresponding to 96 temporal snapshots. The resulting data tensor for each event has dimensions $[3, 256, 128, 96]$, representing the three velocity components over space and time.

We train GMFlow as a conditional generative model to learn the distribution of full spatiotemporal surface velocity wavefields conditioned on event parameters, specifically hypocenter location and moment magnitude. Model performance is evaluated using metrics that assess: (i) amplitude- and frequency-based statistics, including both ensemble mean behavior and predictive uncertainty; and (ii) the fidelity with which the model reproduces the underlying spatial correlation and spatiotemporal coherence structure. Additional details are provided in Sections 5.4

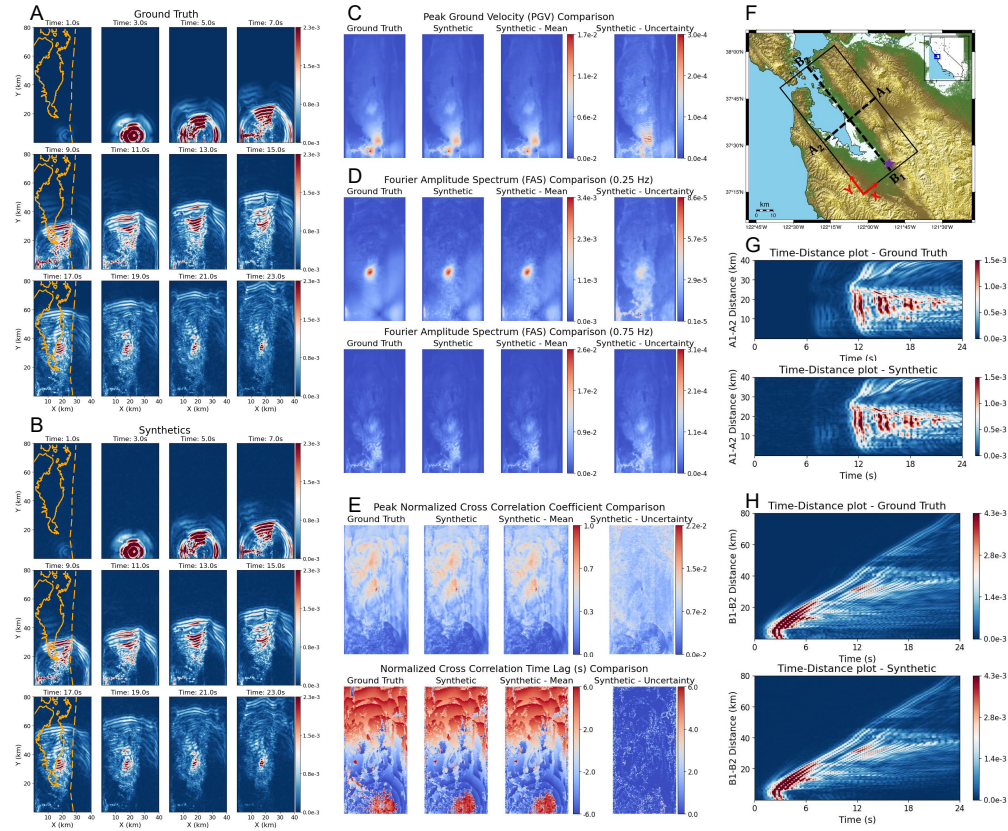


Figure 5.3: **Evaluation of GMFlow on a representative M_w 4.4 point-source scenario.** (A–B) Side-by-side comparison of the ground-truth velocity wavefield and a single stochastic realization generated by GMFlow at selected time steps (snapshots). Orange dash and solid lines represent Hayward fault and coastline, respectively. (C) Peak ground velocity (PGV; m/s), comparing the ground truth, one synthetic realization, the ensemble mean ($N = 100$), and the predictive uncertainty. (D) Spatial maps of the Fourier amplitude spectrum (FAS) at 0.25 Hz (top) and 0.75 Hz (bottom). (E) Normalized cross-correlation (NCC) for spatiotemporal coherence analysis, showing the peak correlation coefficient (top) and the corresponding time lag (bottom). (F) Geographic map of the computational domain, with epicenter (purple star) and the virtual seismic profiles A1–A2 and B1–B2 indicated. (G–H) Time–distance plots along profiles A1–A2 and B1–B2.

and 5.4.

Point-Source Scenario Evaluation

We first evaluate GMFlow on a randomly selected representative M_w 4.4 point-source event from the held-out set. The model is conditioned on the corresponding hypocenter location and magnitude, and an ensemble of 100 stochastic realizations is generated to assess predictive performance and variability.

Figure 5.3A–B compares the amplitudes of the three-component velocity vector

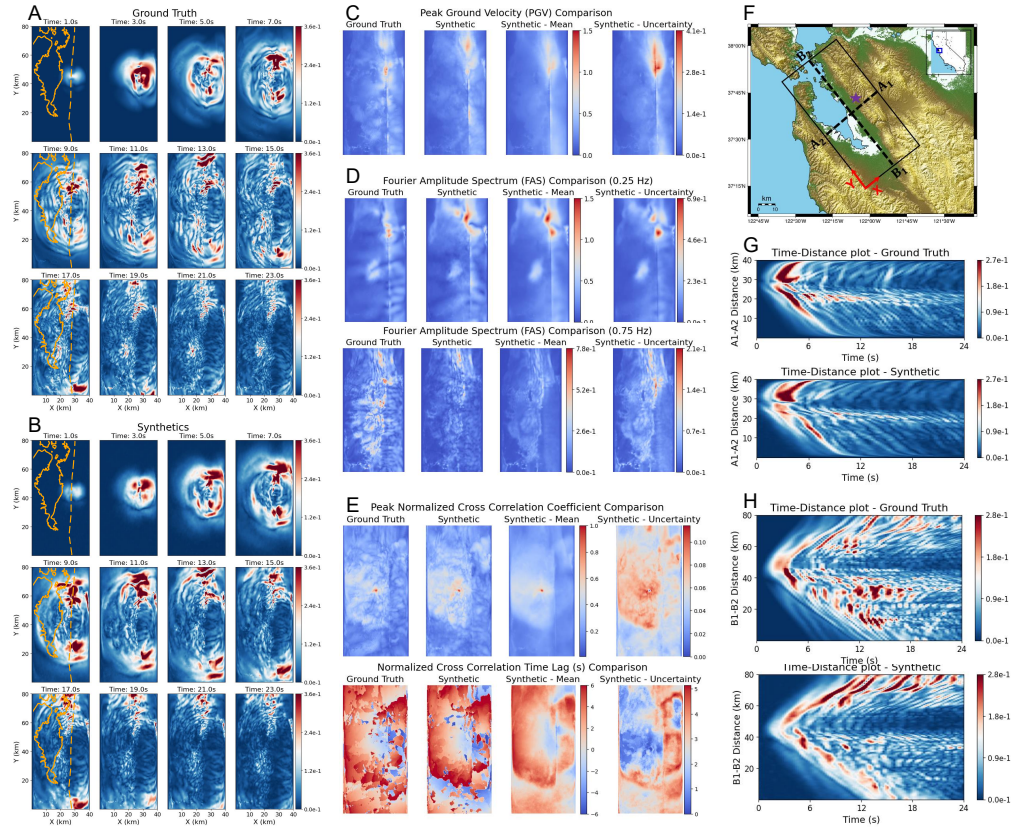


Figure 5.4: **Evaluation of GMFlow on a representative M_w 7.0 finite-rupture scenario.**

(Eq. 5.1) of the simulated event, treated as ground truth, with a randomly selected synthetic realization. Visually, GMFlow captures key features of the complex wavefield, including the primary wavefront geometry, localized scattering, and basin amplification (e.g., snapshot at 13 s), all of which arise from wave propagation in a heterogeneous medium. The generated realization also preserves coherent spatiotemporal structure across the domain.

We quantify ensemble statistics using Peak Ground Velocity (PGV), Fourier Amplitude Spectra (FAS), and Normalized Cross-Correlation (NCC) (defined in Section 5.4). PGV is used to evaluate the spatial distribution of ground-motion intensity (Figure 5.3C). The ensemble mean of the synthetic realizations closely matches the ground-truth field, indicating accurate recovery of large-scale amplitude patterns. The pixel-wise variability, quantified by the standard deviation across the 100 realizations, remains small for the point-source scenario, with a maximum value of 3.0×10^{-4} m/s, demonstrating low stochastic dispersion for this event class. This occurs because all M_w 4.4 share the same point-source mechanism. Figure 5.3D presents FAS maps at 0.25 Hz and 0.75 Hz to assess the model's ability to reproduce

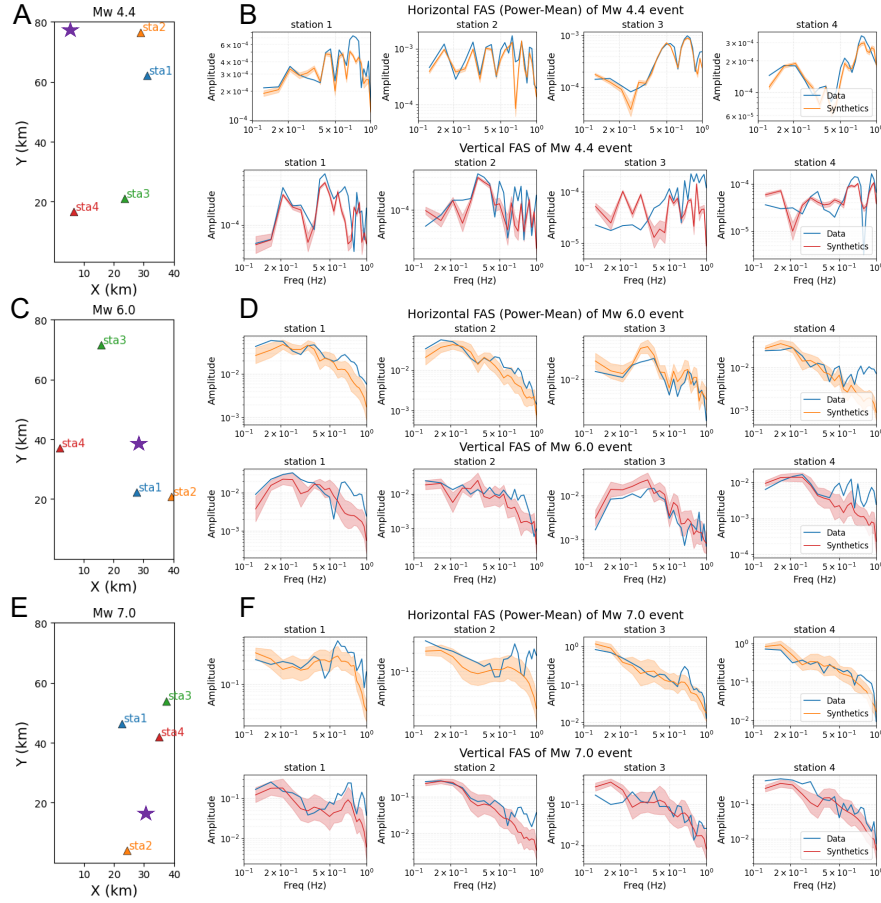


Figure 5.5: **Station-level spectral evaluation across magnitude range.** (A, C, E) Spatial distribution of four randomly selected validation stations (triangles) and the epicenter (star) for M_w 4.4, 6.0, and 7.0 scenarios. (B, D, F) Comparison of the horizontal (power-mean) and vertical Fourier Amplitude Spectra (FAS) for the ground truth (blue) and synthetic realizations (orange/red). The solid lines for the synthetics represent the geometric mean of $N = 100$ realizations, while the shaded regions indicate the ± 1 geometric standard deviation.

frequency-dependent spatial patterns. Consistent with the PGV results, GMFlow accurately preserves the spatial distribution of spectral amplitude across frequencies, showing that GMFlow captures both low- and mid-frequency spatial structure within the resolved bandwidth. Third, to assess spatiotemporal coherence and phase alignment, we compute NCC relative to a reference station. In Figure 5.3E, the reference station is placed at the geometric center of the domain.

From the NCC field, we extract two maps: the peak NCC coefficient (maximum similarity) and the corresponding time lag. Figure 5.3E shows close agreement between synthetic and ground-truth maps, indicating that the generated waveforms preserve both waveform similarity and phase alignment relative to the reference

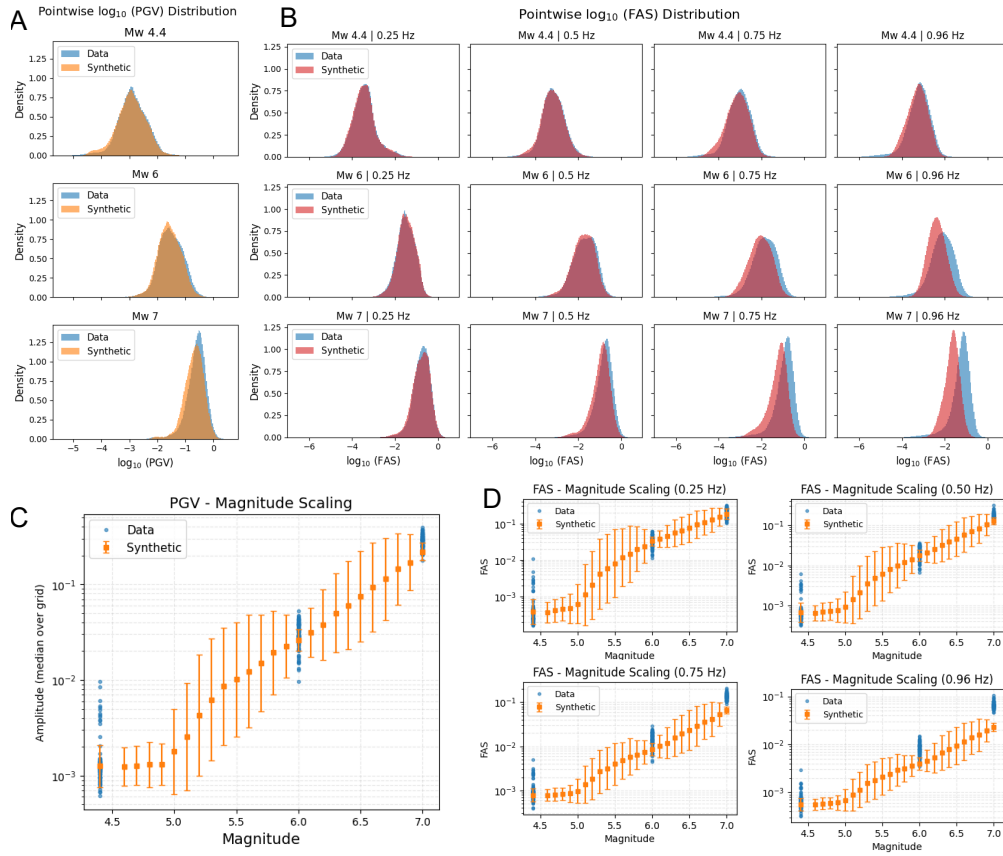


Figure 5.6: Statistical distribution and magnitude scaling laws of generated ground motions. (A) Pointwise distributions of $\log_{10}(\text{PGV})$ for M_w 4.4, 6.0, and 7.0 comparing held-out data to the same amount of synthetic realizations generated under identical conditions. (B) Pointwise distributions of $\log_{10}(\text{FAS})$ at 0.25, 0.5, 0.75, and 0.96 Hz. (C) PGV magnitude scaling from M_w 4.4–7.0 (0.1 magnitude increments), summarized by the spatial median per event over 100 realizations per bin. (D) Analogous magnitude scaling for FAS at the same frequencies; markers and error bars denote the ensemble geometric mean and standard deviation.

station. Importantly, the time-lag maps (bottom row) exhibit similar travel-time gradients, suggesting that GMFlow reproduces travel-time structure consistent with physically plausible wave propagation across the domain. Because a single central reference station may not fully characterize global coherence, we include additional NCC analyses with randomly selected reference stations in the Appendix D, which show consistent behavior. Finally, we evaluate the wavefield propagation consistency using time–distance plots along two virtual station arrays (Profiles A1–A2 and B1–B2 in Figure 5.3F). As shown in Figure 5.3G–H, the synthetic patterns display similar arrival times and apparent velocities for P-waves, S-waves, and surface waves, providing additional evidence of physically consistent regional wave propagation.

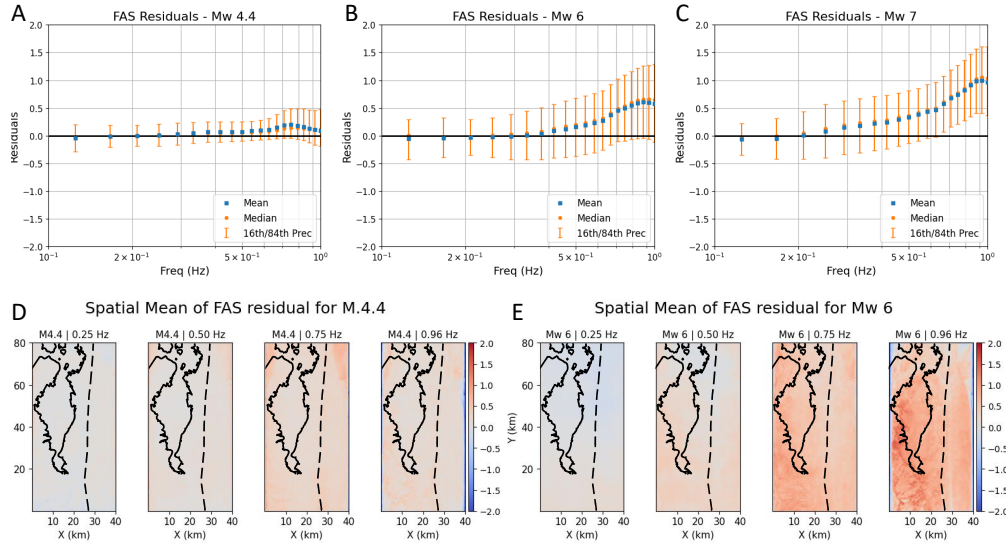


Figure 5.7: Frequency-dependent spectral residuals. (A–C) Residual statistics of horizontal Fourier amplitude spectra (FAS) as a function of frequency for M_w 4.4, 6.0, and 7.0, computed over 300 test events using ensembles of 100 synthetic realizations per event; markers show the mean and median, and error bars indicate the 16th/84th percentiles. (D–E) Spatial maps of the mean FAS residual at 0.25, 0.50, 0.75, and 0.96 Hz for M_w 4.4 and M_w 6.0 events. Black dash and solid lines represent Hayward fault and coastline, respectively.

Finite-Rupture Source Scenario Evaluation

We next evaluate GMFlow on a representative M_w 7.0 finite-rupture scenario on the Hayward Fault. Unlike the point-source case, these simulations involve finite-fault rupture with heterogeneous source-time-function realizations, posing a substantially more challenging generative task. Using the same evaluation protocol as in the point-source case, we first compare ground-truth wavefield snapshots with a representative synthetic realization (Figure 5.4A–B). The model captures several key finite-source characteristics: the wavefield is no longer radially symmetric, but is instead shaped by rupture propagation along the fault trace (approximately in the Y -direction). In particular, GMFlow reproduces spatially asymmetric energy patterns consistent with rupture propagation and directivity, as well as basin-related scattering across the domain. Figure 5.4C shows that the synthetic mean PGV reproduces the high-intensity zone observed in the ground truth. The associated uncertainty map reveals elevated aleatory variability near the fault and in regions of strong geological contrast, consistent with learning physically meaningful variability rather than collapsing to a mean response. The accurate representation of the frequency content is supported by the FAS maps in Figure 5.4D, which show that

GMFlow preserves the spatial distribution of spectral energy and its decay behavior across the domain within the modeled frequency band. Spatiotemporal coherence is further assessed via NCC. The peak NCC coefficient maps (Figure 5.4E) closely reproduce the main spatial decorrelation pattern in ground-truth: correlation decays with distance and medium heterogeneity, while remaining relatively high in the near field around the reference station. The corresponding time-lag maps exhibit similar travel-time gradients, suggesting that GMFlow synthesizes coherent wavefronts with realistic phase delays and apparent velocities under rupture-driven excitation.

Finally, time–distance plots along the virtual arrays (Figure 5.4G–H) provide an additional evaluation of propagation behavior. Compared with the impulsive arrivals of the point-source case, the M_w 7.0 wavefields exhibit prolonged wavetrains due to the finite rupture duration. GMFlow reproduces these extended phase arrivals and their moveout patterns, providing further evidence that the model captures the temporal evolution of large events. Collectively, these results show that GMFlow extends effectively to large-magnitude finite-rupture scenarios and captures key interactions between finite-fault source complexity and regional wave propagation.

Site-Specific Spectral Evaluation with Predictive Uncertainty

To assess local waveform fidelity beyond global spatial coherence, we examine site-specific FAS for three representative scenarios spanning the considered magnitude range (Figure 5.5A, C, E). For each scenario, we randomly select four stations covering a range of azimuths and source-to-site distances. For each event, we generate 100 realizations under identical conditioning and compute the ensemble geometric mean and standard deviation of both the horizontal power-mean FAS (Eq. 5.3) and the vertical FAS.

As shown in Figure 5.5B, D, and F, the synthetic spectra (orange) closely match the ground truth (blue) over the analyzed 0.1–1 Hz band for both horizontal and vertical components. GMFlow reproduces key spectral features, such as magnitude-dependent spectral shifts. The predicted geometric standard-deviation bands capture the overall spread of the ground-truth spectra across stations and magnitudes. Notably, the uncertainty for finite-rupture events (M_w 6.0–7.0) is substantially larger than for the M_w 4.4 point-source case, consistent with increased variability from source complexity, radiation pattern effects, and scattering in large events.

Magnitude Scaling

Beyond scenario-specific comparisons, we evaluate whether GMFlow reproduces the statistical distributions of ground motion and captures physically plausible scaling trends across magnitude. We first assess domain-wide ground-motion distribution by generating 300 synthetic realizations under the same conditioning as the held-out test events (for each event, we generate one synthetic). Figure 5.6A shows pointwise distributions of $\log_{10}(\text{PGV})$ aggregated over the spatial domain for the three magnitude classes. The synthetic distributions (orange) closely match the ground truth (blue) distribution across all magnitudes. Figure 5.6B reports analogous comparisons for $\log_{10}(\text{FAS})$ at four representative frequencies and shows broadly similar behavior. The 0.96 Hz band is chosen to probe the high-frequency limit while avoiding artifacts near the strict 1 Hz low-pass cutoff applied during training dataset preprocessing.

We next conduct a systematic scaling study (Figure 5.6C–D) by sampling magnitudes from M_w 4.4 to 7.0 in increments of 0.1 and generating 100 realizations per magnitude bin. For each realization, we compute PGV and FAS ordinates at selected frequencies across the entire spatial domain, and then evaluate the corresponding mean and standard deviation. The resulting plots report the ensemble mean and standard deviation of the intensity parameters for each magnitude level. The results exhibit a monotonic scaling trend consistent with physical expectations for increasing earthquake magnitude. Notably, the uncertainty (expressed as $\pm$ one standard deviation) is consistent with the training data for magnitudes (M_w 4.4, 6.0, and 7.0), while it broadens in the intermediate ranges where no direct training data are available. This behavior indicates reduced model confidence in interpolated (unseen) regimes while maintaining physically consistent scaling behavior.

A limitation appears in the high-frequency range for large-magnitude events. In the bottom-right panels of Figure 5.6D (0.75 Hz and 0.96 Hz), GMFlow underestimates the median amplitude for the M_w 7.0 case relative to the ground truth, although the scaling trend is accurately reproduced up to M_w 6.0. This discrepancy likely reflects the increased difficulty of representing high-frequency scattering associated with complex finite-fault ruptures within the current latent-space parameterization. One potential remedy would be to increase the dimensionality of the latent space in future training efforts; however, this would increase computational cost. Moreover, given the inherent uncertainty of numerical simulations at these higher frequencies, given the 3D velocity model resolution, such an increase was not pursued in the

present study in order to maintain computational tractability. Instead, in future work, we aim to integrate synthetic simulations with recorded ground motions from real earthquakes, enabling the model to learn high-frequency characteristics directly from observational data. Despite this limitation, the overall magnitude-scaling behavior remains physically consistent, with the ground-truth values generally falling within the predicted uncertainty bounds.

Spectral Residual Analysis

We perform a frequency-domain residual analysis (Eq. 5.6) to quantify systematic bias. For each held-out test event, we generate an ensemble of 100 stochastic realizations under identical conditional parameters. Figure 5.7A–C summarizes the aggregated residual statistics as a function of frequency (positive residuals indicate underestimation of spectral amplitude). For M_w 4.4 events (Figure 5.7A), residuals remain centered near zero across 0.1–1 Hz, with a tight spread, indicating negligible bias. For finite-rupture events (Figure 5.7B–C), the model maintains near-zero bias in the low-frequency range, but exhibits a systematic positive shift above ~ 0.5 Hz, reaching ~ 0.5 – 1.0 (natural-log units) near 1 Hz. This behavior is consistent with the previously noted limitation that GMFlow underestimates high-frequency spectral amplitudes for the large-magnitude scenarios. Given the deterministic nature of the velocity model and the fixed double-couple mechanism for the small-magnitude events, the predicted variability remains limited for these scenarios. In contrast, the broader uncertainty bands observed at larger magnitudes reflect the increased source complexity and enhanced wavefield scattering associated with finite-rupture events.

To assess the geographic distribution of these errors, we also examine spatial maps of the mean FAS residual over the domain. Figure 5.7D–E shows residual fields for representative M_w 4.4 and M_w 6.0 cases. For the M_w 4.4 case, the residual maps are nearly flat and centered around zero, indicating spatially unbiased predictions. For M_w 6.0, the high-frequency residuals show the expected positive bias but remain relatively uniform, without pronounced spatial concentrations. These results suggest that the dominant error mode is a frequency-dependent amplitude bias rather than a strongly spatially localized modeling failure.

To mitigate the observed frequency-dependent bias, we introduce an optional post-processing calibration step that applies a frequency-dependent amplitude correction to the generated spectra. This correction yields an approximately unbiased spectrum

in expectation while preserving the predicted phase structure and spatiotemporal coherence. As a result, agreement in FAS and magnitude-scaling behavior is improved. The full methodology and detailed evaluation of this calibration procedure are provided in the Appendix D. Importantly, this step should be viewed as a practical mitigation strategy rather than a substitute for improved generative modeling of high-frequency rupture processes and wavefield scattering.

5.3 Discussion

In this work, we introduced GMFlow, which to our knowledge is the first large-scale functional generative model for event-level regional ground-motion synthesis. GMFlow adopts a functional generative viewpoint, modeling the regional wavefield as a continuous spatiotemporal function.

To achieve practical scalability, we combine neural operators with latent-space generation via flow matching and structure the model around a physics-inspired factorization through a physics-aligned subspace. The approach first generates a low-frequency wavefield that captures large-scale wave-propagation and spatial coherence effects, and then applies a super-resolution operator to reconstruct the missing high-frequency content. By restricting stochastic generation to this low-frequency, and decomposing synthesis into two stages—coherent low-frequency generation followed by super-resolution, GMFlow reduces both training and sampling complexity. This factorization further enables optional intermediate physical corrections at the low-frequency stage, ensuring that generated samples remain within a physically plausible solution manifold.

Evaluated against withheld physics-based simulations for the San Francisco Bay Area demonstrates that GMFlow captures key multiscale features relevant to hazard analysis, including coherent regional wavefront patterns, realistic spatial variability in intensity measures, and frequency-dependent scaling behavior consistent with the target distribution. Empirically, GMFlow delivers orders-of-magnitude speedups over conventional numerical simulation, opening a path toward uncertainty-aware regional hazard workflows for distributed infrastructure systems. Finally, we provide extensive ablation studies and sensitivity analyses in the Appendix D to examine the effects of key architectural and hyperparameter choices, with particular emphasis on the rationale for selecting a 0.75 Hz cutoff frequency.

Limitations

Despite the strong empirical performance, several limitations remain. First, while GMFlow performs well for point-source events across the modeled frequency band, large finite-rupture events exhibit systematic attenuation near the upper end of the band. As shown in Fig. 5.7, the mean residual becomes increasingly positive above ~ 0.5 Hz, indicating that the current framework tends to underestimate high-frequency spectral amplitudes when rupture complexity and scattering are most pronounced. Extensive ablation studies in the Appendix D further show that this bias is not simply an artifact of the intermediate cutoff frequency. As a practical mitigation, we introduce an optional frequency-dependent calibration based on the mean spectral residual. Because the residual is approximately spatially uniform, the correction can be implemented as a single multiplicative scaling of each synthetic event’s spectral amplitudes across frequency, yielding an approximately unbiased spectrum in expectation. Although this adjustment improves engineering-relevant amplitude metrics, it does not replace the need for improved generative modeling of high-frequency rupture and scattering effects.

Second, although the functional formulation is in principle flexible, the current implementation relies on an FFT-based Fourier Neural Operator (FNO) trained on regular grids. This limits immediate applicability to complex geometries, topography-following or unstructured discretizations, and sparse irregular station networks, all of which are important for observation-driven hazard analysis. Addressing these limitations will require operator backbones that natively support irregular domains and point-set queries, together with training protocols that can fuse sparse observations and dense simulations within a unified functional generative framework.

Overall, we view GMFlow as a first step toward scalable generative modeling of regional ground-motion wavefields rather than as a deployment-ready end-to-end tool. Important challenges remain, including extension to broader frequency bands, richer finite-rupture physics, and principled calibration and adaptation to real recordings. Nevertheless, the framework demonstrates that uncertainty-aware regional synthesis can be made computationally tractable, opening a path toward next-generation hazard workflows for distributed infrastructure systems.

Computational Efficiency and Comparative Analysis

We evaluate the computational performance of GMFlow relative to both physics-based simulations and prior generative approaches. Given four scalar conditioning

variables (three hypocenter coordinates and earthquake magnitude), GMFlow generates a stochastic realization of a three-component spatiotemporal wavefield on a default discretization of $[3, 256, 128, 96]$. Each sample therefore contains approximately 9.4 million scalar values in total, corresponding to about 3.1 million space–time grid points with three waveform components at each point.

Once trained, GMFlow generates the full set of 5,300 events in 0.386 GPU-hours on a single workstation GPU (NVIDIA RTX 6000 Ada). In contrast, producing the same dataset with physics-based simulation required approximately 4,000 NVIDIA A100 GPU-hours and 6,200 AMD EPYC 7763 CPU-hours on a national supercomputing system (Section 5.4), representing to a $> 10^4\times$ reduction in compute. Because GMFlow models a continuous function, it also supports zero-shot super-resolution at inference time: without retraining, it can synthesize wavefields on a finer $[3, 512, 256, 96]$ grid (~ 37.7 million scalar values) in seconds on a single GPU; representative results are provided in the Appendix D.

Compared with prior generative models for single-site waveform synthesis, GMFlow addresses a substantially more challenging field-level generation problem. Methods such as cGM-GANO (Shi et al., 2024b) and recent diffusion-based approaches (Palgunadi et al., 2024) generate three-component time histories at isolated sites but do not model within-event spatial correlations. Measured by total output dimensionality, the spatiotemporal wavefields generated by GMFlow are approximately $500\times$ larger than those of cGM-GANO and $780\times$ larger than the diffusion baseline, while being trained on a substantially smaller dataset than typically used for one-dimensional waveform models. In our framework, each earthquake scenario constitutes a single training sample representing a full regional wavefield, whereas in single-site waveform synthesis models each three-component ground motion at an individual station is treated as a separate sample.

A more closely related comparison is with WaveCastNet (Lyu et al., 2025), which also uses Bay Area physics-based simulations, but the problem setting is fundamentally different. WaveCastNet is a deterministic sequence to sequence forecasting model for earthquake early warning: it predicts future wavefields from an observed history window and introduces uncertainty only through expensive model ensembling. In contrast, GMFlow is a stochastic functional generative model for regional ground-motion synthesis conditioned only on low-dimensional physical parameters. Moreover, WaveCastNet is trained on small point-source earthquakes in a lower-frequency regime (< 0.5 Hz), whereas GMFlow trains a unified generator

across both point-source and finite-rupture events up to 1 Hz. Thus, WaveCastNet forecasts a realized event under strong observational constraints, whereas GMFlow must synthesize full event-level regional wavefields from physical conditions while matching spatial coherence, frequency content, and magnitude scaling.

Difference from Video Generative Models

Although both video generative models and GMFlow produce spatiotemporal outputs, the underlying problem setting is fundamentally different. Video models typically learn distributions over discrete frame tensors on fixed pixel grids and generate short clips conditioned on text, images, or preceding frames (Ho et al., 2022; He et al., 2022; Wan et al., 2025). In contrast, GMFlow models an event-level seismic wavefield as a random spatiotemporal function conditioned on physical parameters. This functional formulation supports resolution-flexible querying and is designed for scientific field synthesis rather than perceptual content generation.

The target distributions and constraints are also different. Video generation addresses broad, open-ended visual scenes, where realism is usually judged perceptually and strict physical consistency is not generally enforced. By contrast, GMFlow targets a specific physics-governed process (i.e., regional seismic wave propagation) that implicitly encodes a physical process. It incorporates this physical structure through a staged factorization and a Physics-Aligned Subspace that leads generation toward propagation-consistent solutions.

The data and evaluation regimes also differ accordingly. State-of-the-art video models are typically trained on internet-scale datasets with very large compute budgets, whereas GMFlow is designed for scientific-scale learning under limited data and compute while still generating outputs with millions of spatiotemporal degrees of freedom on a single workstation GPU. Furthermore, video generation is often assessed using perceptual realism and temporal consistency, while GMFlow is evaluated using physics- and engineering-driven criteria, including spectral fidelity, intensity measures, residual bias across frequency, and process-specific correlation structures.

Future Directions: Irregular Geometry and Data Fusion

A natural next step is to replace FFT-based FNO backbones with operator architectures that natively support irregular geometries, such as Mesh-Informed Neural Operators (MINO) (Shi et al., 2025). Unlike FFT-based operators, these architectures decouple evaluation locations from a fixed latent grid and can there-

fore support efficient inference on irregular station layouts, unstructured domains, and multi-resolution meshes. Extending GMFlow in this direction would improve compatibility with observational networks and move the framework closer to truly geometry-flexible functional generation.

A second priority is to develop hybrid training datasets and generative models that combine physics-based simulations with real observations. Neither source alone is sufficient for earthquake-engineering applications. Observational data provide the most reliable record of real ground motions, especially at high frequencies, but large-magnitude events relevant to infrastructure resilience are rare and station coverage is often too sparse to fully resolve the regional spatiotemporal structure of shaking. Physics-based simulations, by contrast, provide dense spatial coverage and controlled variation over earthquake scenarios, source properties, and receiver locations. However, they are often limited in frequency bandwidth because accurate propagation of high-frequency energy requires both substantial computational cost and highly detailed Earth models.

These complementary strengths suggest a hybrid generative framework that combines the spatial completeness and scenario control of simulations with the spectral realism of observations. The main challenge is that the two data sources differ not only in fidelity, but also in geometry, bandwidth, and error characteristics. One possible strategy is to use the Physics-Aligned Subspace as a shared low-frequency representation, where both simulations and observations can be compared or assimilated at scales that are more reliably resolved. Observations could then be used primarily to calibrate the high-frequency lifting (with super-resolution operator), improving the spectral realism of the generated fields while preserving the large-scale coherence learned from simulations. When combined with irregular-geometry operator backbones such as MINO, such a framework could support mixed-resolution and mixed-geometry training, and may provide a path toward regional generative models that retain simulation-scale coherence while better matching the high-frequency content of recorded ground motions. A systematic investigation of these directions is left to future work.

Broader Impact: Toward Physics–AI Generative Hazard Modeling

Beyond seismology, GMFlow points to a broader framework for generative modeling in natural-hazard science. Hurricanes, wildfires, tsunamis, and volcanic plumes differ in their governing physics, but they share a common modeling challenge:

spatiotemporal dynamics on complex domains under uncertain forcing and incomplete observations (Schwerdtner et al., 2024; Yu et al., 2025; Price et al., 2025). In this context, GMFlow shows how conditional solution operators can be learned directly in function space, complementing expensive grid-based simulation with fast, queryable synthesis of coherent spatiotemporal fields with uncertainty.

Accordingly, GMFlow can be viewed not only as a model for regional ground-motion synthesis, but also as a potential foundational framework for generative modeling of diverse large-scale spatiotemporal natural-hazard phenomena. By addressing the computational bottleneck of uncertainty-aware regional field generation, it could support large probabilistic scenario ensembles for downstream risk analysis of distributed infrastructure and emergency-planning workflows.

5.4 Methods

We first describe simulation dataset preparation, then formulate the problem of regional, event-level ground-motion synthesis in function space, followed by the design of the GMFlow architecture, training procedure, and inference workflow. Additional results, evaluation, and ablation are provided in the Appendix D.

Simulation Dataset Preparation

We selected the San Francisco Bay Area (SFBA) as our computational testbed and used the USGS v21.1 community velocity model to capture realistic heterogeneous geological structure (Aagaard et al., 2021). This model has been rigorously evaluated using high-resolution 3D physics-based simulations and recorded seismicity. We generate simulated events with the SW4 finite-difference code, which solves the elastodynamic wave equations with fourth-order accuracy (McCallen et al., 2021b; McCallen et al., 2024). The domain spans a 43×92 km region covering the Hayward Fault. Simulations use a minimum grid spacing of 31.25 m, a minimum shear-wave velocity of 250 m/s, absorbing boundaries, and are configured to resolve frequencies up to 1 Hz with at least eight points per wavelength throughout the mesh. Wavefields are stored on a sparse 250 m output grid. To simplify the learning setting and avoid boundary effects, we use a truncated 80×40 km subdomain with the same geometric center as the original computational domain, apply a sixth-order low-pass filter at 1 Hz, resample to [256, 128] spatial points, and truncate the duration to 24 seconds with a sampling frequency of 4 Hz, yielding three-component velocity wavefields of shape [3, 256, 128, 96].

The simulation dataset covers both small and large events with different source

regimes. For small-magnitude earthquakes, we simulated 2,100 point-source events (M_w 4.4) using a double-couple source with a Liu source-time function to model asymmetric energy (Liu et al., 2006). Hypocenter locations were sampled via Latin Hypercube Sampling to ensure adequate spatial coverage. For large-magnitude events, we targeted the Hayward Fault and generated 2,100 M_w 6.0 scenarios and 1,100 M_w 7.0 scenarios. These ruptures use the Graves–Pitarka kinematic model with stochastic slip distributions and rupture velocities to introduce realistic source heterogeneity. To improve computational efficiency for large-rupture events, we adopt a linear superposition strategy in which a finite-fault rupture is decomposed into a superposition of smaller rupture segments with appropriate time delays (Additional details are provided in Appendix D). The final dataset contains 5,000 training events (2,000 M_w 4.4, 2,000 M_w 6.0, and 1,000 M_w 7.0) and 300 test events (100 per magnitude). Simulations were performed on Perlmutter at the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory, using a total of 4,000 NVIDIA A100 GPU-hours and 6,200 AMD EPYC 7763 (Milan) CPU-hours.

Function-space formulation of conditional event-level ground-motion synthesis

Let $D = \Omega \times [0, T]$ denote the spatiotemporal domain of wavefields where $\Omega \subset \mathbb{R}^2$ represents the spatial domain and $[0, T]$ the time domain. We represent an event-level three-component ground-motion velocity wavefield as a function $u : D \rightarrow \mathbb{R}^3$. We assume that u belongs to a separable Hilbert space $\mathcal{H}$. The probability distribution of wavefields is defined on the measurable space $(\mathcal{H}, \mathcal{B}(\mathcal{H}))$, where $\mathcal{B}(\mathcal{H})$ denotes the Borel σ -algebra on $\mathcal{H}$. For an earthquake event described by conditioning variables c (magnitude and source location), drawn from a distribution $c \sim \pi_c$, we assume that the wavefield follows an unknown conditional probability law $u \mid c \sim \mu_c^\dagger$. The marginal distribution of wavefields is therefore the mixture $\mu(A) = \int \mu_c^\dagger(A) \pi_c(dc)$ for any measurable set $A \in \mathcal{B}(\mathcal{H})$. Our goal is to learn a conditional generative model $\mathcal{G}_\theta$ that approximates the family of conditional probability laws $\{\mu_c^\dagger\}$.

Let $z_0 \sim \pi_0$ denote a latent noise variable defined on the latent probability space $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$. Given conditioning variables c , we generate a synthetic wavefield through the mapping $\hat{u} = \mathcal{G}_\theta(c, z_0)$, where $\mathcal{G}_\theta$ is a generator function, parameterized by θ . This construction induces, for each c , a probability measure $\mu_{\theta,c}$ on $(\mathcal{H}, \mathcal{B}(\mathcal{H}))$ defined as the pushforward of the latent distribution through the generator: $\mu_{\theta,c} := (\mathcal{G}_\theta(c, \cdot))_\# \pi_0$. In other words, $\mu_{\theta,c}$ is the distribution of the random

wavefield $\hat{u} = \mathcal{G}_\theta(c, z_0)$ when $z_0 \sim \pi_0$.

We parameterize $\mathcal{G}_\theta$ using a neural operator and train it so that the model-induced conditional distributions $\mu_{\theta,c}$ approximate the true conditional distributions $\mu_c^\dagger$ for conditioning variables c drawn from the data distribution.

Adopting a function-space viewpoint for event-level ground-motion synthesis offers several advantages over standard fixed-grid generative modeling:

- **Domain and Discretization Agnosticism.** The functional framework mitigates the "*Curse of Dimensionality*" inherent to fixed-grid generative modeling. For regional wavefields observed on millions of spatiotemporal grid points, directly modeling a distribution on a single fixed discretization can be computationally and statistically burdensome. A function-space formulation provides a more flexible representation by separating the underlying continuous field from any particular discretization, where the discretization only introduces approximation errors. Further, this approach is also theoretically domain-agnostic, which supports domains with irregular boundaries that better aligns with the real-world earthquake scenario (Shi et al., 2025).
- **Flexibility in Training and Inference.** In functional generative models, each training wavefield sample can be observed on its own unique set of spatial locations, ranging from dense regular simulation grids to sparsely sampled station networks from real earthquake, where the traditional fixed-grid generative theory falls short. During inference, the trained model can generate a continuous wavefield that can be evaluated at arbitrary query locations, enabling zero-shot generation on much finer discretizations not seen in the training dataset.

Despite these conceptual advantages, the function-space perspective alone does not resolve the practical computational challenges of large-scale generative modeling. Efficient implementation still requires careful design of the model architecture and training pipeline, which we describe in the following sections.

Learning method and architecture

GMFlow consists of three components organized in a two-stage synthesis pipeline. We elaborate each components in the following part.

Physics-Aligned Subspace and super-resolution operator. Let A_f denote a temporal low-pass filtering operator with cutoff frequency f_c . We define the *Physics-Aligned Subspace* as

$$\mathcal{H}_f := \text{Range}(A_f) \subseteq \mathcal{H}, \quad u_f := A_f u \in \mathcal{H}_f.$$

We refer to $\mathcal{H}_f$ as a *Physics-Aligned Subspace* because it isolates the low-frequency component of the wavefield, which is typically the part most robustly determined by large-scale wave propagation physics and most reliably resolved in regional simulations. The higher-frequency content is more sensitive to the heterogeneity, scattering, source complexity, and discretization effects. Thus, the role of A_f is not to impose a new physical constraint, but to define an intermediate representation aligned with the physically better-resolved scales of the problem, while deferring reconstruction of the remaining frequency content to a separate lifting stage. The super-resolution task is then to map the low-pass wavefield u_f to a full-band reconstruction $\hat{u} \approx u$. We parameterize a super-resolution neural operator (SNO),

$$S_\phi : \mathcal{H}_f \rightarrow \mathcal{H}, \quad \hat{u} := S_\phi(u_f),$$

and optimize its parameters using paired samples $\{(u_f^{(i)}, u^{(i)})\}_{i=1}^N$ by minimizing the L^2 reconstruction loss,

$$\mathcal{L}_{\text{SNO}} = \mathbb{E}_{u \sim \mu} \left[\|S_\phi(u_f) - u\|_{L^2(\mathcal{D})}^2 \right] \approx \frac{1}{N} \sum_{i=1}^N \|S_\phi(u_f^{(i)}) - u^{(i)}\|_{L^2(D)}^2,$$

where $\|\cdot\|_{L^2(D)}$ is implemented by standard discrete quadrature on the chosen output discretization, i.e., mean-squared error over spatiotemporal grid points, and S_ϕ can be interpreted as the super-resolution operator, with the minimum mean-square-error estimator. Although A_f is non-injective and therefore does not admit a unique inverse in general, the mapping $u_f \mapsto u$ can still be highly predictable on the support of physically realizable wavefields. In our fixed-region setting with a prescribed medium (the SFBA velocity model), the dominant variability is primarily source-driven: the low-frequency response retains large-scale source information, such as seismic moment and rupture directivity, which helps constrain the conditional statistics of the higher-frequency content.

In our experiments, we use $N = 5,000$ paired training samples and set the cutoff frequency to $f_c = 0.75$ Hz. For computational efficiency, we evaluate u_f on a coarser spatiotemporal discretization. Let $(\mathbf{x}, \mathbf{t})$ and $(\mathbf{x}, \mathbf{t})_f$ denote the uniform

grid coordinates associated with the fine and coarse discretizations of the same domain D , with $(\mathbf{x}, \mathbf{t}) \in \mathbb{R}^M$ and $(\mathbf{x}, \mathbf{t})_f \in \mathbb{R}^{M_f}$, where $M = 256 \times 128 \times 96$ and $M_f = 128 \times 64 \times 48$. Accordingly, each sample is represented as $u^{(i)}(\mathbf{x}, \mathbf{t}) \in \mathbb{R}^{3 \times M}$ and $u_f^{(i)}((\mathbf{x}, \mathbf{t})_f) \in \mathbb{R}^{3 \times M_f}$. We parameterize S_ϕ with a residual Fourier Neural Operator (rFNO), which adds inter-block residual connections to a standard FNO backbone. To reduce dynamic-range issues across event magnitudes, we normalize both u and u_f by their sample-wise standard deviation and append $\log_{10}$ of this scalar as an additional constant channel. The resulting inputs are $u^{(i)}(x) \in \mathbb{R}^{4 \times M}$ and $u_f^{(i)}(x_f) \in \mathbb{R}^{4 \times M_f}$. This normalization is important because waveform amplitudes can vary substantially with magnitude; without rescaling, small events may be taken as noise or neglected during training relative to large events, degrading performance across the magnitude range.

Autoencoding Neural Operator. Inspired by latent diffusion frameworks (Rombach et al., 2022) and U-shape neural operator (UNO) (Rahman et al., 2023), we introduce an Autoencoding Neural Operator (AENO) to learn a compact, fixed-size latent representation of the low-frequency wavefield u_f . Conceptually, AENO maps the low-frequency wavefield to a low-dimensional latent tensor, reducing the computational cost of the downstream generative model by allowing flow matching to operate in a compressed latent space.

Formally, the AENO consists of an encoder operator $\mathcal{E}_\psi$ and a decoder operator $\mathcal{D}_\psi$, parameterized by ψ . Given an input $u_f \sim \mu_f$, where $\mu_f = (A_f)_\# \mu$ is the pushforward of the original measure μ under the low-pass filter A_f , the encoder maps u_f to a latent variable $z_1 = \mathcal{E}_\psi(u_f) \in \mathbb{R}^{1 \times 32 \times 16 \times 16}$. Thus, the latent representation has a single channel and fixed spatiotemporal dimensions. In principle, because $\mathcal{E}_\psi$ is formulated as an operator, it can be applied across varying input discretizations while mapping them into the same latent space $\mathcal{Z}$. The decoder then reconstructs the low-frequency wavefield from the latent representation $\hat{u}_f = \mathcal{D}_\psi(z_1)$.

We optimize ψ using an L^2 reconstruction objective over the data distribution:

$$\mathcal{L}_{\text{AENO}} = \mathbb{E}_{u_f \sim \mu_f} \left[\left\| \mathcal{D}_\psi(\mathcal{E}_\psi(u_f)) - u_f \right\|_{L^2(D)}^2 \right] \approx \frac{1}{N} \sum_{i=1}^N \left\| \mathcal{D}_\psi(\mathcal{E}_\psi(u_f^{(i)})) - u_f^{(i)} \right\|_{L^2(D)}^2.$$

As in the SNO objective, the $L^2(D)$ norm is implemented discretely on the chosen training grid. Although the AENO framework can in principle support variable input discretizations, in our experiments we evaluate u_f on a fixed coarse grid for computational efficiency. We instantiate both $\mathcal{E}_\psi$ and $\mathcal{D}_\psi$ with the UNO architecture

using rFNO blocks, for consistency with the SNO implementation, and additionally apply Instance Normalization (Ulyanov et al., 2016) after each block to stabilize latent activations.

Conditional flow matching with clean latent prediction. Let $z_1 \sim \pi_1$ denote a latent data sample obtained by encoding a physics-aligned wavefield u_f via $z_1 = \mathcal{E}_\psi(u_f)$, where $z_1 \in \mathbb{R}^{32 \times 16 \times 16}$ in our implementation. We sample the base noise as $z_0 \sim \pi_0 := \mathcal{N}(0, I)$. Rectified flow (Liu et al., 2022) (equivalently, independent-coupling flow matching (Tong et al., 2024)) defines a linear coupling path between z_0 and z_1 , $z_t := (1 - t)z_0 + tz_1$, where $t \in [0, 1]$, with constant marginal target velocity along the path,

$$v_t(z_t) := \frac{d}{dt}z_t = z_1 - z_0.$$

We parameterize a conditional velocity field v_θ and train it with the conditional rectified-flow objective

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{c \sim \pi_c, z_1 \sim \pi_1(\cdot|c), z_0 \sim \pi_0, t \sim \mathcal{U}[0,1]} \left[\|v_\theta(z_t, t, c) - (z_1 - z_0)\|_2^2 \right].$$

Following recent practice, instead of predicting v_θ directly, we adopt *clean-prediction with velocity-loss*: under a manifold assumption, clean samples lie on a low-dimensional data manifold where directly predicting the clean samples and then recalculate the velocity yields better performance (Li et al., 2025). Specifically, we parameterize a neural network $\mathcal{F}_\theta$ and set

$$\hat{z}_1 = \mathcal{F}_\theta(z_t, t, c), \quad v_\theta(z_t, t, c) = \frac{\hat{z}_1 - z_t}{1 - t},$$

for $t \in [0, 1]$ (denominator is clipped around $t=1$). At inference time, given conditioning variables c , we sample $z_0 \sim \pi_0$ and integrate the conditional ODE

$$\frac{dz_t}{dt} = v_\theta(z_t, t, c), \quad z_{t=0} = z_0,$$

to obtain $z_1 = z_{t=1}$. In practice, we solve this ODE numerically and use a 50-step Euler solver by default.

End-to-end inference. Given conditioning variables c , we first sample a latent realization $\hat{z}_1$ by integrating the conditional rectified-flow ODE. We then decode $\hat{z}_1$ to obtain a low-frequency wavefield $\hat{u}_f = \mathcal{D}_\psi(\hat{z}_1)$. Because the Physics-Aligned Subspace is defined by an explicit cutoff frequency ($f_c = 0.75$ Hz), we can optionally

enforce this band limit at inference time to suppress spurious energy above f_c in the intermediate stage, for example by applying an additional low-pass projection. In our experiments, this extra step is not required, as the generated $\hat{u}_f$ already satisfies the prescribed constraint to a good level. Finally, we recover a broader-band wavefield, resolved up to 1 Hz and sampled at 4 Hz, using the super-resolution neural operator, $\hat{u} := S_\phi(\hat{u}_f)$. Overall, the inference pipeline is

$$z_0 \sim \pi_0 \xrightarrow[\text{rectified flow}]{v_\theta(\cdot, \cdot, c)} \hat{z}_1 \xrightarrow[\text{decode}]{\mathcal{D}_\psi} \hat{u}_f \xrightarrow[\text{super-resolve}]{S_\phi} \hat{u}.$$

Metrics for Evaluation

We evaluate GMFlow using standard engineering metrics that quantify both point-wise intensity and spatiotemporal structure. For a three-component wavefield $u(x, t)$, we denote the two horizontal components by u_{h_1}, u_{h_2} and the vertical component by u_v . The instantaneous 3C amplitude is

$$\|u(x, t)\|_2 = \sqrt{u_{h_1}(x, t)^2 + u_{h_2}(x, t)^2 + u_v(x, t)^2}, \quad \forall (x, t) \in D. \quad (5.1)$$

Peak ground velocity (PGV). We define PGV at location x as the maximum 3C amplitude over time,

$$\text{PGV}(x) = \max_{t \in [0, T]} \|u(x, t)\|_2, \quad \forall x \in \Omega. \quad (5.2)$$

Fourier amplitude spectrum (FAS). Let $A_{h_1}(x, f)$ and $A_{h_2}(x, f)$ denote the Fourier amplitude spectra of the horizontal traces $u_{h_1}(x, [0, T])$ and $u_{h_2}(x, [0, T])$ at frequency f . We report the power-mean horizontal FAS,

$$A_h(x, f) = \sqrt{\frac{1}{2} (A_{h_1}(x, f)^2 + A_{h_2}(x, f)^2)}. \quad (5.3)$$

Peak normalized cross-correlation (NCC). Let x_0 be a reference location with reference traces $r_{h_1}(t) = u_{h_1}(x_0, t)$, $r_{h_2}(t) = u_{h_2}(x_0, t)$, and $r_v(t) = u_v(x_0, t)$. For each location x and lag $k \in \{-K, \dots, K\}$, define the valid overlap index set

$$\mathcal{T}_k = \begin{cases} \{0, \dots, T-1-k\}, & k \geq 0, \\ \{-k, \dots, T-1\}, & k < 0, \end{cases}$$

and the 3C normalized cross-correlation

$$\begin{aligned}
\rho(x, k) &= \frac{\sum_{t \in \mathcal{T}_k} [u_{h_1}(x, t+k)r_{h_1}(t) + u_{h_2}(x, t+k)r_{h_2}(t) + u_v(x, t+k)r_v(t)]}{\sqrt{A_k(x) B_k}}, \\
A_k(x) &= \sum_{t \in \mathcal{T}_k} [u_{h_1}^2(x, t+k) + u_{h_2}^2(x, t+k) + u_v^2(x, t+k)], \\
B_k &= \sum_{t \in \mathcal{T}_k} [r_{h_1}^2(t) + r_{h_2}^2(t) + r_v^2(t)].
\end{aligned} \tag{5.4}$$

The peak NCC coefficient $\rho^*(x)$ and corresponding lag $\tau^*(x)$ are

$$\rho^*(x) = \max_{k \in [-K, K]} \rho(x, k), \quad k^*(x) = \arg \max_{k \in [-K, K]} \rho(x, k), \quad \tau^*(x) = k^*(x) \Delta t, \tag{5.5}$$

where $\Delta t = 0.25$ s is the sampling interval and

$$K = \min \left(\left\lfloor \frac{\text{max_lag_s}}{\Delta t} + \frac{1}{2} \right\rfloor, T - 1 \right).$$

In this work, we choose $\text{max_lag_s} = 6$ s.

Interpolated conditions for magnitude scaling. Since GMFlow requires conditions as input for generating synthetic events, thus for the magnitude scaling section 5.2. Interpolation conditions are built for target magnitudes from 4.6 to 6.9 (step 0.1), with 100 samples per target magnitude. For each target magnitude m , the code linearly mixes two neighboring source pools: m4.4–m6 when $m \leq 6.0$, and m6–m7 when $m > 6.0$. The mixing weight is computed as

$$\alpha = \frac{m - 4.4}{6.0 - 4.4} \quad (m \leq 6.0), \quad \alpha = \frac{m - 6.0}{7.0 - 6.0} \quad (m > 6.0).$$

Then it randomly samples the required number from each pool (without replacement), copies the hypocenter from those condition vectors.

Spectral residuals. To quantify frequency-dependent bias, for each test event we generate 100 synthetic realizations under the same conditioning variables (e.g., magnitude and hypocenter). Let $A_h^{\text{data}}(x, f)$ and $A_h^{\text{syn}}(x, f)$ denote the horizontal FAS of the observed and synthetic wavefields, respectively. We compute the ensemble mean in log space and define the residual

$$\epsilon(x, f) = \ln(A_h^{\text{data}}(x, f)) - \text{mean}(\ln(A_h^{\text{syn}}(x, f))) \tag{5.6}$$

Chapter 6

CONCLUSIONS AND FUTURE WORK

This thesis developed generative modeling methods for earthquake ground-motion synthesis, progressing from single-station waveform generation to functional generative modeling and regional coherent wavefield synthesis. A central theme is that ground motions can be modeled not only as scalar intensity measures or fixed arrays, but as samples from conditional probability measures over functions. This viewpoint enables probabilistic generation of time histories and wavefields while retaining the structure needed for seismic hazard and regional risk applications.

6.1 Summary of Previous Chapters

Chapter 2 introduced a conditional generative adversarial neural operator for broadband three-component ground-motion synthesis at individual sites. The model generates acceleration time histories conditioned on earthquake and site parameters, including magnitude, rupture distance, site condition, and faulting style. The results show that the developed method can reproduce important magnitude, distance, and site-condition scaling trends, while also highlighting the limitations of independent site-level synthesis under sparse data, near-fault conditions, soft-soil sites, and high-frequency variability.

Chapter 3 developed operator flow matching for stochastic-process learning. This work generalized generative modeling from fixed-dimensional arrays to functions by learning non-Gaussian stochastic-process priors through function-space transport. The framework enables sampling, likelihood evaluation on queried function values, and posterior inference at arbitrary collections of points, providing a foundation for functional regression and inverse problems.

Chapter 4 introduced the Mesh-Informed Neural Operator, a domain- and discretization-flexible backbone for functional generative modeling. By combining graph neural operators with cross-attention, MINO extends functional generation beyond regular grids to irregular domains, meshes, and sensor layouts. This chapter also highlighted the importance of general evaluation metrics, such as Sliced Wasserstein Distance and Maximum Mean Discrepancy, for comparing functional generative models.

Chapter 5 presented Ground-Motion Flow, a physics-inspired latent operator flow-

matching framework for regional earthquake wavefield synthesis. The model generates scenario-conditioned, three-component spatiotemporal wavefields by first sampling the dominant low-frequency structure in a physics-aligned subspace and then reconstructing the target spatiotemporal wavefield using a super-resolution neural operator. Evaluated on large-scale San Francisco Bay Area simulations, GMFlow reproduces key ground-motion features, including PGV, Fourier amplitude spectra, magnitude scaling, spatial variability, and spatiotemporal coherence, while achieving orders-of-magnitude acceleration relative to repeated direct numerical simulation.

Together, these chapters establish a unified view of earthquake ground-motion synthesis as conditional generative modeling over functions. In this view, single-station records and regional wavefields are not separate modeling problems, but different discretizations of stochastic physical fields. Neural operators provide the functional representation, while generative models provide the probabilistic sampling mechanism. This combination enables scenario-conditioned ensemble generation, uncertainty representation, and scalable synthesis from individual time histories to regional spatiotemporal wavefields.

6.2 Future Directions

A key future direction is data–simulation fusion. Physics-based simulations provide dense spatial coverage, scenario control, and physically coherent wavefields, while recorded data provide direct constraints on real source, path, attenuation, and site effects, especially at high frequencies. Hybrid generative models that combine both sources could preserve the spatial coherence of simulations while improving realism in frequency bands and site conditions where simulations remain uncertain.

A related direction is broadband regional synthesis. The regional framework developed in this thesis focuses on the dominant low-frequency structure of coherent wavefields, while reliable high-frequency generation remains challenging because it depends on fine-scale rupture complexity, scattering, attenuation, and local site response. Future work could combine physics-aligned low-frequency generation with data-informed high-frequency correction, multiscale neural operators, or conditional generative refinement models to produce broadband regional ground motions.

Another important direction is observation-conditioned posterior sampling. Given a small number of recorded waveforms from an event, the goal would be to infer plausible regional wavefields over the target domain that agree with the observations

and remain physically coherent. Such a framework would connect operator flow matching, functional regression, and regional ground-motion synthesis, with potential applications in rapid response, shaking interpolation, event-specific updating, and uncertainty-aware hazard assessment.

Finally, the methods developed in this thesis are not limited to earthquake ground motion. The combination of neural operators, function-space generative modeling and Bayesian methods may be useful in other scientific domains, such as fluid dynamics, climate modeling, wave propagation, and problems where uncertainty is naturally represented by distributions over functions or fields. In summary, this thesis shows that generative modeling can provide fast, probabilistic, and physically informed tools for earthquake ground-motion synthesis. By complementing empirical models and physics-based simulations, these methods enable efficient scenario-conditioned ensemble generation at both site-specific and regional scales. The resulting framework opens a path toward scalable, uncertainty-aware seismic hazard workflows for resilient infrastructure systems.

Appendix A

APPENDIX FOR CHAPTER 2

As an illustrative example, Fig A1 compares the time history and Fourier Amplitude Spectrum acceleration, velocity, and displacement of a M_w 6.9, $R_{rup} = 23\text{km}$ event recorded at station ISKH02 of the Kik-Net strong motion network with $V_{s30} = 721\text{m/s}$. In the figure, the acceleration time series are rich in high frequencies and attain constant Fourier spectral content at frequencies larger than $\sim 5\text{Hz}$. The opposite is true for displacement time series, their energy content decays for frequencies higher than 0.1Hz . Spanning a much broader frequency range of spectral amplitudes from ~ 0.1 to 10Hz , the velocity time series provided the optimal balance of input frequency content to train cGM-GANO in the broadband frequency range $0.1 \sim 30\text{Hz}$.

Fig A2 shows the fault location (indicated by the black line) and the corresponding station positions (depicted as red dots) used in the BBP simulation. The yellow star on the fault represents the hypocenter.

Fig A3 compares acceleration time history and Fourier amplitude spectrum of BBP and cGm-GANO generated ground motions for both strike-slip and reverse scenarios under $M_w = 7.0$, $R_{rup} = 10\text{ km}$. In this figure, cGm-GANO captures the envelope of the BBP waveforms in the time domain. In the frequency domain.

Fig A4 presents the inter-frequency correlation of the cGM-GANO synthetically generated ground motions based on the BBP dataset.

Fig A5 shows the residual results of Fig 10 using only the validation Kik-Net dataset; that is, the subset not included in the training of cGm-GANO. The negligible differences between Fig 10 and S5 indicate no overfitting on the training dataset.

Fig A6 and A7 compare the velocity and displacement time histories between data and synthetic ground motions from cGM-GANO for the same scenarios shown in Figure 8.

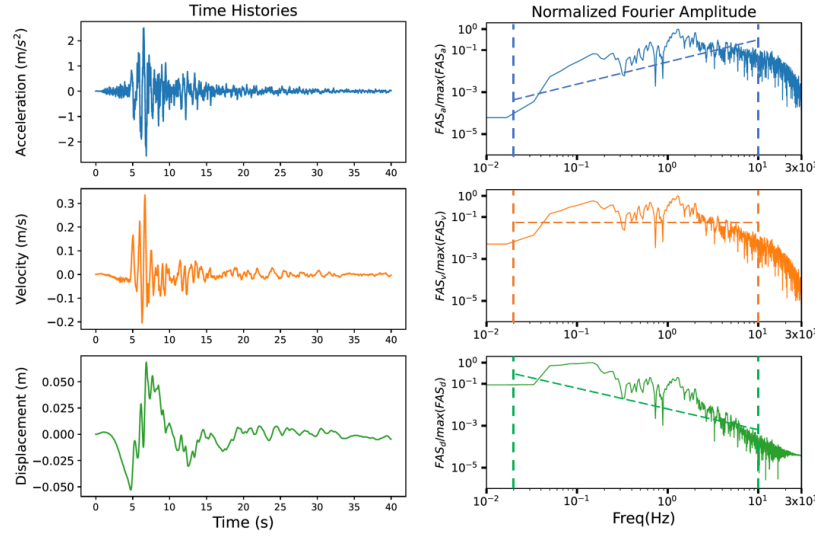


Figure A.1: Example of frequency content shift between acceleration, velocity, and displacement time series, depicted in the time and frequency domain. Example time series corresponds to event id ISKH020703250942 with $M_w = 6.9$ and $R_{rup} = 23\text{km}$ from the Kik-net dataset. The left-hand panels show the acceleration, velocity, and displacement time series from top to bottom, while the right-hand panels show the corresponding normalized frequency amplitude spectra. Each spectrum is normalized by its maximum amplitude. The dash lines show the FAS trend in 0.02Hz ~ 10Hz frequency range.

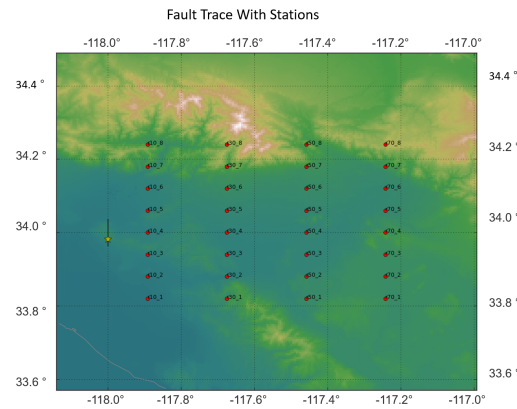


Figure A.2: An example of the location of the fault (black line) and stations (red dots) used in BBP simulation.

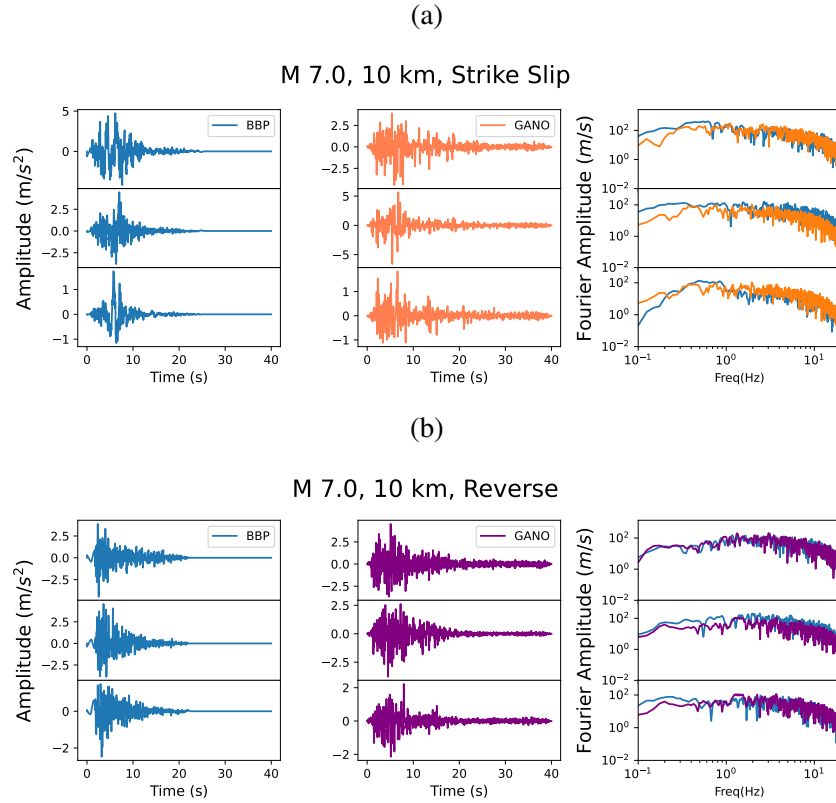


Figure A.3: Time-domain and frequency-domain waveforms comparisons of BBP and GANO generated ground motions. Subfigure (a) corresponds to a three-component waveform comparison for $M_w = 7$, $R_{rup} = 10km$, and strike-slip style of faulting. Subfigure (b) corresponds to a three-component waveform comparison for $M_w = 7$, $R_{rup} = 10km$, and reverse style of faulting.

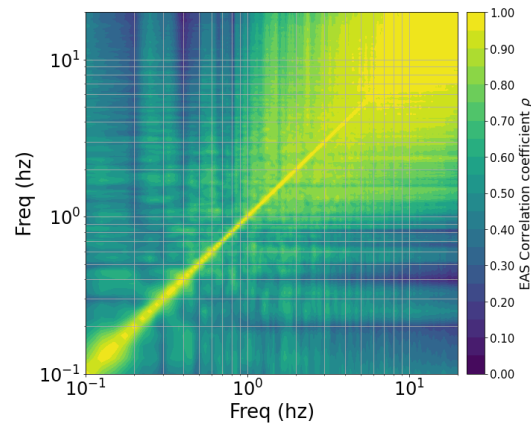


Figure A.4: Inter-frequency correlation of synthetically generated ground motions through cGM-GANO based on the BBP training dataset.

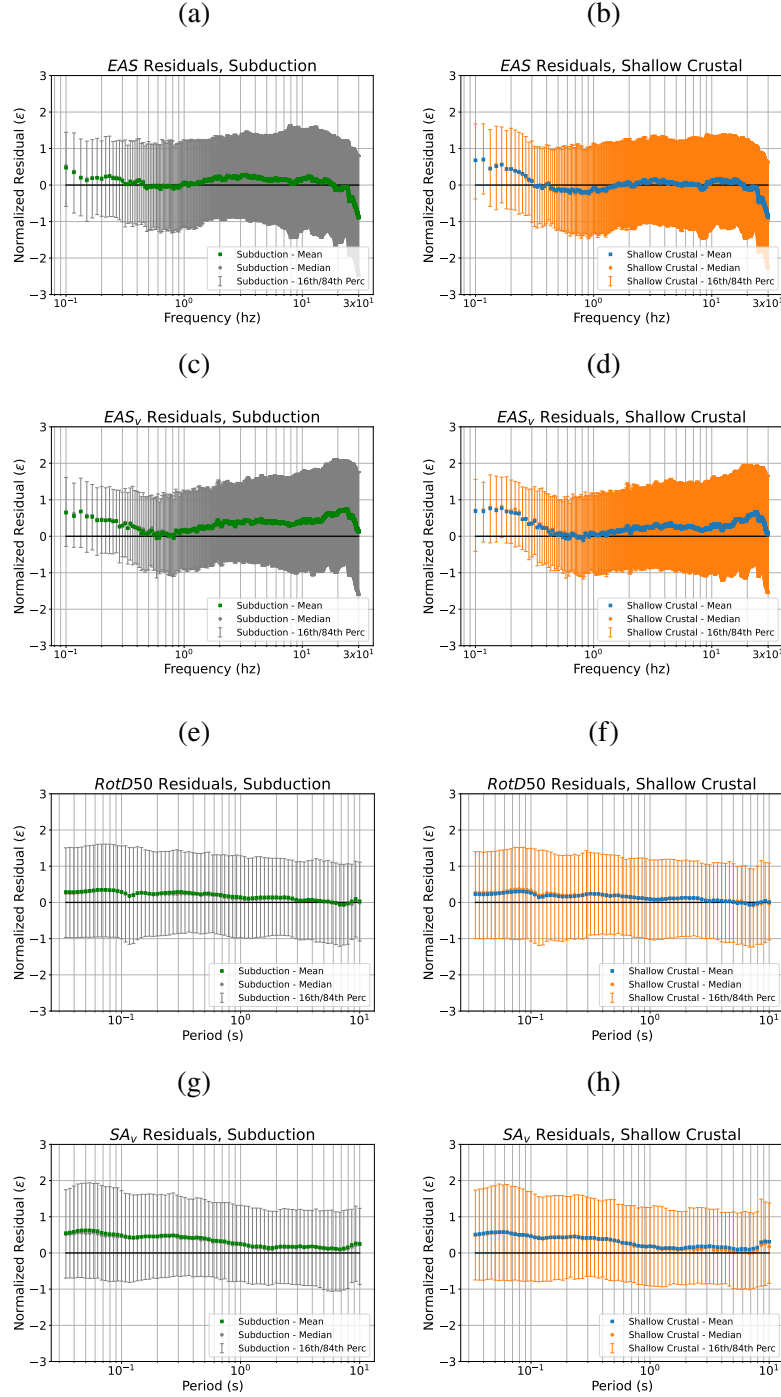


Figure A.5: Residual plots of the kik-net trained model based on the validation kik-net dataset (10 % of the total kik-net dataset) . Subfigures (a) and (b) show the horizontal EAS residuals for subduction and shallow crustal events, respectively. Subfigures (c) and (d) show the vertical EAS residuals for subduction and shallow crustal events, respectively. Subfigures (e) and (f) show the horizontal RotD50 PSA residuals for subduction and shallow crustal events, respectively. Subfigures (g) and (h) show the vertical PSA residuals for subduction and shallow crustal events, respectively.

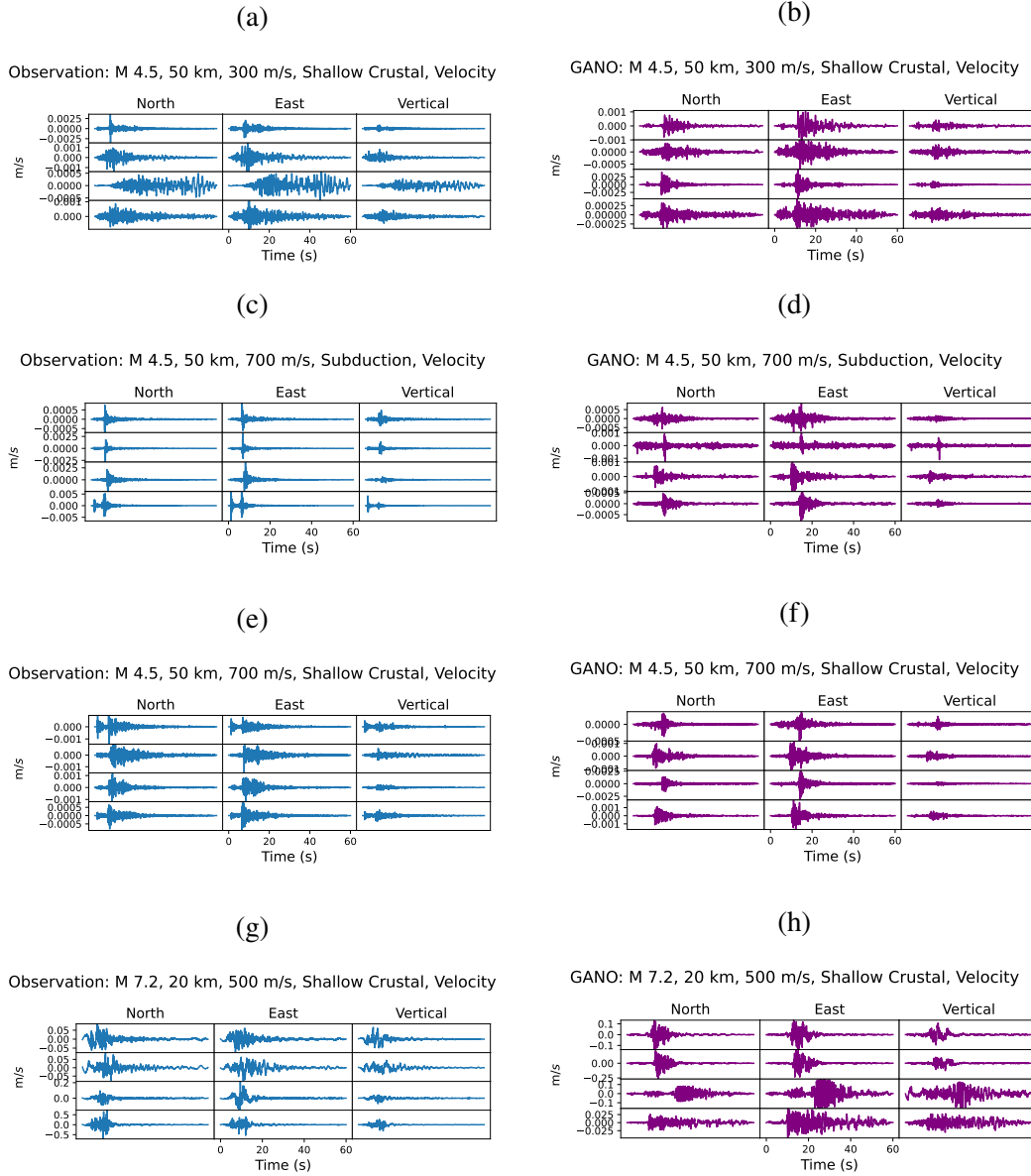


Figure A.6: Velocity waveforms comparisons between data and synthetic ground motions from cGM-GANO trained on the Kik-net dataset, unit in m/s .

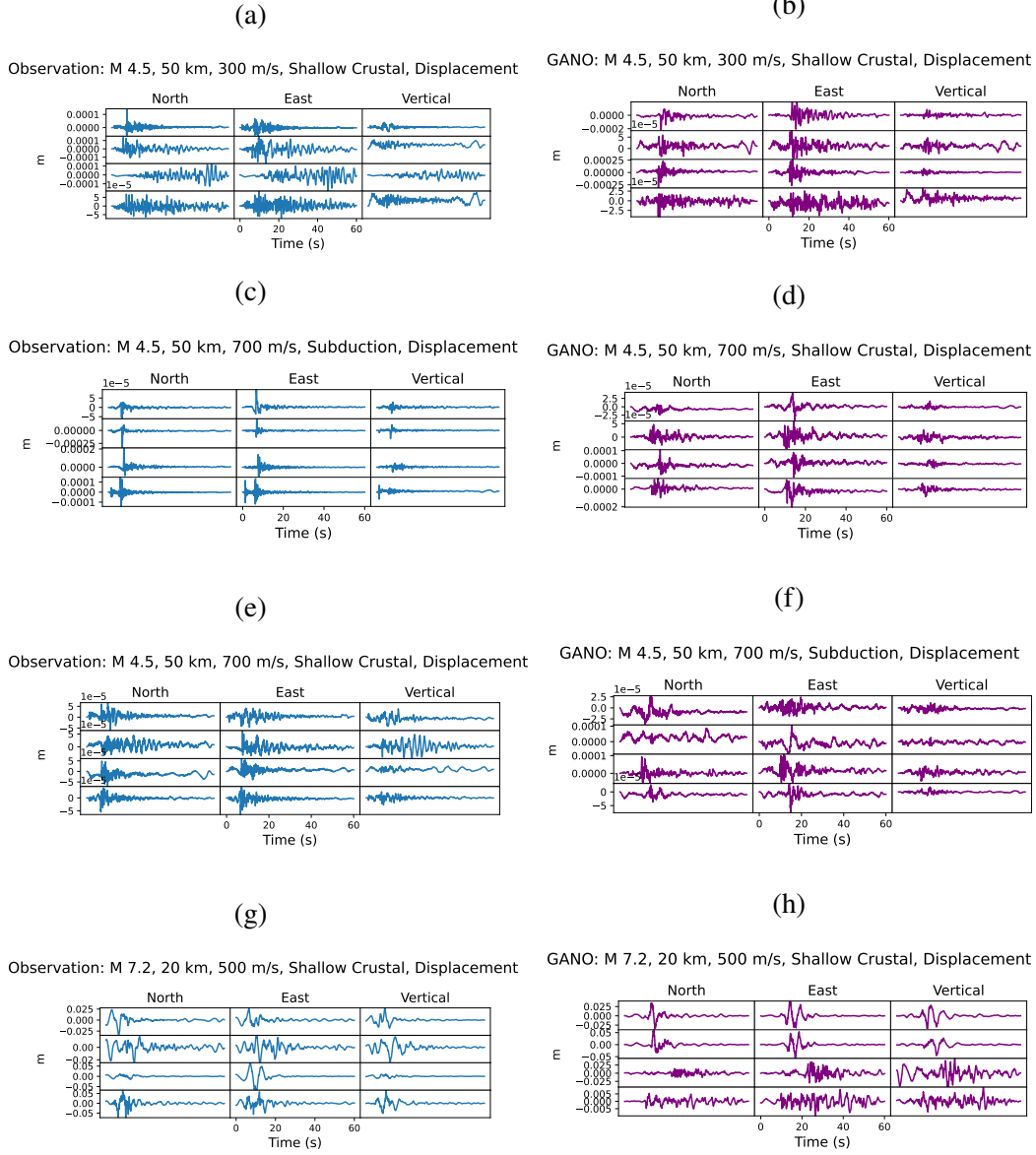


Figure A.7: Displacement waveform comparisons between data and synthetic ground motions from cGM-GANO trained on the Kik-net dataset, unit in m .

Appendix B

APPENDIX FOR CHAPTER 3

B.1 Potential questions and answers

Our method combines many vital fields, including

- **Operator learning**
- **Flow-based generative models (flow matching); Optimal transport in function space**
- **Bayesian uncertainty; Gradient Markov chain Monte Carlo, SGLD; Stochastic trace estimation**
- **Gaussian processes; Stochastic processes (Kolmogorov extension theorem)**

Because readers will have diverse backgrounds and expertise, we provide some potential questions and answers to enhance the clarity and highlight our contributions.

Q1: Why is it hard to obtain the true prior and posterior of an unknown stochastic process, and what would we gain if we could?

A1: Generalizing GP regression to arbitrary stochastic process regression has remained a long-standing challenge that has kept many researchers busy for decades. The core difficulties are twofold: first, learning a general stochastic-process prior from historical data with an expressive enough model, and second, deriving both the exact posterior distribution and an efficient sampling scheme.

Prior work falls into three main categories—GPs, deep GPs, and conditional models (NP family). Only the classical GP can fully characterize a stochastic process, and then only when the exact GP prior (mean function, covariance kernel, all hyper-parameters, and observation-noise variance) is already known. Even fitting one GP to data generated by another GP with unknown parameters rarely yields the true posterior. Deep GPs and NPs possess stronger representational power compared to GP, but **representational**

power $\neq$ exactness: their optimization rely on approximate posterior and **DO NOT** produce a closed-form “true posterior.”

If the true prior and posterior were available, we would gain near-complete control over general stochastic process, enabling perfect uncertainty quantification and Bayes-optimal decisions, which has numerous applications in finance market, science and engineering problems

Q2: How does OFM differ from standard finite-dimensional flow matching? What does it mean to generalize flow matching to stochastic processes, and how does extending a generative model to a function space differ from extending it to a full stochastic process?

A2: A standard flow matching model learns a transport map between two distributions defined on a fixed grid (e.g., a pixel lattice). Consequently, it can only generate samples at that specific resolution. In contrast, OFM learns a map between two stochastic processes in function space. This endows it with several properties that standard flow matching lacks:

1. **Resolution agnosticism.** OFM learns a transport map between two distributions defined on any given collection of points, without regard for the number of points, or their locations in the domain. This enables capabilities like zero-shot generation without retraining.
2. **Stochastic process consistency.** OFM respects the metric of the underlying space, ensuring that points close to each other in the input domain have appropriately correlated values in the output distribution. This is part of learning a valid stochastic process, which also satisfies crucial theoretical properties like the Kolmogorov extension theorem (i.e, consistency under marginalization)
3. **Convergence to a continuous function.** As the collection of query points becomes denser within the domain, the output of OFM converges to the underlying continuous function.
4. **Backwards compatibility.** When evaluated on a fixed set of points, OFM recovers the behavior of a conventional (finite-dimensional) flow matching up to a linear transformation.

Extending a generative model (e.g., flow matching) to an *infinite-dimensional function space* and extending it to a *full stochastic process* are related but distinct goals. The first concerns generating function values at any finite collection of points; Lebesgue measure and Kolmogorov consistency come into play, but the focus remains on finite dimensional marginals. The second addresses the distribution of the entire function itself, requiring a probability measure over an infinite-dimensional space.

Because of this distinction, lifting a model merely to a *function space* is strictly weaker. For example, GANO (Rahman et al., 2022) extends GANs to function space but not to a stochastic process, as it cannot describe the joint law of function values at arbitrary query sets. In OFM, we bridge these two viewpoints for flow matching with neural operators: by exploiting the discretization-convergence property of neural operators and the invertibility of flow matching, we extend the method to both function spaces and stochastic processes. See Appendix B.3 and B.4 for details.

Q3: These methods may appear similar to image inpainting or restoration with flow- or diffusion-based models; could you elaborate on the differences?

A3: Current inpainting with flow/diffusion models treats the task as an inverse problem in a finite-dimensional setting: one learns to reconstruct missing pixels on a fixed resolution grid, effectively solving a regression problem at a predetermined set of points. By contrast, OFM operates directly on *functions* and is *resolution-agnostic*. Given the same partial observations, our model can predict the entire function at any collection of query points, coarse or fine.

Moreover, OFM delivers principled uncertainty: its posterior quantifies the full distribution of possible completions, whereas finite-dimensional inpainting methods typically rely on an ensemble of visually plausible samples whose variability is not a calibrated measure of uncertainty. Finally, OFM remains effective even when observations are extremely sparse. (e.g 0.7% of total observations) —a regime in which grid-based inpainting approaches struggle.

Q4 : Could you elaborate further on the connections to related work? The approach appears to intersect with several studies, including OrFlow (Shi

et al., 2024a), COT-FM (Kerrigan et al., 2024), OT-CFM (Tong et al., 2024), and others.

A4: Because our work combines multiple fields, it naturally has connections with many other studies. Readers are referred to the appendix B.17 for a detailed discussion.

Q5: In Eq 3.11. What is the argument of replacing the suprema in Eq 3.8 with expectations? When is this valid?

A5: The supremum in Eq. 3.8 represents a worst-case error over the entire function space, which is computationally intractable to optimize directly. We therefore relax this hard constraint by replacing the supremum with an expectation, as formulated in Eq 3.11. This is a common empirical consideration.

Instead of minimizing the worst-case error, our objective becomes minimizing the tractable average-case error across the distribution. Minimizing the error on average provides a strong practical incentive for the model to perform well across the entire function space, thereby effectively reducing the worst-case error. Such replacement is always valid under the weaker goal. The validity of this approach as a tractable proxy is further confirmed by our empirical results, which show it successfully guides the model to learn the intended functional mapping (Appendix B.13).

Q6: Why not use log-likelihood as the evaluation metric for non-GP regression tasks when comparing to NP models? And why not include more recent conditional model baselines, such as NDP (Dutordoir et al., 2023)

A6: There are several reasons that likelihood as an evaluation metric is not relevant in our case. First, as shown in Eq B.23, there is an evidence term, which is a constant, in the posterior distribution in OFM framework. The constant evidence term is intractable but does not contribute to MAP estimation, mean estimation, and posterior sample in general.

Second, even if we can compute the evidence term, we still cannot make the comparison. The graphical model in the NP is such that the conditional

model is trained using the MLE principle and the learned likelihood model dependent. The posterior in OFM provides the posterior using the Bayes rule, utilizes a different model, and has a different graphical model. These quantities are not directly relevant to be compared.

Third, computing the true posterior for a general stochastic process is a known challenging problem. Due to the complexity nature of the problem, there doesn't exist a well-recognized metric. The evaluation of quality of posterior performance requires domain knowledge from experts and varies case by case, which is an exciting research direction.

For baseline models, We already include a broad set of SOTA baselines. On the operator-learning side, the latest OpFlow (Shi et al., 2024a) (2024) is covered. For deep GPs, we adopt Doubly Stochastic Variational Deep GPs (Salimbeni et al., 2017) and the Deep Sigma-Point Process (Jankowiak et al., 2020), both widely recognized and backed by publicly reproducible code.

Within the NP family, ConvCNP (Gordon et al., 2019) remains a standard benchmark, and we include two additional NP variants. Our baselines are limited to models with publicly reproducible implementations. Although we attempted to add the newer NDP (Dutordoir et al., 2023), we encountered significant reproducibility issues with the authors' code—an obstacle reported by others as well. Instead, we offer a detailed theoretical comparison between OFM and NDP in Appendix B.17.

Q7: What do you mean by an “exact posterior” and a practical solution? How can I apply the model to my own tasks, and what do I need to prepare?

A7: For functional regression, posterior error has two sources for any method: **(1) formulation error**—the gap between the model's theoretical posterior and the true one—and **(2) approximation error** from finite data, limited capacity, and optimization. Deep GPs rely on variational inference, optimizing an ELBO and thereby introducing a formulation gap: the posterior is only an *approximation*. NPs make even stronger simplifying assumptions, and doesn't provide the true posterior. All models, including OFM, incur approximation error, but the discussion above concerns only **formulation error**.

A *practical* solution means an expressive backbone that can learn a complex stochastic-process prior and a posterior-sampling routine whose runtime and

memory footprint remain acceptable for typical users; see Appendix B.15 for quantitative details.

Using the model is straightforward. OFM offers GP-style regression for non-GP tasks: given noisy observations, it returns the posterior at arbitrary query points. The key difference is that a GP prior is fixed by a hand-tuned kernel, whereas OFM learns the stochastic-process prior directly from data. For details on prior learning and SGLD-based posterior sampling, see Appendices B.13 and B.12.

B.2 Background: Flow Matching, Gaussian Measures on Function Spaces, and the Cameron–Martin Theorem

In this section, we provide essential background and high-level intuitive explanations of the topics involved in this paper for readers.

Flow Matching. Flow matching is a state-of-the-art generative paradigm that learns a time-dependent velocity field whose ODE transports samples from a simple reference (e.g., Gaussian) to the target data distribution, yielding an unbiased, simulation-free training objective that directly regresses ground-truth velocities along probability paths.

The approach is tightly linked to physics via the continuity equation, and—when cast as a continuous normalizing flow—provides a (deterministically) invertible transformation with tractable likelihoods, while also admitting stochastic variants when desired. In practice it matches diffusion-level quality with faster training and sampling (often few- or even single-step generation) and excellent scalability, which is why it has been adopted in challenging domains such as large-scale video generation, in-context image generation, and protein ensemble generation (Jin et al., 2025; Liu et al., 2025). We therefore use flow matching as the foundation for prior learning, leveraging its simple training objective, physics-grounded structure, and strong empirical performance.

Gaussian measures. A probability measure μ on a separable Hilbert space $\mathcal{H}$ is *Gaussian* if for any finite collection of vectors $\{h_1, \dots, h_n\} \subset \mathcal{H}$, the random vector $(\langle X, h_1 \rangle, \dots, \langle X, h_n \rangle)$ has a multivariate Gaussian distribution, where $X \sim \mu$. The family $\{\langle X, h \rangle : h \in \mathcal{H}\}$ is therefore a *Gaussian process* indexed by $\mathcal{H}$, with mean $h \mapsto \langle m, h \rangle$ for some $m \in \mathcal{H}$. Conversely, if a Gaussian process $\{X(t) : t \in T\}$ has sample paths that almost surely belong to a function space $\mathcal{H}$ (e.g., $C(T)$ or $L^2(T)$), then the law of the random path $t \mapsto X(t)$ is a *Gaussian measure* on $\mathcal{H}$. In short: a

Gaussian measure is the path-space law of a GP, and a GP is the collection of linear probes of a Gaussian measure.

Cameron–Martin space and theorem. Let μ be a Gaussian measure on a separable Hilbert space $\mathcal{H}$ with mean $m \in \mathcal{H}$ and covariance operator $C : \mathcal{H} \rightarrow \mathcal{H}$ (self-adjoint, positive, trace-class). The *Cameron–Martin space* (the RKHS associated with μ) is

$$\mathcal{H}_\mu = \overline{\text{Range}(C^{1/2})} \subseteq \mathcal{H},$$

equipped with the inner product $\langle u, v \rangle_{\mathcal{H}_\mu} := \langle C^{-1/2}u, C^{-1/2}v \rangle_{\mathcal{H}}$ on $\text{Range}(C^{1/2})$, extended by completion. With this choice, the inclusion $\mathcal{H}_\mu \hookrightarrow \mathcal{H}$ is continuous. For $h \in \mathcal{H}$, write $\mu_h(A) := \mu(A - h)$ for the translate. The *Cameron–Martin theorem* states:

(i) If $h \in \mathcal{H}_\mu$, then μ_h is absolutely continuous with respect to μ (in fact, μ_h and μ are equivalent), with

$$\frac{d\mu_h}{d\mu}(x) = \exp\left(\langle C^{-1/2}h, C^{-1/2}(x - m) \rangle_{\mathcal{H}} - \frac{1}{2} \|h\|_{\mathcal{H}_\mu}^2\right) \quad \text{for } \mu\text{-a.e. } x,$$

where the pairing is understood in the Paley–Wiener sense (which in finite dimensions reduces to $\langle h, C^{-1}(x - m) \rangle$).

(ii) If $h \notin \mathcal{H}_\mu$, then $\mu_h \perp \mu$ (mutually singular).

In finite dimensions this reduces to the classical mean-shift formula for $\mathcal{N}(m, \Sigma)$ with $C = \Sigma$, and $\|h\|_{\mathcal{H}_\mu}^2 = \langle h, \Sigma^{-1}h \rangle$. Thus, $\mathcal{H}_\mu$ pinpoints exactly the directions along which a Gaussian measure can be translated while remaining (mutually) absolutely continuous.

B.3 Stochastic process learning

Let $(\Omega, \mathcal{F}, P)$ denote a probability space and let $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ denote a measurable space where $\mathcal{B}(\mathbb{R})$ is the Borel space. Following the standard definition of stochastic processes (Brémaud (2020), Chapter 5.1), a stochastic process $\mathcal{P}$ on a domain D is a collection of $\mathbb{R}^d$ -valued random variables indexed by members of D , i.e.,

$$\{a(x) : x \in D\}$$

jointly following the probability law P . In the special case of Gaussian processes, e.g., Wiener process, following the Gaussian law for P , for any collection points $\{x_1, x_2, \dots, x_n\}$, the random variables $\{a(x_1), a(x_2), \dots, a(x_n)\}$ are jointly

Gaussian, resulting in a function a to be drawn from a GP. We need to emphasize, $\{a(x_1), a(x_2), \dots, a(x_n)\}$ is a collection of random variables (random vector) equipped with Lebesgue measure, and represents a discretized observation of one continuous function a . In practice, the joint probability distribution of the collection of the random variables is unknown a priori, and needs to be learned.

In SPL, one way we suggest is to learn an invertible operator $\mathcal{T}$ that maps a base stochastic process $\mathcal{P}$ to another stochastic process $\mathcal{Q}$ that represents the data via discretization convergence theorem (see Appendix B.4). That is, for any collection of points $\{x_1, x_2, \dots, x_n\}$, and for any n , the operator $\mathcal{T}$ maps the law on $\{a(x_1), a(x_2), \dots, a(x_n)\}$ to $\{u(x_1), u(x_2), \dots, u(x_n)\}$ and vice versa for the inverse $\mathcal{T}^{-1}$, where $u(x)$ is a pointwise evaluation of function data sample, i.e.,

$$\{u(x_1), u(x_2), \dots, u(x_n)\} = \mathcal{T}(\{a(x_1), a(x_2), \dots, a(x_n)\})$$

Then, the probability of $\{u(x_1), u(x_2), \dots, u(x_n)\}$, at evaluation points $\{x_1, x_2, \dots, x_n\}$, for any n and collection of points on D is given by,

$$\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\}) = \mathbf{JT} \Big|_{\{a(x_1), a(x_2), \dots, a(x_n)\}} \mathbb{P}(\{a(x_1), a(x_2), \dots, a(x_n)\})$$

where with abuse of notation $\mathbb{P}(u(x))$ denotes the density of $u(x)$ at point x , same for $\mathbb{P}(a(x))$, and similarly, following the notation in Theorem 11.1 of [villani_optimal_2009](#), $\mathbf{JT} \Big|_{\{a(x_1), a(x_2), \dots, a(x_n)\}}$ is the absolute value of the Jacobian determinant of the map from the random vector $\{a(x_1), a(x_2), \dots, a(x_n)\}$ at points $\{x_1, x_2, \dots, x_n\}$ to the random vector $\{u(x_1), u(x_2), \dots, u(x_n)\}$ via inverse operator $\mathcal{T}^{-1}$. We further show that the pushedforward $\mathcal{Q}$ is indeed a valid stochastic process via Kolmogorov Extension Theorem (KET) (Kolmogorov, [2018](#)) with a proof provided in Appendix B.4. In SPL, we aim to learn a neural operator $\mathcal{T}_\theta$ such that the resulting $\mathcal{Q}$ matches the data process under the true $\mathcal{T}$.

B.4 Model stochastic process with infinite-dimensional flow matching via Kolmogorov Extension Theorem

In operator learning, neural operators (Li et al., [2020b](#); Kovachki et al., [2023](#); Azizzadenesheli et al., [2024](#)) are typically designed to map an input function to an output function. When the input function is provided at a specific discretization (e.g., a set of points with their corresponding values), the model processes this discretized input as a collection of points and their values. Traditionally, in operator learning, this process is seen as an approximation of the operator's application to the

underlying continuous function, where the discretization introduces approximation errors. Thus, the input is conceptually still treated as a function.

Moreover, the application of the operator to a collection of points is well-defined, and, by the discretization convergence theorem, as the number of points increases, this operation converges to a well-defined mapping. In this paper, leveraging these properties, we adopt a different perspective as described in the introduction. We extend neural operators to define explicit maps between collections of points. In this framework, the input is not the abstract function itself but rather a collection of points and their associated values. Importantly, this mapping remains well-defined regardless of the number of points in the collection and, by the discretization convergence theorem, converges to a unique mapping as the point collection approaches the underlying continuous function.

Next, we show that given an invertible operator $\mathcal{T}$ and a valid stochastic process $\mathcal{P}$ whose finite dimensional marginal is $\mathbb{P}(\{a(x_1), a(x_2), \dots, a(x_n)\})$, there exist a valid stochastic process $\mathcal{Q}$ with finite-dimensional marginal $\{u(x_1), u(x_2), \dots, u(x_n)\}$.

Once again, as defined in Section B.3

$$\{u(x_1), u(x_2), \dots, u(x_n)\} = \mathcal{T}(\{a(x_1), a(x_2), \dots, a(x_n)\})$$

Then, the probability of $\{u(x_1), u(x_2), \dots, u(x_n)\}$, at evaluation points $\{x_1, x_2, \dots, x_n\}$, for any n and collection of points on D is given by,

$$\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\}) = \mathbf{JT} \Big|_{\{a(x_1), a(x_2), \dots, a(x_n)\}} \mathbb{P}(\{a(x_1), a(x_2), \dots, a(x_n)\}) \quad (\text{B.1})$$

where with abuse of notation $\mathbb{P}(u(x))$ denotes the density of $u(x)$ at point x , same for $\mathbb{P}(a(x))$, and similarly, following the notation in Theorem 11.1 of **villani_optimal_2009**,

$\mathbf{JT} \Big|_{\{a(x_1), a(x_2), \dots, a(x_n)\}}$ is the absolute value of the Jacobian determinant of the map from the random vector Jacobian of the map from the random vector $\{a(x_1), a(x_2), \dots, a(x_n)\}$ at points $\{x_1, x_2, \dots, x_n\}$ to the random vector $\{u(x_1), u(x_2), \dots, u(x_n)\}$ via inverse operator $\mathcal{T}^{-1}$. We should notice Eq. B.1 represents the changes of variables between two random vectors, with Lebesgue measure involved. The connection between the finite-dimensional marginal (equipped with Lebesgue measure) and the probability measure of a stochastic process in infinite-dimensional space is described by Kolmogorov Extension theorem (KET) (Kolmogorov, 2018), which assures that if all finite-dimensional distributions (i.e., distributions of function at finite collection of points) are consistent, then a stochastic process exists that matches finite-dimensional distributions.

Formally, according to KET, to establish that a valid stochastic process $\mathcal{Q}$, which has $\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\})$ as its finite dimensional distributions, it is essential to demonstrate that such a joint distribution satisfies the following two consistency properties:

Permutation invariance. For any permutation π of $\{1, \dots, n\}$, the joint distribution should remain invariant when elements of $\{x_1, \dots, x_n\}$ are permuted, such that

$$\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\}) = \mathbb{P}(\{u(x_{\pi(1)}), u(x_{\pi(2)}), \dots, u(x_{\pi(n)})\}) \quad (\text{B.2})$$

Marginal Consistency. This principle specifies that if a portion of the set is marginalized, the marginal distribution will still align with the distribution defined on the original set, such that for $m \geq n$

$$\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\}) = \int \mathbb{P}(\{u(x_1), \dots, u(x_m)\}) du(x_{n+1}) \cdots du(x_m) \quad (\text{B.3})$$

The permutation invariance property is naturally upheld when utilizing operator, as there is no inherent order among the elements in the set $\{x_1, x_2, \dots, x_n\}$. Furthermore, the marginal consistency property is also maintained due to the definition of operator $\mathcal{T}$ (see Eq. B.1), which ensures that $\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\})$ is closed under marginalization. This is because $\mathbb{P}(\{a(x_1), a(x_2), \dots, a(x_n)\})$ is closed under marginalization, which fully determines $\mathbb{P}(\{u(x_1), u(x_2), \dots, u(x_n)\})$ through the Jacobian. While verifying that $\mathcal{Q}$ constitutes a valid induced stochastic process is straightforward given the $\mathcal{T}$, approximating the $\mathcal{T}$ with a neural operator is non-trivial and depends highly on the model used (related to expressiveness). For instance, in Transforming GP (Maroñas et al., 2021), the authors employ a marginal normalizing flow, which acts as a point-wise operator to transform values from a GP to another. Consequently, the induced Jacobian is a diagonal matrix. More recently, OpFlow (Shi et al., 2024a) introduces an invertible neural operator by generalizing RealNVP to function space, which induces a triangular Jacobian matrix. In our work, we extend this framework to a more comprehensive case: a diffeomorphism. Here, the induced Jacobian is a full-rank matrix and is not necessarily triangular or diagonal, the determinant of the Jacobian for any collection of points is calculated through Eq 3.15.

Last, we want to clarify the connection between the notions of operator $\mathcal{T}$ and operator $\mathcal{G}$ throughout this paper. The operator $\mathcal{T}$ is the Φ_t (a diffeomorphism) defined in Eq. 3.5, which is the integral of $\mathcal{G}$ over time interval $[0, 1]$. Due to the

nature of an ODE system, $\mathcal{T}$ is invertible. However, $\mathcal{G}$ is not necessary invertible, which enables us to parameterize it with a classical neural operator, like FNO (Li et al., 2020b).

B.5 Universal Functional Regression

UFR is concerned with Bayesian regression on function spaces (Shi et al., 2024a), where it can be used to infer the posterior of an unknown function on a domain D from a collection of pointwise observations. The observations are often corrupted with noise of variance σ^2 , denoted as $\{\widehat{u}(x_1), \widehat{u}(x_2), \dots, \widehat{u}(x_n)\}$ or $\{\widehat{u}(x_i)\}_{i=1}^n$. More specifically, for $m \geq n$ points at which the function is to be inferred,

$$\mathbb{P} \left(\{u(x_1), u(x_2), \dots, u(x_m)\} \middle| \{\widehat{u}(x_1), \widehat{u}(x_2), \dots, \widehat{u}(x_n)\} \right)$$

Note that when the prior over the function space is Gaussian, UFR reduces to the celebrated GP regression. Following Bayes rule, and maps between stochastic processes, we obtain the log posterior as follows,

$$\begin{aligned} \log \mathbb{P} \left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n \right) &= -\frac{1}{2} \sum_i^n \frac{(\widehat{u}(x_i) - u(x_i))^2}{\sigma^2} - n \log(\sigma) - \frac{n}{2} \log(2\pi) \\ &\quad + \log \mathbb{P}(\{u(x_i)\}_{i=1}^m) - \log \mathbb{P}(\{\widehat{u}(x_i)\}_{i=1}^n) \end{aligned}$$

This equality holds for any collection of points. It is worth noting that the posterior is exact up to constants, i.e., the second, third, and last terms are constant. Therefore, they do not contribute in MAP estimation, mean estimation, and functional regression in general, and there is no need to compute them.

B.6 Marginal (dynamic) optimal-transport flow matching in function space via optimal coupling and dynamic Kantorovich formulation

Consider a joint probability measure $\pi(\nu_0, \nu_1)$ on $\mathcal{H} \times \mathcal{H}$, where the reference measure ν_0 , is chosen as a Gaussian measure, whose absolute continuity is well-studied (Bogachev, 1998). We characterize ν_0 by a GP with trace-class covariance operator. e.g. $\nu_0 = \mathcal{N}(m_0, C_0)$, where m_0 is the mean, C_0 is the covariance operator. With the joint measure $\pi(\nu_0, \nu_1)$, we sample a function pair $z := (h_0, h_1)$.

Assuming ν_1 has full support on the Cameron-Martin space associated with ν_0 (following the convention of literatures (Kerrigan et al., 2023; Lim et al., 2025)), we construct a conditional probability measure $\mu_t(\cdot|z)$ as a Gaussian measure with trace-class covariance operator and small operator norm to approximate Dirac measures

in the sense of weak convergence. Such that, at $t = 0$ and $t = 1$, $\mu_t(\cdot|z)$ is a centered around h_0, h_1 , approximating $\delta_{h_0}, \delta_{h_1}$ respectively; Subsequently, we can construct a new marginal probability measure by mixing these approximated Dirac measures:

$$\mu_t(A) = \int \mu_t(A|z) d\pi(z), \forall A \in \mathcal{B}(\mathcal{H}) \quad (\text{B.4})$$

Due to $d\pi(z)$ being always positive, the conditional probability measure (Dirac measure approximated by Gaussian measure) is absolutely continuous with respect to μ_t . Eq. B.4 indicates that $\mu_0 = \int \delta_{h_0} d\pi(z) \approx \nu_0$, and $\mu_1 = \int \delta_{h_1} d\pi(z) \approx \nu_1$. This formulation suggests that μ_0, μ_1 represent convolutions of ν_0, ν_1 with Gaussian measures. For a more detailed discussion on convolution with Gaussian measures, we refer the readers to Appendix B.1 of Lim et al. (2025).

Please note, dirac measure is not a necessary condition for Eq. B.4; the only constraint is the boundary conditions. One viable choice for the conditional measure is a Gaussian measure with a small operator norm, which (in fact) approximates the Dirac measure in the weak convergence sense. Alternative probability path, as described in Lipman et al. (2022), Albergo et al. (2022), and Liu et al. (2022) are equally valid.

Suppose $\int_0^1 \int_{\mathcal{H}} \int_{\mathcal{H} \times \mathcal{H}} \|\mathcal{G}_t(h|z)\| d\mu_t(h|z) d\pi(z) dt$ is finite to guarantee the vector field is sufficiently regular (Lipschitz continuity), where $\mathcal{G}_t(\cdot|z)$ is the conditional vector field. Under this condition, the vector field that generates μ_t as specified in Eq. B.4 and Eq. 3.3 can be expanded as follows :

$$\mathcal{G}_t(h) = \int_{\mathcal{H} \times \mathcal{H}} \mathcal{G}_t(h|z) \frac{d\mu_t(\cdot|z)}{d\mu_t}(h) d\pi(z) \quad (\text{B.5})$$

Eq. B.5 is an extension of the Theorem 1 as detailed in Kerrigan et al. (2023), and we provide the derivation in Appendix ???. We note that $\mu_t(\cdot|z)$ is a Gaussian measure and can be expressed as $\mu_t(\cdot|z) = \mathcal{N}(m_t, C_t)$, with mean m_t and trace-class covariance operator C_t . Inspired by Tong et al. (2024), we choose m_t and C_t to have the following forms:

$$m_t = t \cdot h_1 + (1 - t) \cdot h_0 \quad (\text{B.6})$$

$$C_t = \sigma_{\min}^2 C_0 \quad (\text{B.7})$$

where C_0 is the same Gaussian covariance operator defined for ν_0 and $\sigma_{\min}$ is a small constant. Further, similar to finite-dimensional flow matching, we only consider the simplest vector field that applies a canonical transformation for Gaussian measures, such that the flow has the form: $\Phi_t(h_0|z) = m_t + \sigma_{\min} h_0 \approx t \cdot h_1 + (1 - t) \cdot h_0$. From

Eq. 3.1, we can get $\mathcal{G}_t(h|z) = h_1 - h_0$, indicating $\mathcal{G}_t(h|z)$ is independent of the time t and the path from h_0 to h_1 is a direct, straight line. Equipped with well-constructed conditional vector field and probability measures, we can train a neural operator $\mathcal{G}_\theta$ with the conditional flow matching loss

$$\mathcal{L}_{\text{CFM}}^\dagger = \mathbb{E}_{t \sim \mathcal{U}[0,1], h \sim \mu_t, z \sim \pi(\nu_0, \nu_1)} \|\mathcal{G}_\theta(t, h) - \mathcal{G}_t(h|z)\|^2 \quad (\text{B.8})$$

Next, we explore how to approximate the true optimal transport plan from optimal coupling of the joint measure $\pi(\nu_0, \nu_1)$. A common way for measuring the distance between two probability measure is 2-Wasserstein distance, which a special case of static Kantorovich formulation (Kantorovich et al., 1958). The static 2-Wasserstein distance is defined as follows

$$W_{\text{sta}}(\nu_0, \nu_1)_2^2 = \inf_{\pi \in \Pi} \int_{\mathcal{H} \times \mathcal{H}} \|h_0 - h_1\|^2 d\pi(h_0, h_1) \quad (\text{B.9})$$

In the ODE framework, we also care about the dynamic form of the 2-Wasserstein distance to estimate the cost along the transport trajectory, which also is a special case of dynamic Kantorovich formulation (Chizat et al., 2018).

$$W_{\text{dyn}}(\nu_0, \nu_1)_2^2 = \inf_{\mu_t, \mathcal{G}_t} \int_{\mathcal{H}} \int_0^1 \|\mathcal{G}_t(h)\|^2 d\mu_t(h) dt \quad (\text{B.10})$$

Within the OFM framework, the marginal probability measure is a sum of Dirac measures as described in Eq. B.4, and we selected ν_0 as a Gaussian measure and assumed ν_1 has full support on the Cameron-Martin space associated with ν_0 . Furthermore, the cost function of 2-Wasserstein distance is squared L^2 norm, which is continuous by nature. According to Theorem 4.3 and Lemma 4.4 of Chizat et al. (2018), $W_{\text{sta}} = W_{\text{dyn}}$ for our specifically constructed μ_t and $\mathcal{G}_t$ in the sense of weak convergence. Therefore, to get the dynamic optimal transport plan, we only need to find a joint measure $\pi(\nu_0, \nu_1)$ that achieves the infimum in Eq. B.9. In practice, we use a minibatch approximation of optimal coupling between ν_0 and ν_1 . The above approach extends the dynamic (marginal) optimal transport framework of (Tong et al., 2024) to infinite-dimensional function space. The related work of Kerrigan et al. (2024) addresses a similar problem, but from a different perspective. For a detailed comparison, please refer to Appendix B.17.

B.7 Derivation of Eq. B.5

In this part, we show the derivation of Eq. B.5, which extends Theorem 1 of Kerrigan et al. (2023). The problem setting is given continuity equation and its weak form:

$$\int_0^1 \int_{\mathcal{H}} \frac{\partial \varphi(h, t)}{\partial t} + \langle \mathcal{G}_t(h), \nabla_h \varphi(h, t) \rangle d\mu_t(h) dt = 0, \quad \forall \varphi \in \text{Cyl}(\mathcal{H} \times [0, 1]) \quad (\text{B.11})$$

we want to derive the following form of the conditional vector field under absolute continuity assumption and other mild conditions, where $z := (h_0, h_1) \in \mathcal{H} \times \mathcal{H}$.

$$\mathcal{G}_t(h) = \int_{\mathcal{H} \times \mathcal{H}} \mathcal{G}_t(h|z) \frac{d\mu_t(\cdot|z)}{d\mu_t}(h) d\pi(z) \quad (\text{B.12})$$

First, $\int_0^1 \int_{\mathcal{H}} \frac{\partial \varphi(h, t)}{\partial t} d\mu_t(h) dt = \int_0^1 \int_{\mathcal{H}} \int_z \frac{\partial \varphi(h, t)}{\partial t} d\mu_t(h|z) d\pi(z) dt$. With continuity equation in strong form and the fact that $\mathcal{G}_t(h|z)$ induces $\mu_t(h|z)$ we have:

$$\begin{aligned} & \int_0^1 \int_{\mathcal{H}} \int_z \frac{\partial \varphi(h, t)}{\partial t} d\mu_t(h|z) d\pi(z) dt \\ &= \int_0^1 \int_{\mathcal{H}} \int_z -\nabla \cdot (\varphi(h, t) \mathcal{G}_t(h|z)) d\mu_t(h|z) d\pi(z) dt. \end{aligned}$$

By the divergence-form identity:

$$\nabla \cdot (\varphi(h, t) \mathcal{G}_t(h|z)) = \langle \mathcal{G}_t(h), \nabla_h \varphi(h, t) \rangle + \varphi(h, t) \nabla \cdot \mathcal{G}_t(h|z)$$

Since we choose the smooth test function $\varphi(h, t)$ from $\text{Cyl}(\mathcal{H} \times [0, 1])$ and use the continuity equation in weak form, we assume term $\varphi(h, t) \nabla \cdot \mathcal{G}_t(h|z)$ disappears under integration. Define

$$r_t(h, z) := \frac{d\mu_t(\cdot|z)}{d\mu_t}(h), \quad \bar{\mathcal{G}}_t(h) := \int_z \mathcal{G}_t(h|z) r_t(h, z) d\pi(z).$$

Then,

$$\begin{aligned} & \int_0^1 \int_{\mathcal{H}} \frac{\partial \varphi(h, t)}{\partial t} d\mu_t(h) dt \\ &= - \int_0^1 \int_{\mathcal{H}} \int_z \langle \mathcal{G}_t(h|z), \nabla_h \varphi(h, t) \rangle d\mu_t(h|z) d\pi(z) dt \\ &= - \int_0^1 \int_{\mathcal{H}} \int_z \langle \mathcal{G}_t(h|z) r_t(h, z), \nabla_h \varphi(h, t) \rangle d\mu_t(h) d\pi(z) dt \\ &= - \int_0^1 \int_{\mathcal{H}} \langle \bar{\mathcal{G}}_t(h), \nabla_h \varphi(h, t) \rangle d\mu_t(h) dt. \end{aligned}$$

On the other side, from Eq B.11, $\forall \varphi \in \text{Cyl}(\mathcal{H} \times [0, 1])$ we have

$$\int_0^1 \int_{\mathcal{H}} \frac{\partial \varphi(h, t)}{\partial t} d\mu_t(h) dt = - \int_0^1 \int_{\mathcal{H}} \langle \mathcal{G}_t(h), \nabla_h \varphi(h, t) \rangle d\mu_t(h) dt = 0$$

Thus $\mathcal{G}_t(h) = \int_z \mathcal{G}_t(h|z) \frac{d\mu_t(\cdot|z)}{d\mu_t}(h) d\pi(z) = \int_{\mathcal{H} \times \mathcal{H}} \mathcal{G}_t(h|z) \frac{d\mu_t(\cdot|z)}{d\mu_t}(h) d\pi(z)$

B.8 Derivation of Eq. 3.14

In this part, we show the detailed derivation of Eq. 3.14. In Flow Matching, the variable z is chosen as a single data point from the coupling $\pi(u_0, u_1)$ where $u_1 \sim q_1$, and $u_0 \sim q_0 = \mathcal{N}(\mathbf{0}, K(\{x_1, x_2, \dots, x_n\}))$. Considering the class of Gaussian conditional probability paths

$$p_t(u_t|z) = \mathcal{N}(u_t|m_t(z), \sigma_t(z)^2 K(\{x_1, x_2, \dots, x_n\})) \quad (\text{B.13})$$

With conditional flow $\phi_t(u_t|z) = \sigma_t u_0 + m_t$. Specially, we choose $m_t = tu_1 + (1-t)u_0$ and $\sigma_t = \sigma$, where $\sigma > 0$ is a small constant. From Eq. 3.1 (or Theorem 3 of Lipman et al. (2022)), a vector that defines the Gaussian conditional flow is :

$$\mathcal{G}_t(u_t|z) = \frac{\sigma'_t}{\sigma_t}(u_t - m_t) + m'_t(u_1) \quad (\text{B.14})$$

Then we can derive a closed-form expression for both the conditional probability and corresponding vector field (Tong et al., 2024) by plug in μ_t and σ_t into Eq. B.13 and Eq. B.14

$$p_t(u_t|z) = \mathcal{N}(u_t|tu_1 + (1-t)u_0, \sigma^2 K(\{x_1, x_2, \dots, x_n\})) \quad (\text{B.15})$$

$$\mathcal{G}_t(u_t|u_1) = 0 + (u_1 - u_0) = u_1 - u_0 \quad (\text{B.16})$$

Now, let's check the boundary conditions. At $t = 0$,

$$p_0(u_t|z) = \mathcal{N}(u_t|u_0, \sigma^2 K(\{x_1, x_2, \dots, x_n\})) \xrightarrow{\sigma \rightarrow 0} \delta_{u_0} \quad (\text{B.17})$$

At $t = 1$,

$$p_1(u_t|z) = \mathcal{N}(u_t|u_1, \sigma^2 K(\{x_1, x_2, \dots, x_n\})) \xrightarrow{\sigma \rightarrow 0} \delta_{u_1} \quad (\text{B.18})$$

From Eq. 3.9, we have $p_0(u_0) = \int p_0(u_t|z)\pi(z)dz = \int \delta_{u_0}\pi(u_0, u_1)du_0du_1 = q_0$ and $p_1(u_1) = \int p_1(u_t|z)\pi(z)dz = \int \delta_{u_1}\pi(u_0, u_1)du_0du_1 = q_1$, which show boundary conditions are satisfied.

B.9 Proof of Proposition 3.3.1

Proposition 3.3.1. Given noisy observations $\{\widehat{u}(x_i)\}_{i=1}^n$, the posterior distribution is

$$\log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n\right) = -\frac{\sum_{i=1}^n \|\widehat{u}(x_i) - u(x_i)\|^2}{2\sigma^2} + \log \mathbb{P}(\{u(x_i)\}_{i=1}^m) + C \quad (\text{B.19})$$

Where the constant $C = -\frac{n}{2} \log(2\pi\sigma^2) - \log \mathbb{P}(\{\widehat{u}(x_i)\}_{i=1}^n)$.

Proof. With Bayes rule, we have:

$$\mathbb{P}\left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n\right) = \frac{\mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^m\right) \cdot \mathbb{P}\left(\{u(x_i)\}_{i=1}^m\right)}{\mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n\right)} \quad (\text{B.20})$$

Taking the logarithm of Eq. B.20, we have:

$$\log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n\right) = \log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^m\right) + \log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m\right) - \log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n\right) \quad (\text{B.21})$$

Given $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\{\epsilon_i\}_{i=1}^n$ is a multivariate Gaussian, then $\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^n$ is a shifted multivariate Gaussian with mean $\{u(x_i)\}_{i=1}^n$ translated from the original multivariate Gaussian $\{\epsilon_i\}_{i=1}^n$. Due to the translation invariance property of Gaussian distribution, We have :

$$\log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^n\right) = \log \mathbb{P}\left(\{\epsilon_i\}_{i=1}^n\right) = -\frac{\sum_{i=1}^n \|\widehat{u}(x_i) - u(x_i)\|^2}{2\sigma^2} - \frac{n}{2} \log(2\pi\sigma^2) \quad (\text{B.22})$$

We notice $m > n$ and $\{\widehat{u}(x_i)\}_{i=1}^n$ only depends on $\{u(x_i)\}_{i=1}^n$, and doesn't depend on $\{u(x_i)\}_{i=n+1}^m$. Thus $\log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^m\right) = \log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n \middle| \{u(x_i)\}_{i=1}^n\right)$.

For evaluating $\log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m\right)$, which is the second part on the right-hand side of Eq. B.21, we can efficiently calculate it with the trace estimator. The third part on the right hand side of Eq. B.21 ($\log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n\right)$) represents the evidence and is constant. Thus the posterior distribution of Eq B.21 can be simplified as:

$$\log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m \middle| \{\widehat{u}(x_i)\}_{i=1}^n\right) = -\frac{\sum_{i=1}^n \|\widehat{u}(x_i) - u(x_i)\|^2}{2\sigma^2} + \log \mathbb{P}\left(\{u(x_i)\}_{i=1}^m\right) + C \quad (\text{B.23})$$

Where the constant $C = -\frac{n}{2} \log(2\pi\sigma^2) - \log \mathbb{P}\left(\{\widehat{u}(x_i)\}_{i=1}^n\right)$. $\square$

B.10 Example of Posterior Samples

In this section, we initially present the regression result of OFM in another additional N-S scenario, as illustrated in Fig B.2. Subsequently, we display more posterior samples used in the 2D regression examples. As depicted in Fig B.4, B.5, B.6, OFM successfully generates realistic posterior samples that are consistent with the ground truth and demonstrate appropriate variability. In contrast, GP regression fails to produce explainable posterior samples.

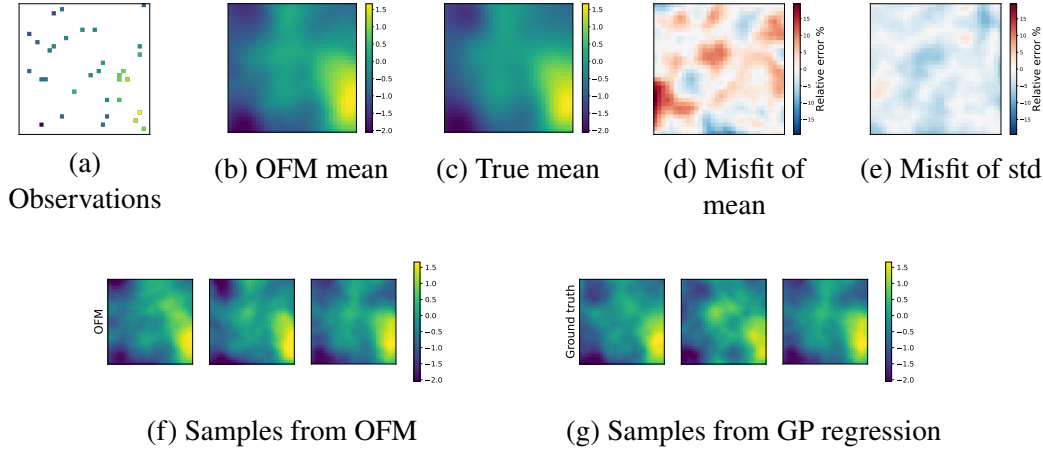


Figure B.1: OFM regression on 2D GP data with resolution 32×32 . (a) 32 random observations. (b) Predicted mean from OFM. (c) Ground-truth mean from GP regression. (d) Misfit of the predicted mean. (e) Misfit of the predicted standard deviation. (f) Posterior samples from OFM. (g) Posterior samples from GP regression.

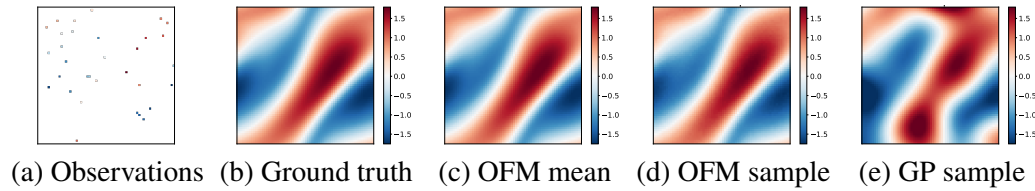


Figure B.2: OFM regression on Navier-Stokes functional data with resolution 64×64 . (a) 32 random observations. (b) Ground truth sample (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from best fitted GP.

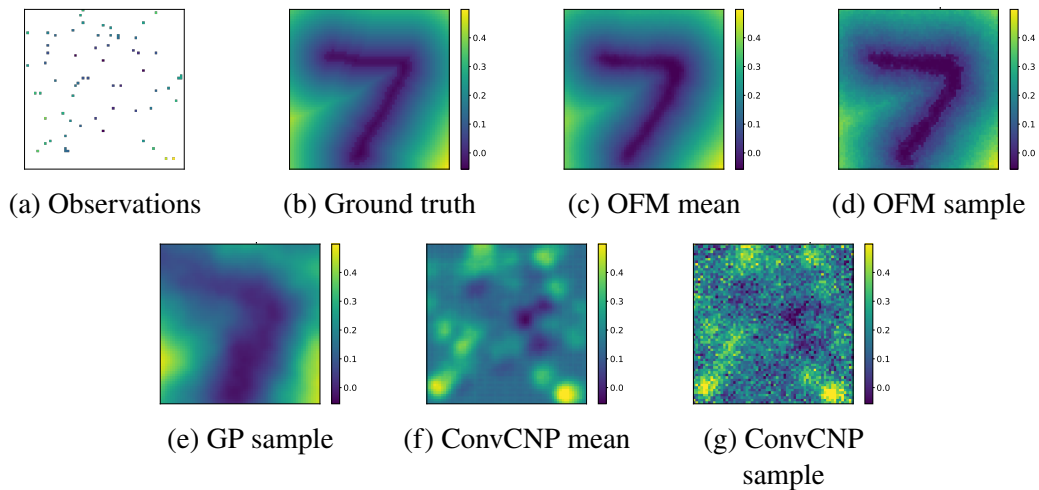


Figure B.3: OFM regression on MNIST-SDF with resolution 64×64 . (a) 64 random observations. (b) Ground-truth sample. (c) Predicted mean from OFM. (d) One posterior sample from OFM. (e) One posterior sample from the best fitted GP. (f) Predicted mean from ConvCNP. (g) One posterior sample from ConvCNP.

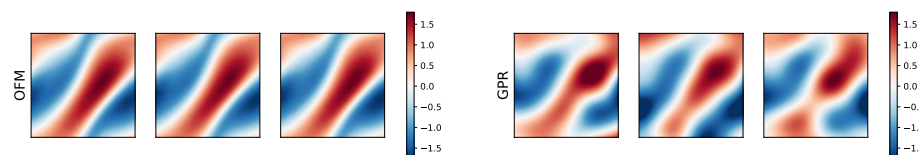


Figure B.4: OFM regression on NS data. **(left)** Posterior samples from OFM. **(right)** Posterior samples from GP regression.

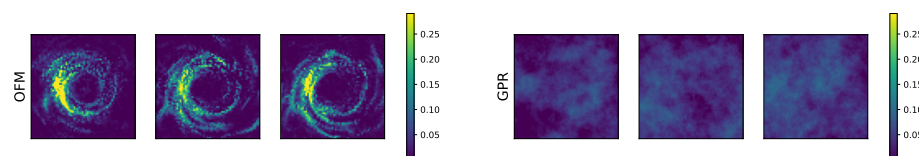


Figure B.5: OFM regression on black hole data. **(left)** Posterior samples from OFM. **(right)** Posterior samples from GP regression.

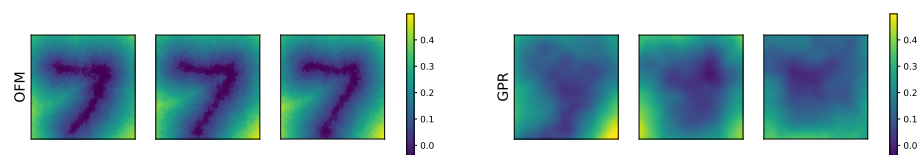


Figure B.6: OFM regression on MNIST-SDF data. **(left)** Posterior samples from OFM. **(right)** Posterior samples from GP regression.

B.11 Co-domain functional regression with OFM

In this section, we expand our regression framework to accommodate co-domain settings, as many function datasets feature a co-domain dimension greater than one. For example, earthquake waveform data commonly include three directional components, leading to a three-dimensional co-domain. Similarly, the velocity field in fluid dynamics usually features three directional components, also resulting in a dimension of co-domain of three.

We illustrate this extension through a 2D GP example with a co-domain of 3 (channel dimension of 3). In learning the prior, we define the reference measure (ν_0) as a joint measure (Wiener measure) of three identical but independent Gaussian measures while the target measure (ν_1) is another Wiener measure. We keep all other parameters unchanged as those described in the 2D GP regression tasks, with the only modification being an increase in the channel dimension from one to three. After training the prior (training detail provided in Appendix B.13), and provided 32 random observations across the three channels at co-locations, we then perform regression with OFM across these channels jointly. As demonstrated in Fig B.7, OFM accurately estimate the mean and uncertainty across three channels.

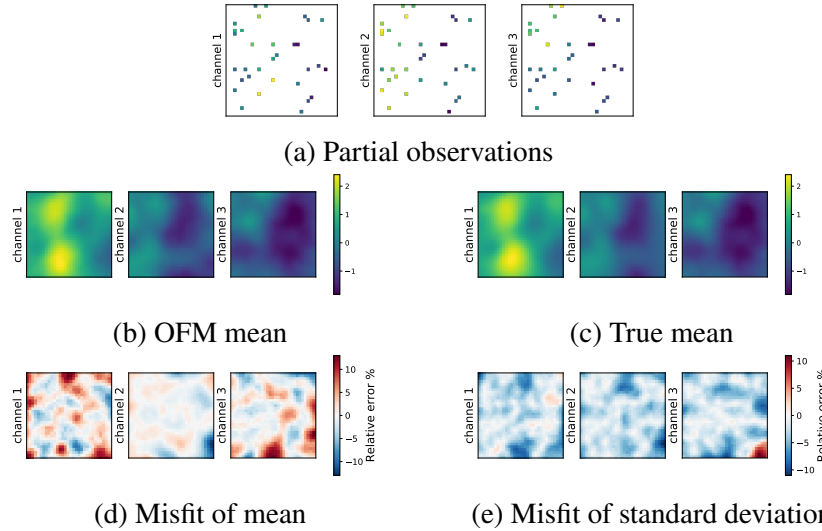


Figure B.7: OFM regression on co-domain GP data with resolution 32x32. (a) 32 random observations at co-locations. (b) Predicted mean from OFM. (c) Ground truth mean from GP regression. (d) Misfit of the predicted mean. (e) Misfit of predicted standard deviation.

B.12 Posterior sampling with Stochastic Gradient Langevin Dynamics

In this section, we describe how to sample from posterior distribution with SGLD. We denote logarithmic posterior distribution (Eq. B.23) as $\log \mathbb{P}_\theta$ and denote a set of posterior samples as $\{u_\theta^t\}_{t=1}^N$, where each u_θ^t is defined on a collection of point $\{x_i\}_{i=1}^m$.

By following the standard SGLD pipeline as described by Welling et al. (2011), we can obtain a set of N posterior samples $\{u_\theta^t\}_{t=1}^N$. However, SGLD is known to be sensitive to the choice of regression parameters and can become trapped in local minima, leading to convergence issues, especially in regions of high curvature (Li et al., 2016). To mitigate these challenges, Shi et al. (2024a) proposed that within an invertible framework, drawing a posterior sample u_θ^t is equivalent to drawing a sample a_θ^t in Gaussian space, since u_θ^t uniquely defines a_θ^t and vice versa. This approach can stabilize the posterior sampling process and is less sensitive to the regression parameters due to the inherent smoothness of the Gaussian process. Additionally, Shi et al. (2024a) suggests starting from maximum *a posteriori* (MAP) estimate of a_θ^t , denoted as $\bar{a}_\theta$, which can reduce the number of burn-in iterations needed in SGLD. We adopt the same sampling strategy and the algorithm is reported in Algorithm 1

When the size of observations or context points ($\{\widehat{u}(x_i)\}_{i=1}^n$) is 0, sampling from the posterior degrades to sampling from the prior, the results of which are presented in the subsequent section.

Algorithm 1 Posterior sampling with SGLD

Input and Parameters: Logarithmic posterior distribution $\log \mathbb{P}_\theta$, temperature T , learning rate η_t , MAP $\bar{a}_\theta$, burn-in iteration b , sampling iteration t_N , total iteration N .

- 1: **Initialization:** $a_\theta^0 = \bar{a}_\theta$
 - 2: **for** $t = 0, 1, 2, \dots, N$ **do**
 - 3: Compute gradient of the posterior: $\nabla_{a_\theta} \log \mathbb{P}_\theta$
 - 4: Update a_θ^{t+1} : $a_\theta^{t+1} = a_\theta^t + \frac{\eta_t}{2} \nabla \log \mathbb{P}_\theta + \sqrt{\eta_t T} \mathcal{N}(0, I)$
 - 5: **if** $t \geq b$ **then**
 - 6: Every t_N iterations: obtain new sample a_θ^{t+1} , and corresponding u_θ^{t+1}
 - 7: **end if**
 - 8: **end for**
-

B.13 Prior learning with OFM

We now elaborate on the prior learning process and the corresponding performance evaluation. As shown in Algorithm 2, the training dataset is sampled from the un-

known data measure ν_1 . Concretely, the training dataset consists of M discretized functions $\{u_i|D_i\}_{i=1}^M$, where $u_i|D_i$ denotes a discretized observation of the u_i function.

In practice, to simplify dataset preparation, one often uses the same discretization grid D_i for all function samples, e.g. $D_i = \{x_1, \dots, x_n\}$ regardless of the sample index "i". For the consistency of notions, let h_0 represents a batch of i.i.d discretized functions sampled from the training dataset (equivalently, sampled from ν_1). Next, the reference Gaussian process $\nu_0 = \mathcal{N}(m_0, C_0)$ is known and determined by the user. With a slight abuse of notation, We choose to use notation h_0, h_1 for consistency purpose, in other parts of this paper, discretized h_0, h_1 is replaced with a, u respectively.

For specific experiments setting, we employ Matern kernel to construct the reference GP and to prepare training datasets for 1D GP, 2D GP, and 1D TGP. We have set the kernel length $l = 0.01$ with a smoothness factor $\zeta = 0.5$ for all reference GPs. OFM maps the GP samples from reference GPs to data samples and is resolution-invariant, which means OFM can be trained with functions at any resolution and evaluated at any resolution.

Algorithm 2 Learning a prior

Input: Reference Gaussian process $\nu_0 = \mathcal{N}(m_0, C_0)$, data measure ν_1 , batch size b , small constant $\sigma_{\min}$, discretized domain $D = \{x_1, \dots, x_n\}$

```

1: while Training do
2:    $h_0 \sim \nu_0; \quad h_1 \sim \nu_1$       # sample functions of size  $b$  i.i.d from the measures
      on  $D$ 
3:    $\pi \leftarrow \text{OT}(h_0, h_1)$       # mini-batch optimal transport plan
4:    $(h_0, h_1) \sim \pi$ 
5:    $t \sim \mathcal{U}(0, 1)$ 
6:    $\mu_t \leftarrow t h_1 + (1 - t) h_0$ 
7:    $h_t \sim \mathcal{N}(\mu_t, \sigma_{\min}^2 C_0)$ 
8:    $\mathcal{L}_{\text{CFM}}^\dagger(\theta) \leftarrow \|\mathcal{G}_\theta(t, x) - (h_1 - h_0)\|^2$ 
9:    $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{\text{CFM}}^\dagger(\theta))$ 
10: end while
11: return  $\mathcal{G}_\theta$ 

```

1D GP dataset. We choose $l = 0.3$ and $\zeta = 1.5$ and generate 20,000 training samples on domain $[0, 1]$ with a fixed resolution of 256. We use autocovariance and histogram of point-wise value as metrics for evaluation. We evaluate OFM at

several different resolutions shown Fig B.8, B.9, B.10, which demonstrate OFM’s excellent capability to learn the function prior.

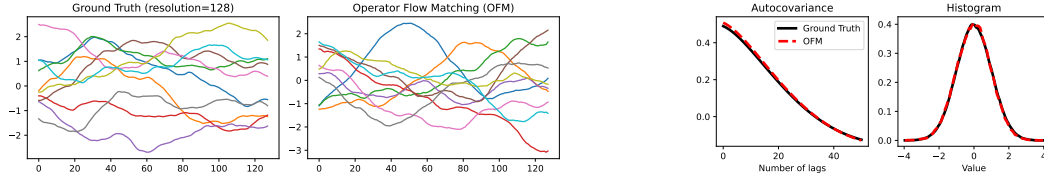


Figure B.8: OFM for 1D GP prior learning, evaluated at resolution=128. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

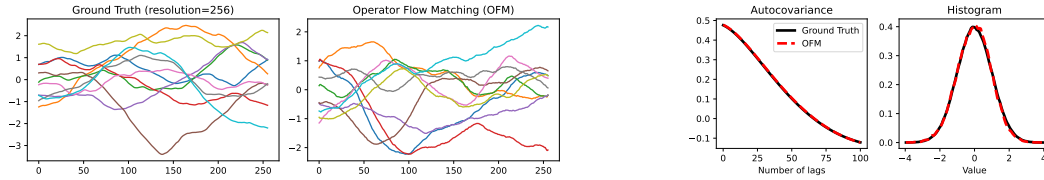


Figure B.9: OFM for 1D GP prior learning, evaluated at resolution=256. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

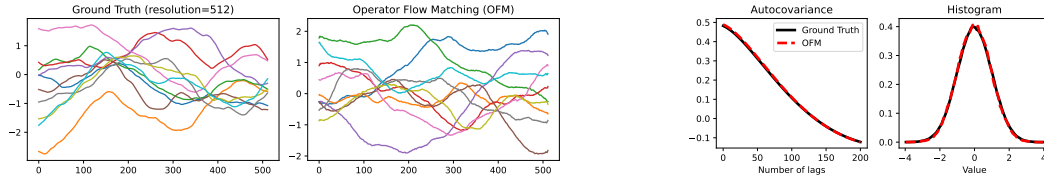


Figure B.10: OFM for 1D GP prior learning, evaluated at resolution=512. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

1D TGP dataset. We choose $l = 0.3$ and $\zeta = 1.5$ and generating 20,000 training samples on domain $[0, 1]$ with a fixed resolution of 256. We set $[-1.2, 1.2]$ for the bounds. Results provided in Fig B.11, B.12, B.13.

2D Navier-Stokes, Black hole, MNIST-SDF datasets. All the following 2D datasets are defined on domain $[0, 1] \times [0, 1]$ and have a resolution of 64×64 . We collected a 2D Navier-Stokes dataset consisting of 20000 samples, with viscosity $= 1e-4$. The results, including zero-shot super-resolution, are provided in Fig B.14, B.15. The learning of Black hole dataset, generated using expensive Monte Carlo method, is detailed in Fig B.16, B.17. Additionally, we trained OFM on 20,000 MNIST-SDF samples, the outcomes are illustrated in Fig B.18, B.19.

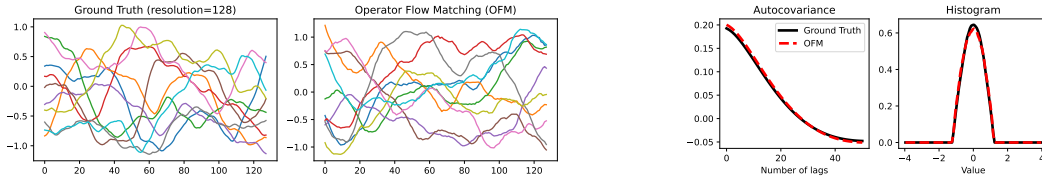


Figure B.11: OFM for 1D TGP prior learning, evaluated at resolution=128. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

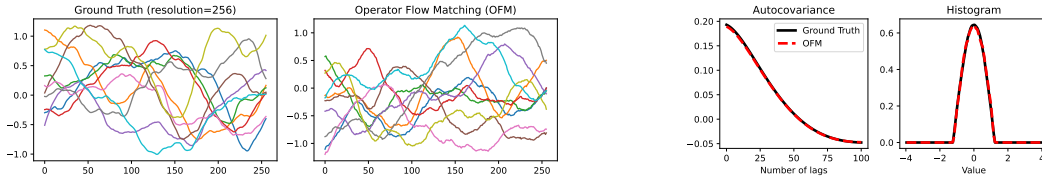


Figure B.12: OFM for 1D TGP prior learning, evaluated at resolution=256. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

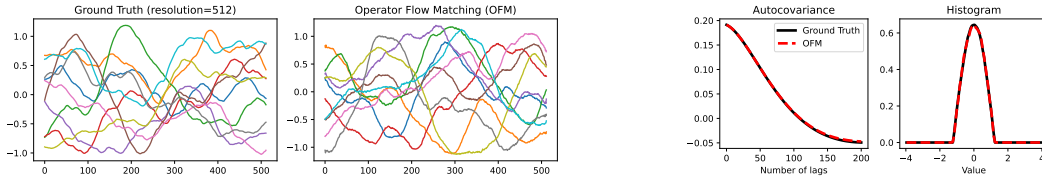


Figure B.13: OFM for 1D TGP prior learning, evaluated at resolution=512. **(left two)** Random samples from ground truth and generated by OFM. **(right two)** Autocovariance and histogram comparison

B.14 Additional results of 1D GP regression with more complex kernels

In this section, we extend the 1D GP experiments by introducing more complex kernels for both data generation and regression, while holding all other settings identical to those used with the Matérn kernel. Specifically, we report results for the non-stationary Gibbs kernel and the Rational Quadratic (RQ) kernel.

For the Gibbs kernel, we use an input-dependent length-scale

$$\ell(x) = \ell_0 + \ell_1 x, \quad x \in [0, 1],$$

which induces the covariance

$$k(x, x') = \sigma^2 \sqrt{\frac{2\ell(x)\ell(x')}{\ell(x)^2 + \ell(x')^2}} \exp\left(-\frac{(x-x')^2}{\ell(x)^2 + \ell(x')^2}\right).$$

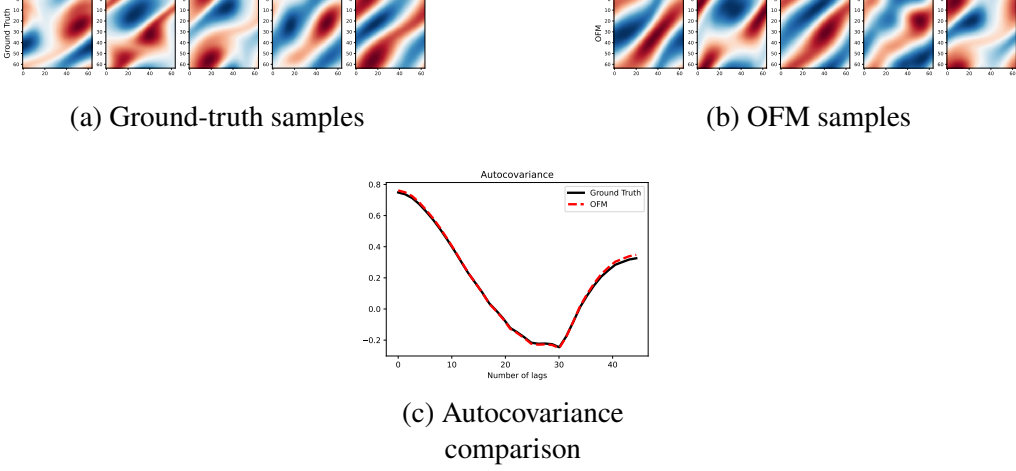


Figure B.14: OFM for 2D Navier–Stokes prior learning, evaluated at resolution 64×64 . (a) Random samples from the ground truth. (b) Random samples generated by OFM. (c) Autocovariance comparison.

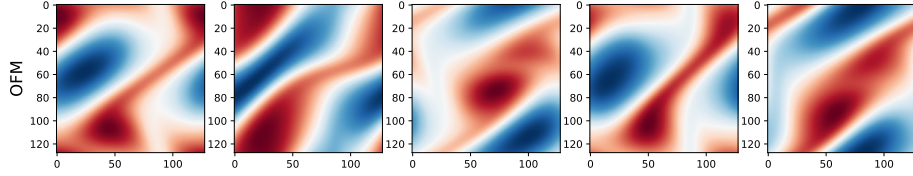


Figure B.15: OFM for 2D N-S prior learning, evaluated at 128×128 resolution (zero-shot super-resolution)

In our setup we take $\ell_0 = 0.05$, $\ell_1 = 0.25$, and $\sigma = 1.0$.

For the RQ kernel, we use `sklearn.gaussian_process.kernels.RationalQuadratic` with length-scale 0.15. As shown in Table B.1, OFM consistently outperforms all baselines on these tasks.

Table B.1: Comparison of OFM with baseline models on 1D GP with Gibbs kernel and RQK. Best performance in bold.

Dataset →	1D GP-Gibbs		1D GP-RQK	
Algorithm ↓ Metric →	SMSE	MSLL	SMSE	MSLL
NP	$4.0 \cdot 10^{-1}$	$7.1 \cdot 10^{-1}$	$5.8 \cdot 10^{-1}$	$2.2 \cdot 10^0$
ANP	$3.5 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$2.1 \cdot 10^0$
ConvCNP	$3.4 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$2.2 \cdot 10^{-1}$	$6.9 \cdot 10^{-1}$
DGP	$4.3 \cdot 10^{-1}$	$8.6 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$1.2 \cdot 10^0$
DSPP	$4.2 \cdot 10^{-1}$	$7.1 \cdot 10^{-1}$	$2.5 \cdot 10^{-1}$	$4.2 \cdot 10^{-1}$
OpFlow	$3.1 \cdot 10^{-1}$	$6.9 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$	$5.0 \cdot 10^{-1}$
OFM(Ours)	$2.9 \cdot 10^{-1}$	$8.7 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$9.4 \cdot 10^{-2}$

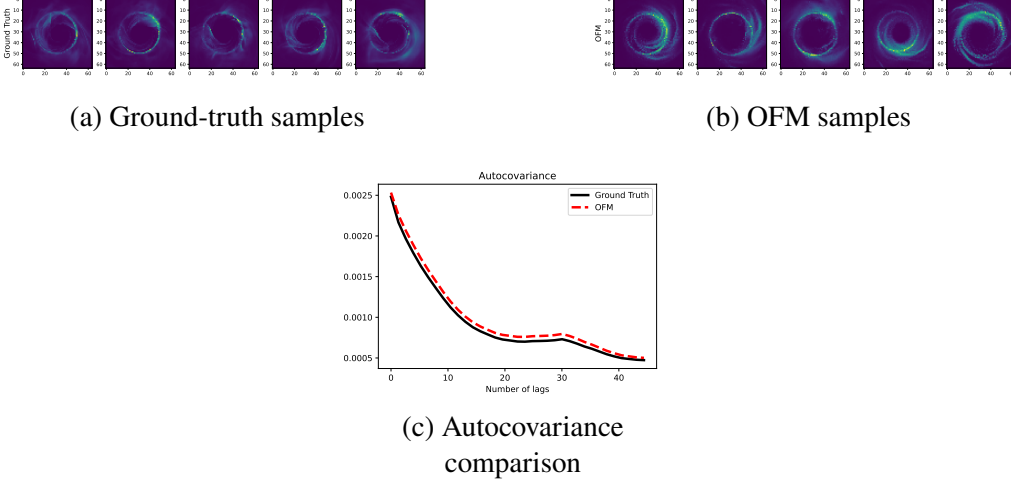


Figure B.16: OFM for 2D black hole prior learning, evaluated at resolution 64×64 . **(a)** Random samples from the ground truth. **(b)** Random samples generated by OFM. **(c)** Autocovariance comparison.

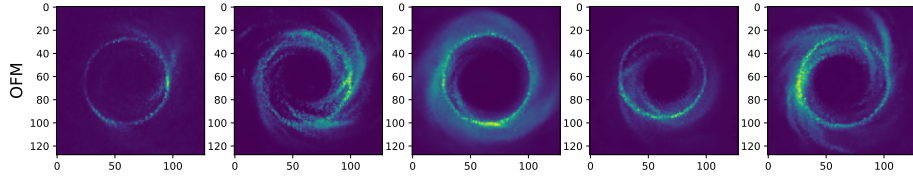


Figure B.17: OFM for 2D black hole prior learning, evaluated at 128×128 resolution (zero-shot super-resolution)

B.15 Details of experimental setup

In this section, we outline the details of experiments setup used in this paper. Since regression with OFM requires learning the prior first, we list the parameters used for learning the prior and regression separately. We employ FNO as the backbone, implemented using `neuraloperator` library (Li et al., 2020b). All time reported in the subsequent tables are based on one computations performed using a single NVIDIA RTX A6000 (48 GB) graphics card. In all experiments, we use the `dopri5` ODE solver provided by `torchdiffeq` Chen et al. (2018) with `atol=1e-5` and `rtol=1e-5`. Detailed posterior sampling algorithm is provided in Appendix B.12.

Table B.2 details the parameters used for training the prior. For instance, in the 1D GP prior learning experiment, the dataset consists of 20,000 samples, each with a co-domain dimension (or channel) of one. The batch size is set at 1024, and the model is trained over 500 epochs. The total training time is about 0.76 hours, and the size of the trained model is 37.1 megabytes.

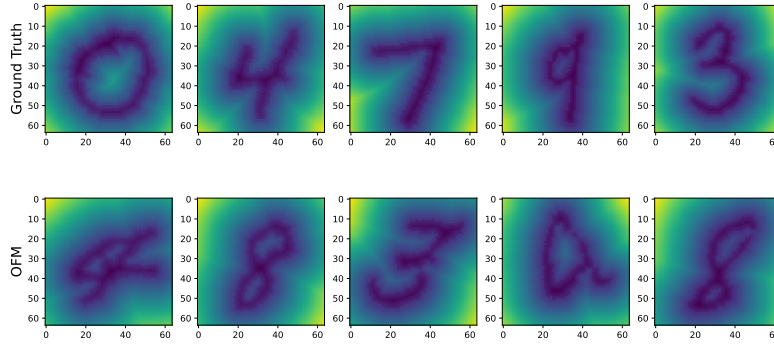


Figure B.18: OFM for 2D MNIST-SDF prior learning, evaluated at 64×64 resolution. **(top)** Random samples from ground truth. **(bottom)** Random samples generated by OFM.

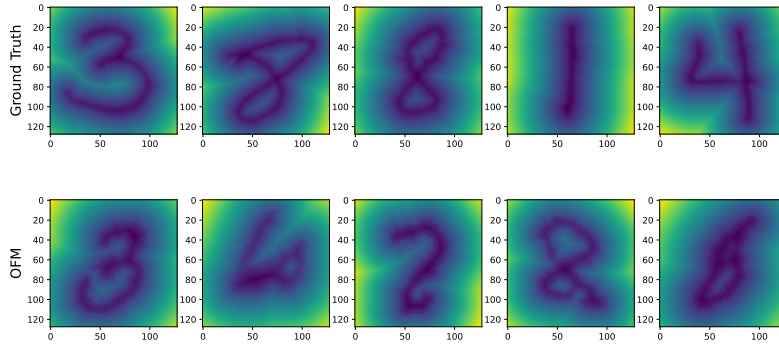


Figure B.19: OFM for 2D MNIST-SDF prior learning, evaluated at 128×128 resolution. **(top)** Random samples from ground truth. **(bottom)** Random samples generated by OFM.

Tables B.3, B.4, and B.5 detail the parameters for SGLD sampling as described in Algorithm 1. For example, in the 1D GP regression experiment, the regression takes 40,000 iterations with a burn-in phase of 3,000 iterations. Posterior samples are collected every 10 iterations. The temperature for the injected noise during the gradient update is set at 1, and the learning rate decays exponentially from 0.005 to 0.004 (defined in Algorithm 1). We average 32 runs with the Hutchinson trace estimator to evaluate the likelihood, utilizing GPU parallel computing. The noise level, as specified in Equation B.23, is 0.01 in this regression task. Then given 6 random observations, we ask for the posterior samples across 128 points. The GPU memory usage for the regression task is 4 gigabytes, with the total runtime to 4.91 hours.

Table B.2: Parameters used in experiments of prior learning

Datasets	Size of Dataset	Channels	Batch Size	Epochs	Training Time	Model Size
1D GP	$2 \cdot 10^4$	1	1024	$5 \cdot 10^2$	0.76 h	37.1 MB
1D TGP	$2 \cdot 10^4$	1	1024	$5 \cdot 10^2$	1.24 h	37.1 MB
2D GP	$2 \cdot 10^4$	1	256	$5 \cdot 10^2$	1.14 h	76 MB
2D co-domain GP	$2 \cdot 10^4$	3	256	$5 \cdot 10^2$	1.01 h	76 MB
2D N-S	$2 \cdot 10^4$	1	256	$5 \cdot 10^2$	3.79 h	286 MB
2D Black hole	$1.2 \cdot 10^4$	1	256	$5 \cdot 10^2$	2.28 h	286 MB
2D MNIST-SDF	$2 \cdot 10^4$	1	256	$5 \cdot 10^2$	8.31 h	286 MB

Table B.3: Parameters used in regression experiments - Part A

Datasets	Total Iteration	Burn-in Iteration	Sampling Iterations	Temperature of Noise
1D GP	$4 \cdot 10^4$	$3 \cdot 10^3$	10	1
1D TGP	$4 \cdot 10^4$	$3 \cdot 10^3$	10	1
2D GP	$2 \cdot 10^4$	$3 \cdot 10^3$	10	1
2D co-domain GP	$2 \cdot 10^4$	$3 \cdot 10^3$	10	1
2D N-S	$2 \cdot 10^4$	$3 \cdot 10^3$	10	1
2D Black hole	$2 \cdot 10^4$	$3 \cdot 10^3$	10	1
2D MNIST-SDF	$2 \cdot 10^4$	$3 \cdot 10^3$	10	1

Table B.4: Parameters used in regression experiments - Part B

Datasets	Initial Learning Rate	End Learning Rate	Hutchinson Samples	Noise Level
1D GP	$5 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	32	$1 \cdot 10^{-2}$
1D TGP	$5 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	32	$1 \cdot 10^{-3}$
2D GP	$1 \cdot 10^{-3}$	$8 \cdot 10^{-4}$	32	$1 \cdot 10^{-2}$
2D co-domain GP	$1 \cdot 10^{-3}$	$8 \cdot 10^{-4}$	16	$1 \cdot 10^{-2}$
2D N-S	$3 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	8	$1 \cdot 10^{-3}$
2D Black hole	$5 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	8	$1 \cdot 10^{-3}$
2D MNIST-SDF	$5 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	8	$1 \cdot 10^{-3}$

Table B.5: Parameters used in regression experiment - Part C

Datasets	Number of Observations	Inquired Grids	GPU Memory	Running Time
1D GP	6	128	4 GB	4.91 h
1D TGP	3	128	4 GB	5.42 h
2D GP	32	32×32	22 GB	9.70 h
2D co-domain GP	32	32×32	31 GB	5.05 h
2D N-S	32	64×64	44 GB	13.65 h
2D Black hole	32	64×64	44 GB	13.37 h
2D MNIST-SDF	64	64×64	44 GB	9.41 h

B.16 Ablation and scaling studies for mini-batch optimal transport and Hutchinson trace estimator

We first present an ablation study of the optimal transport plan. In this study, we revisit prior learning on the N-S dataset by training two models: one using independent coupling and the other employing a mini-batch optimal transport plan. Both models were trained with a batch size of 64. For evaluation, we compare the

mean squared error (MSE) of density, autocovariance and spectral characteristics between 1,000 real and generated samples. Additionally, we report the convergent training loss (squared L^2 loss) and the number of function evaluations (NFE) required for sampling with adaptive ODE solver (dopri5). As shown in Table B.6, the mini-batch OT plan outperforms the independent coupling approach in terms of pointwise accuracy, spectral fidelity, and convergent L^2 loss, while also requiring fewer NFE and enabling faster sampling. Our findings indicate that as the mini-batch size increases, the model learns the prior with reduced error. This improvement may be attributed to the mini-batch optimal transport plan approaching the true optimal transport plan with larger batch sizes.

Next, we explore the variance of the Hutchinson trace estimator in likelihood estimation with a scaling experiment. For this experiment, we use a 1D GP example with parameters described in Section B.13 and a resolution of 128. We randomly draw a GP sample from the prior and evaluate the log likelihood by integrating the divergence as shown in Eq. 3.15, while also estimating it using the Hutchinson trace estimator from Eq. 3.16. In this scaling experiment, the number of noise sample (n_{noise}) for the Hutchinson estimator is set to (4, 8, 16, 32, 64, 128). For each n_{noise} , we repeat the experiment 100 times and report the mean and standard deviation of the predicted likelihood. The exact likelihood is computed by directly integrating the trace of the Jacobian. We repeat this procedure for 5 different random 1D GP samples. As reported in Table B.7, the standard deviation of the Hutchinson trace estimator decreases rapidly as n_{noise} increases, and the predicted means always align closely with the ground truth.

Furthermore, even at smaller n_{noise} values, where a relative larger variance is expected, the performance of the posterior sampling appears robust (see Table B.4). We hypothesize that this is due to stochastic nature of posterior sampling algorithm (SGLD), which requires injected perturbations, renders its performance relatively insensitive to the choice of n_{noise} .

Table B.6: Scaling study for the size of mini-batch given optimal transport plan, best performance in bold

Metrics	Density-MSE	Autocovariance-MSE	Spectra-MSE	Convergent L^2 Loss	NFE
Independent	$3.4 \cdot 10^{-5}$	$7.4 \cdot 10^{-5}$	$1.3 \cdot 10^1$	$5.1 \cdot 10^{-2}$	168
mini-batch = 32	$3.0 \cdot 10^{-5}$	$1.3 \cdot 10^{-4}$	$5.0 \cdot 10^0$	$2.3 \cdot 10^{-2}$	141
mini-batch = 64	$9.8 \cdot 10^{-6}$	$9.8 \cdot 10^{-5}$	$6.3 \cdot 10^0$	$1.9 \cdot 10^{-2}$	153
mini-batch = 128	$2.4 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$	$3.6 \cdot 10^0$	$1.5 \cdot 10^{-2}$	182

Table B.7: Scaling study for the number of noise samples of Hutchinson trace estimator

	$n_{\text{noise}} = 4$	$n_{\text{noise}} = 8$	$n_{\text{noise}} = 16$	$n_{\text{noise}} = 32$	$n_{\text{noise}} = 64$	$n_{\text{noise}} = 128$	exact
sample 1	-481.7 ± 9.4	-482.9 ± 7.7	-480.6 ± 4.9	-481.4 ± 3.8	-481.1 ± 2.5	-480.8 ± 1.8	-482.7
sample 2	-482.8 ± 9.8	-481.8 ± 6.7	-481.3 ± 4.9	-480.8 ± 3.5	-481.1 ± 2.7	-481.2 ± 1.7	-482.0
sample 3	-479.7 ± 10.2	-479.7 ± 7.6	-478.2 ± 4.7	-478.6 ± 3.4	-478.8 ± 2.6	-478.8 ± 1.7	-479.1
sample 4	-476.9 ± 10.0	-477.0 ± 6.6	-477.9 ± 5.6	-478.5 ± 4.0	-478.1 ± 2.7	-478.0 ± 1.8	-477.4
sample 5	-479.4 ± 10.7	-479.2 ± 6.6	-479.2 ± 5.3	-479.4 ± 3.6	-479.6 ± 2.8	-479.3 ± 2.0	-478.5

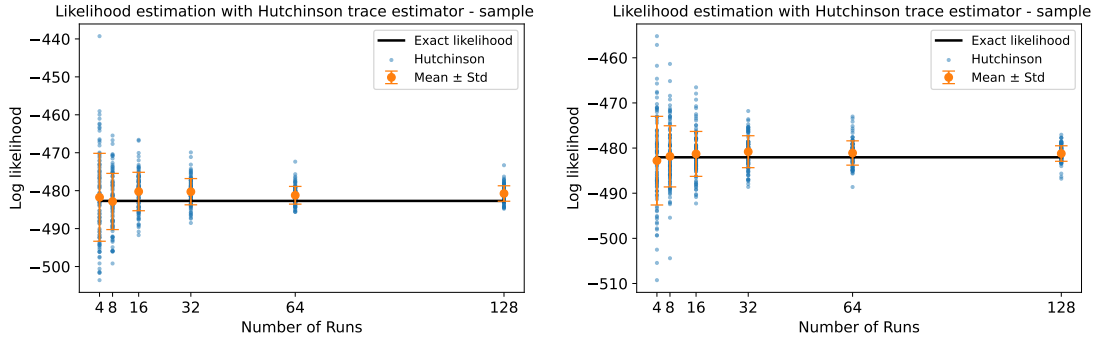


Figure B.20: Log likelihood by the integration of divergence, (left) plot for first GP sample. (right) plot for second GP sample

B.17 Detailed analysis of OFM and comparison with existing methods

In this section, we elaborate the connection and difference with pervious work, highlight contributions and potential limitations of our work. The regression with OFM involves a two-steps process: (i) learning a prior on function space, and (ii) sampling from the posterior given observations. Consequently, the OFM framework has connections with both generative models on function space and the models developed for functional regression. In the following, we provide a comprehensive comparative analysis with related models and baselines, including operator flow (OpFlow) (Shi et al., 2024a), conditional optimal transport flow matching (COT-FM) (Kerrigan et al., 2024), conditional models (NPs) (Garnelo et al., 2018; Dutordoir et al., 2023)

Comparison with OpFlow. OpFlow introduces invertible neural operators, which generalizes RealNVP (Dinh et al., 2016) to function space and maps any collection of points sampled from a GP to a new collection of points in the data space, using the maximum likelihood principle (Shi et al., 2024a). This method captures the likelihood of any collection of point consistently as the resolution increases and allows for UFR using SGLD. Despite these advantages, the requirement for an invertible neural operator brings training and expressiveness challenges. To be specific, OpFlow failed on all non-GP regression tasks in this paper because the prior

learning stage suffered from mode collapse during training. On the contrary, OFM adopts a simulation-free ODE framework for prior learning, which offers enhanced expressiveness and ensures training stability through a simple regression objective while avoiding using the invertible neural operator. In addition, OFM proposes a non-trivial extension of UFR to the simulation-free ODE framework. These improvements render OFM a more practical solution for challenging functional regression tasks.

Comparison with COT-FM. COT-FM (Kerrigan et al., 2024) proposes a conditional generalization of Benamou-Brenier Theorem (Benamou et al., 2000), formulating a conditional optimal transport plan that applicable for both Euclidean and Hilbert space. In contrast, OFM employs an unconditional optimal transport plan in Hilbert space based on dynamic Kantorovich formulation, which is initially generalized for unbalanced optimal transport (Chizat et al., 2018). The advantage of COT-FM lies in its ability to flexibly incorporate specific conditions tailored for conditional generative tasks. However, COT-FM is not suitable for functional regression tasks due to: (i) COT-FM is contingent upon both the reference and target being influenced by conditions, and the vector field learnt is triangular, designed to transport jointly the coupling of a reference measure and a condition measure. In UFR setting, the learnt prior is required to be unconditioned, (ii) the coupling with condition measure typically prevents inducing valid stochastic process, even when the reference measure is a Gaussian measure, (iii) cannot provide point evaluation of probability density. Last, We should notice, the development of OFM is different and independent of COT-FM, the former with a focus on stochastic process learning and Bayesian functional regression.

Comparison with conditional models. NPs were developed to address the computational and restrictive prior challenges of Gaussian Processes, utilizing neural networks for efficiency (Garnelo et al., 2018). However, several recent studies have discussed the drawbacks in the formulation of NPs, raising concerns that NPs might not learn the underlying function distribution (Rahman et al., 2022; Dupont et al., 2021; Shi et al., 2024a).

Notably, NPs treats the point cloud data as a set of values, ignoring the metric space of the data (Dupont et al., 2021). This can lead to misinterpretations of a function sampled at different resolutions as distinct functions (Appendix A.1 of (Rahman et al., 2022)). Furthermore, NPs rely on encoding input data into finite-dimensional, Gaussian-distributed latent variables before projecting these into an

Table B.8: High-level comparison of OFM and NP.

Aspect	OFM	Neural Processes (NP)
Modeling target	Global <i>prior</i> over stochastic processes with a valid joint density.	Amortized <i>conditional</i> model without explicit tractable joint density over full functions.
Arbitrary queries	Set-size / location agnostic; evaluate on any grid or point set.	Set-size / location agnostic; trained with random context/target splits. Performance may degrade under extreme sparsity or distribution shift
Uncertainty & likelihoods	Tractable log-densities and posterior sampling enable principled Bayesian inference.	Predictive densities are available and trained via conditional likelihood; no closed-form function-level likelihood is typically specified. Uncertainty reflects model/context coverage and may be miscalibrated.
Structure & extrapolation	Captures global geometry and non-stationarity via the learned flow.	Learns inductive biases from data through context summaries; strong interpolation, but extrapolation and long-range generalization are not guaranteed and depend on training distribution.

infinite-dimensional space. This process tends to lose consistency at higher resolutions. Moreover, the Bayesian regression framework underpinning NPs focuses on point sets rather than the functions themselves, leading to a dilution of prior information with increasing data points.

In recent study, diffusion-based variants of NPs (NDP) (Dutordoir et al., 2023), was proposed to leverage the expressiveness of diffusion models (Ho et al., 2020; Song et al., 2020). Nonetheless, the formulation of NDP does not address the aforementioned issues of NPs and introduces two more problems: (i) NDP fails to induce a valid stochastic process as it does not satisfy the marginal consistency criterion required by Kolmogorov Extension Theorem (Kolmogorov, 2018), and (ii) it relies on uncorrelated Gaussian noise for denoising, which is not applicable in function spaces (Lim et al., 2025). Oppositely, OFM establishes a more theoretically sound framework by rigorously defining learning within function spaces. Additionally, Bayesian functional regression within the OFM framework adheres to valid stochastic processes, offering a robust and theoretically grounded solution. Last, we provide a high-level comparison of OFM and NP as shown in Table B.8.

Limitations. Despite these advances, the current regression framework with OFM is primarily limited to low-dimensional data (1D and 2D in this study). This limitation stems from the challenges associated with learning operators for functions defined on high-dimensional domains—an area that remains underdeveloped both computationally and in terms of dataset availability (Kovachki et al., 2023). Additionally, while the time complexity for regression with OFM is $O(m^2)$, the incorporation of additional components significantly increases its computational resource requirements compared to classical GP regression.

There are several potential paths to mitigate the computation concerns: (i) Carefully adjusting the step size (η_t) and temperature (T) in the SGLD algorithm to prevent overly large gradient updates. (ii) Running the trace estimator on multiple GPUs, use mixed-precision FNO (Sitzmann et al., 2020), or adopt efficient neural operators (Shi et al., 2025). (iii) Using more efficient ODE solvers. Our current implementation uses a high-order ODE solver (dopri5) that requires over 100 steps for sampling. Switching to a more modern and efficient solver, such as the DPM-Solver (Lu et al., 2022), could potentially reduce the number of evaluations, freeing up significant GPU resources.

APPENDIX FOR CHAPTER 4

C.1 Experimental setup**Datasets description**

We curated a benchmark of six challenging functional datasets to evaluate performance across different domain types, as summarized in Table 4.1.

Navier-Stokes. This dataset contains solutions to the 2D Navier-Stokes equations on a torus at a resolution of 64×64 . Following the pre-processing of previous work (Kerrigan et al., 2023; Shi et al., 2026b), we use 30,000 samples for training and 5,000 for testing, drawn from the original dataset introduced in (Li et al., 2020b).

Shallow Water. This dataset contains solutions to the shallow-water equations for a 2D radial dam-break scenario on a square domain, from PDEBench (Takamoto et al., 2022). Each of the 1,000 simulations has 1,000 time steps at 128×128 resolution; we downsample spatially to 64×64 for efficiency and treat each time step as an independent snapshot. We randomly select 30,000 snapshots for training and 5,000 for testing

Darcy Flow. This dataset contains steady-state solutions of 2D Darcy Flow over the unit square, obtained directly from the PDEBench benchmark (Takamoto et al., 2022). We downsample the original 128×128 resolution to 64×64 . The dataset contains 10,000 samples and we split it into 8,000 samples for training and 2,000 for testing.

Cylinder Flow. We use the Cylinder Flow dataset of Han et al. (2022), which describes flow past a cylinder on a fixed mesh of 1,699 nodes. Each sample is a 3-channel function (x-velocity, y-velocity, pressure). From 101 simulations $\times$ 400 time steps, we ignore temporal order and treat each time step as an independent sample, randomly selecting 30,000 training and 5,000 testing samples.

Mesh GP. This is a synthetic dataset generated on a fixed irregular mesh of 3,727 nodes provided by (Zhao et al., 2022). We generate function samples from a Gaussian Process (GP) with a Matérn kernel (length scale = 0.4, smoothness factor = 1.5) given the domain, creating a training set of 30,000 samples and a test set of 5,000 samples.

Global Climate. We use the real-world global climate dataset from (Dupont et al., 2021), which contains global temperature measurements over the last 40 years. Each data sample is a function defined on a grid of 46×90 evenly spaced latitudes and longitudes. Following the previous pipeline (Dupont et al., 2021), we convert the latitude-longitude pairs to Euclidean coordinates ($\mathbb{R}^3$) before passing them to the models. The dataset contains 9,676 training samples and 2,420 test samples.

Baselines description

For all baseline models, we adopt their official implementations to ensure reproducibility; the corresponding code repositories are linked in the referenced papers. The key initialization hyperparameters for the baselines : GINO (Li et al., 2023), UPT (Alkin et al., 2024), Transolver (Wu et al., 2024), LNO (Wang et al., 2024b), and FNO (Li et al., 2020b)—are detailed in Table C.1. Note that a slightly different configuration was used for the Global Climate experiment. For a complete understanding of all arguments, we refer the reader to the official repository for each respective model (github link provided in all referred papers)

Table C.1: Hyperparameter settings for all baseline models.

Hyperparameter	GINO	UPT	Transolver	LNO	FNO
Parameters	19.7 M	19.6 M	15.0 M	22.2 M	20.6 M
GNN radius	-	0.07	-	-	-
GNO radius	0.06	-	-	-	-
Width/Dim.	128	192	512	512	128
Blocks/Layers	4	12	10	8×4	4
Attn. Heads	-	3	6	8	-
Latent Grid	32×32	-	-	-	-
Latent Tokens	-	256	-	256	-
Fourier Modes	16	-	-	-	24
FNO Channels	180	-	-	-	128
Slice Number	-	-	24	-	-

Hyperparameters for MINO

For the decoder, the single cross-attention block is followed by a multi-head self-attention (MHSA) block. Although this MHSA block is optional, we found that its inclusion slightly improves performance for generation task. To maintain the overall linear time complexity of MINO, the MHSA module can be replaced by any self-attention variant with linear complexity (like transolver layer).

MINO-T. The GNO maps input functions to a latent representation on a 16×16 grid of query points, defined over the $[0, 1]^2$ domain. We set the GNO search radius to 0.07, the latent dimension L_{dim} to 256, and the number of attention heads to 4. The encoder consists of $M_1 = 5$ cross-attention blocks. For the Global Climate dataset, which is defined on a spherical manifold $\mathbb{S}^2$, we adjust the latent query positions to a 32×16 spherical grid and increase the GNO radius to 0.2 to account for the different coordinate system; other hyperparameters remain unchanged. The total parameter count for MINO-T is 21.5 M.

MINO-U. The GNO maps input functions to a latent representation on a 16×16 grid of query points, defined over the $[0, 1]^2$ domain. We set the GNO search radius to 0.07, the latent dimension L_{dim} to 256, and the number of attention heads to 4. The encoder consists of $M_1 = 2$ cross-attention blocks. For its latent-space processor, we adopt a Diffusion U-Net architecture from the `torchcfn` library (Tong et al., 2024), which operates on the $[16, 16]$ latent tensor with 64 channels, 1 residual block, and 4 attention heads for the processor. Same setting of MINO-T on the Global Climate dataset. The total parameter count for MINO-U is 19.2 M.

Details for training and inference

Reference Measures. We choose μ_0 as a Gaussian measure characterized by a Gaussian Process (GP) with a Matérn kernel. Unless otherwise specified, the function domain is $[0, 1]^2$ (or subset of it), and we use a kernel length scale of 0.01 and a smoothness parameter of 0.5. For the Global Climate dataset, the domain is a spherical manifold represented as a subset of $[-1, 1]^3$, and the kernel length scale is adjusted to 0.05. Furthermore, for the GP on the sphere, we explored two distance metrics for the kernel. The first uses the chordal distance (the Euclidean distance in the $\mathbb{R}^3$ embedding space), while the second employs the geodesic distance on $\mathbb{S}^2$, for which we leverage the `GeometricKernels` library (Mostowsky et al., 2025). Although both approaches yield a valid GP, the results presented in this paper are based on the chordal distance implementation. We selected this method for two primary reasons: its superior computational speed and to ensure consistency with other benchmarks. To facilitate further exploration, we will release the code repository with both implementations. The variance is fixed at 1 for all GPs.

Training Details. We train all models for 300 epochs using the AdamW optimizer with an initial learning rate of $1e-4$. We employ a step learning rate scheduler that decays the learning rate by a gamma of 0.8 every 25 epochs. The default batch size

is 96. However, Transolver consumes significantly more GPU memory than other models. To accommodate it on a single NVIDIA RTX A6000 Ada GPU (48 GB memory), we reduced its batch size to 48 and, to maintain a comparable training iteration (duration), limited its training to 200 epochs. Despite these adjustments, Transolver still required approximately 1.76x more total GPU-hours than MINO-T and MINO-U, respectively. The only exception was the Global Climate experiment, where we explicitly matched the total GPU computation time of MINO variants to that of Transolver by extending their training epochs. All experiments reported in Table 4.4 and Table 4.3 were conducted three times, and we report the best performance (among the three) achieved for each model. To further strengthen our comparative analysis, Appendix C.7 details an experiment with matched settings (parameter counts, batch size, epochs) that reveals MINO’s significant speedup and superior performance over Transolver.

Inference Details. To evaluate the models, we generate the same number of samples as contained in the test set for each dataset shown in Table 4.2. All samples are generated by solving the learned ODE numerically using the `dopri5` solver from the `torchdiffeq` library (Chen et al., 2018), with an error tolerance set to $1e-5$ for all experiments.

C.2 SWD and MMD as general metrics for functional generation tasks

Unlike image generation, where established metrics like the Fréchet Inception Distance (FID) can leverage a well pretrained models to compare distributions in a latent space, the field of functional generative models lacks such standardized evaluation protocols. Functional data covers a wide range of modalities, often without a common pretrained model, what’s more, the learnt objective is probability measure not probability distribution, which requires the metrics should be consistent regardless of discretization of function sample. This makes fair and direct comparison between models challenging, as previous studies frequently rely on dataset-specific metrics. To address this gap, our work proposes and validates two general metrics that directly estimate the distance between a learned probability measure, ν , and a target measure, μ , using samples drawn from each.

We are given two sets of i.i.d. function samples, $X = \{x^{(j)}\}_{j=1}^N$ drawn from ν and $Y = \{y^{(j)}\}_{j=1}^N$ drawn from μ . Each function is observed at the same N_{in} locations. To compute the distance, each function sample is first flattened into a single vector representation.

Sliced Wasserstein Distance (SWD). In practice, we estimate the distance between two probability measure, μ and ν , using the discretized function samples in the datasets X and Y . To make the metric shown in Eq. 4.9 computable, we need to derive the discretized version of it, which is achieved by choosing the base measure γ_s to be the Haar measure (uniform surface measure) on the sphere $\mathbb{R}^{d-1}$ (d is the discretization of functions and $d = N_{in}$ if co-domain is 1 in our case) and replace the integral and normalization with an expectation with respect to the probability measure defined by normalizing γ_s . Finally, we can get a discretized version of Eq. 4.9:

$$\text{SWD}_p(\mu, \nu) = \left(\mathbb{E}_{\theta \sim \mathcal{U}(\mathbb{S}^{d-1})} [\mathcal{W}_p^p(\theta_{\#}\mu, \theta_{\#}\nu)] \right)^{\frac{1}{p}} \quad (\text{C.1})$$

where θ is a random direction on the unit sphere $\mathbb{S}^{d-1}$. SWD converges to the exact p-Wasserstein distance for two probability measures as the number of projections goes to infinity. In our experiments, we use the official implementation from the POT library (Flamary et al., 2021). We use the Sliced Wasserstein-2 distance ($p = 2$) (Bonneel et al., 2015), the time complexity for SWD is $\mathcal{O}(L \cdot N \cdot N_{in} + L \cdot N \log N)$, where L is the number of projections.

Maximum Mean Discrepancy (MMD). For MMD (Gretton et al., 2012), the time complexity is $\mathcal{O}(N^2 \cdot N_{in})$ we use the Gaussian RBF kernel $k(\cdot, \cdot)$. Given the sample sets X and Y , we compute the squared MMD using the standard unbiased U-statistic estimator:

$$\begin{aligned} \widehat{\text{MMD}}_u^2(X, Y) &= \frac{1}{N(N-1)} \sum_{i \neq j} k(x^{(i)}, x^{(j)}) \\ &\quad + \frac{1}{N(N-1)} \sum_{i \neq j} k(y^{(i)}, y^{(j)}) \\ &\quad - \frac{2}{N^2} \sum_{i=1}^N \sum_{j=1}^N k(x^{(i)}, y^{(j)}). \end{aligned} \quad (\text{C.2})$$

A practical consideration for the SWD is that it relies on a Monte Carlo approximation over random projections. This can introduce variance; calculating the SWD multiple times on the same two test datasets may yield slightly different results. To ensure our evaluations are stable and meaningful, we must reduce this variance to a negligible level. We propose a simple procedure we term **averaged-SWD**: instead of a single SWD estimation, we perform n_{run} independent estimations via Eq C.1 (each with a large number of projections, $L = 256$ in our case) and report the

mean, we would expect averaged-SWD has very small variance introduced by the Monte Carlo approximation.

To validate this approach, we created two distinct Gaussian measures, characterized by GP, GP₁ (length scale=0.3, smoothness=1.5) and GP₂ (length scale=0.01, smoothness=0.5), on a 64×64 regular grid. We drew 5,000 GP samples from each measure to create X, Y and then calculated the averaged-SWD between them for varying n_{run} and repeated this entire process 20 times to report the mean and standard deviation of the averaged-SWD. As shown in Table C.2, when $n_{\text{run}} = 10$, the variance of the averaged-SWD is sufficiently small. We therefore adopt $n_{\text{run}} = 10$ for all SWD computations in our experiments. In contrast, MMD is deterministic for a given kernel and does not exhibit this variance.

Metric Consistency Across Discretizations. A crucial property of a metric for functional data is its consistency with respect to the discretization of the function domain. The computed distance between two measures should be consistent regardless of the discretization for functions samples in two function datasets drawn from the two probability measure. That is, if we draw new function samples $X_{\dagger}$ and $Y_{\dagger}$ from the same measures μ and ν but observe them on a different set of locations, we expect $\text{MMD}(X_{\dagger}, Y_{\dagger}) \approx \text{MMD}(X, Y)$ and $\text{SWD}(X_{\dagger}, Y_{\dagger}) \approx \text{SWD}(X, Y)$.

To verify this property, we calculated SWD and MMD in three representative scenarios while varying the number of observation points by randomly sub-sampling the full discretization (N_{in}). The scenarios are: (i). GP₁ vs. GP₂. (ii). GP₁ vs. Navier-Stokes (test dataset) .(iii) GP₁ vs. Cylinder Flow (test dataset). For each case, we will unify the observed position (discretization) of function samples

As shown in Fig. C.1, both MMD and SWD remain highly consistent across different sub-sampling ratios of the observation points. This analysis demonstrates that SWD and MMD are not only theoretically sound but also empirically robust and sample-efficient evaluation metrics for functional generative models.

Table C.2: Variance analysis of the averaged-SWD metric. We report the mean and standard deviation over 20 trials for different numbers of averaging runs (n_{run}).

n_{run}	5	10	20	40	60
SWD, mean $\pm$ std	0.2482 ± 0.0052	0.2485 ± 0.0038	0.2494 ± 0.0029	0.2497 ± 0.0017	0.2498 ± 0.0014

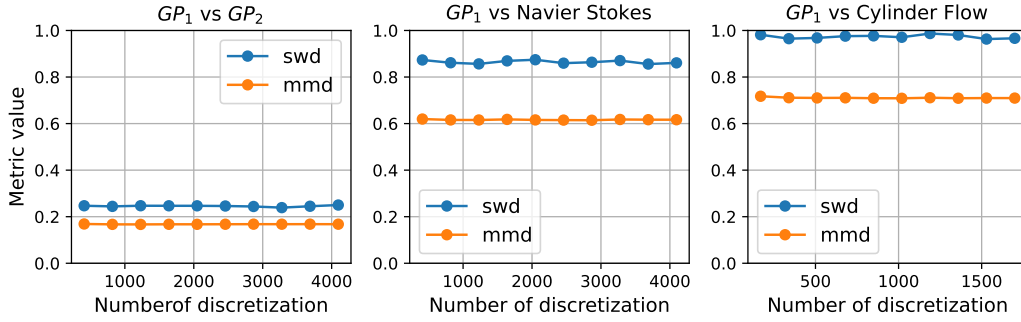


Figure C.1: Consistency of SWD and MMD with respect to the number of discretization points. The metrics remain stable as the number of observation points is varied for three comparison scenarios: two synthetic GPs (left), a GP vs. a regular grid dataset (center), and a GP vs. an irregular mesh dataset (right).

C.3 Connection and Difference to Prior Works

Functional Generative Models vs. Surrogate PDE Solvers

While both functional generative models and surrogate PDE solvers utilize neural operators as their backbone architecture, the demands placed on the operator in these two settings are fundamentally different. A surrogate PDE solver typically learns a one-step mapping between function states, such as mapping initial conditions to a final solution. In contrast, a functional diffusion or flow model must learn to parameterize a continuous transformation between two probability measures, and also requires many steps evaluation during inference. This contrast is further highlighted by the signal frequencies involved: surrogate solvers often map between smooth, low-frequency functions (Li et al., 2023), while functional generative models must transform high-frequency noise into structured data samples. Consequently, designing a neural operator for generative tasks is significantly more demanding, as the learning task itself is more challenging.

Architectural Comparison to Other Neural Operators

The MINO architecture builds upon insights from previous work, including GINO and UPT (Li et al., 2023; Alkin et al., 2024), but introduces several key modifications to address their limitations in the context of functional generative tasks.

Comparison with GINO. GINO adopts a GNO encoder and decoder with an FNO processor. However, since a GNO acts as a low-pass filter (Laplacian smoothing) (Li et al., 2018), which creates an information bottleneck for decoding. The GNO decoder receives only the low-frequency output from the FNO processor, making it structurally difficult to reconstruct the high-frequency details essential

for high-fidelity generative modeling, which is supported by our experiments (see Appendix C.5).

Comparison with UPT. The UPT model uses a GNN for super-node pooling, a choice we identify as suboptimal for operator learning for three primary reasons. First, as the input discretization becomes finer, the GNN converges to a pointwise operator, which decouples it from the discretization convergence property required of a neural operator. Second, GNN can also be viewed as a low pass filter in the generative tasks, which has the same issue of GNO. Third, UPT relies on random subsampling to define super-nodes is ill-suited for adaptive meshes common in PDE tasks, where important regions with low grid density may be undersampled. For instance, on a dense mesh of the Earth, random subsampling would likely over-concentrate super-nodes in the polar regions that of less interest. Furthermore, UPT’s decoder uses a single-layer, perceiver-style (Jaegle et al., 2021) cross-attention where the queries are derived only from the output positions, which struggles to reconstruct the high-frequency information (Dong et al., 2021). We found that for generative tasks, it is crucial to use a combination of the input function and its position as the query to the cross attention decoder and to use multiple attention layers as shown in the ablation study (Appendix C.5). In Table C.3, we present a detailed architectural comparison between MINO and GINO/UPT.

Comparison with Transolver. Key differences lie in the receptive field and latent-space design of MINO. Transolver employs a purely global receptive field, whereas our architecture integrates both local and global information processing. In addition, MINO follows the modern, highly scalable, and efficient DiT paradigm by replacing the grid-dependent “patchify” operation with a domain-agnostic GNO encoder. Finally, because the slicing and de-slicing operations in Transolver act on every point, and these operations happen in every Transolver layer, which can be inefficient. In contrast, MINO performs encoding and decoding only once and can be viewed as a reduced-order operator; its decoder is shallow, consisting of one cross-attention block followed by a single self-attention block. As shown in our experiments, MINO is markedly more efficient than Transolver in both training and inference.

Designing neural operators with general deep learning architectures

The composition of multiple neural operators remains a valid neural operator (Wang et al., 2024a). Previous works such as GINO and UPT adopt the standard en-

coder–processor–decoder paradigm. However, directly replacing the processor with a generic neural network is sub-optimal: it forces the decoder to rely solely on a fixed-size latent representation, whereas the desired output lives in an infinite-dimensional function space. To overcome this limitation, we design the decoder via cross attention to take both the latent representation and the original input function as inputs, thereby circumventing the bottleneck imposed by a fixed latent shape.

C.4 Additional Results

In this part, we present additional qualitative results to further demonstrate the performance of MINO compared to baseline models. As shown in Figure C.2, MINO generates realistic samples for the Navier-Stokes benchmark that closely match the ground truth. In contrast, both GINO and UPT capture only the general, low-frequency trends of the solution, failing to reconstruct the high-frequency information essential for high-fidelity generation. This difficulty arises because generative models must learn a velocity field with correct high-frequency components to successfully transform high-frequency Gaussian noise into structured data samples. We provide further qualitative results for MINO on the Darcy Flow, Shallow Water, and Mesh-GP benchmarks in Figures C.3, C.4, and C.5, respectively, which visually confirm the effectiveness of our approach.

C.5 Ablation Study

We conduct an ablation study to validate the effectiveness of the specific cross-attention mechanisms used in the encoder and decoder of our MINO architecture. We take the standard MINO-T as our baseline and evaluate its performance against two architectural variants:

Self-Attention Encoder: We replace the cross-attention mechanism in the encoder with a standard self-attention mechanism. In this setup, the Key-Value (KV) pair for each attention block is derived from the output of the previous block, rather than being fixed as the initial latent representation from the GNO.

Position-Only Decoder: We modify the decoder’s query to be derived only from the function’s observation positions, similar to the decoder design in prior works like UPT, GINO. This removes the explicit dependency on the input function’s values in the query.

The results of this study, presented in Table C.7, confirm that our proposed design choices are critical for achieving high performance. The standard MINO-T con-

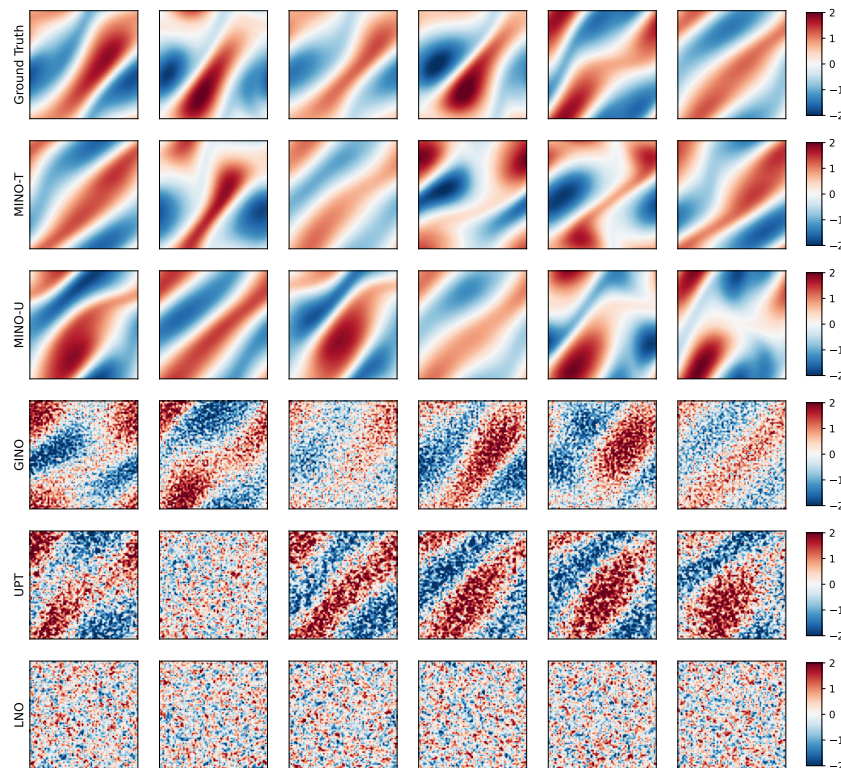


Figure C.2: Generation of Navier-Stokes samples under different baselines

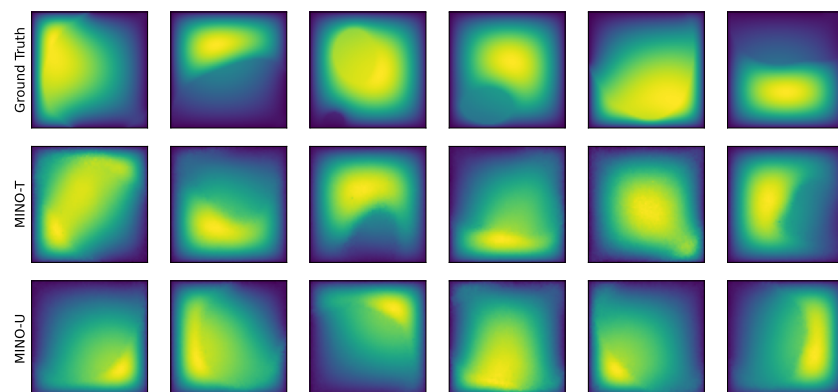


Figure C.3: Generation of Darcy-Flow samples

sistently and significantly outperforms both ablated variants across all datasets and metrics. The performance degradation is most severe in the *Position-Only Decoder* variant. By relying only on positional queries, the decoder struggles to reconstruct fine-grained details. This result highlights the importance of conditioning the decoder’s queries on the input function’s values for high-fidelity generative tasks. The *Self-Attention Encoder* variant also performs worse than our standard model, which suggests that using the initial latent representation as a fixed context for iterative refinement in the encoder is a more effective strategy than a standard self-attention

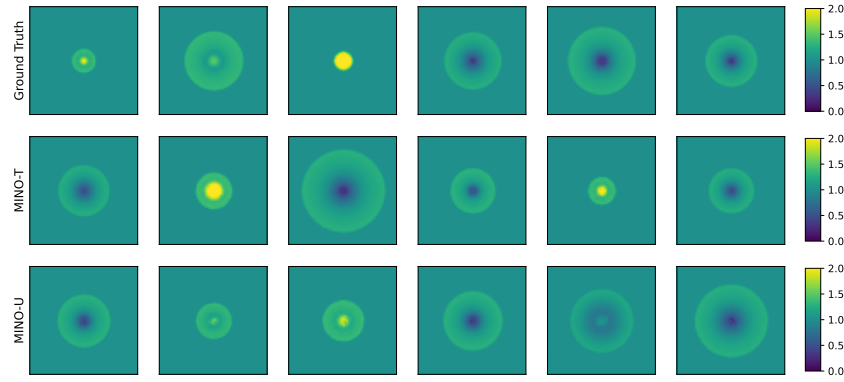


Figure C.4: Generation of Shallow Water samples

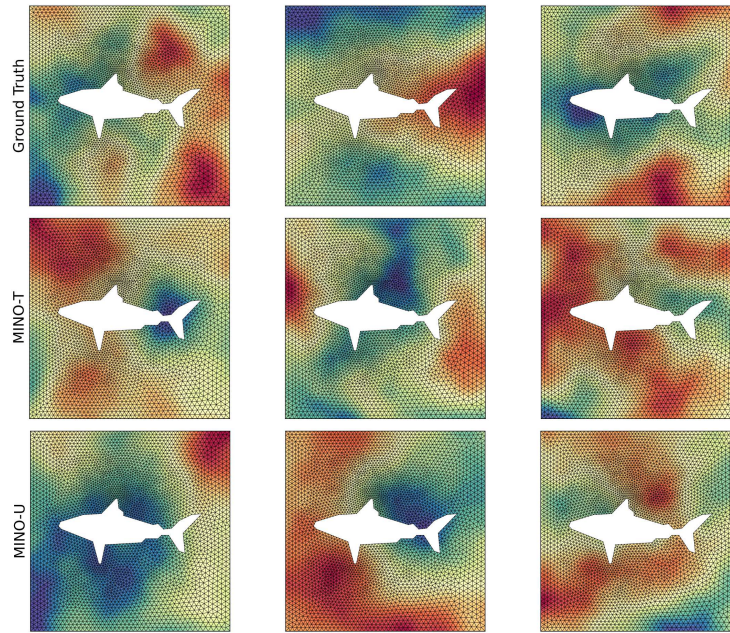


Figure C.5: Generation of Mesh GP samples

approach.

C.6 Empirical Analysis of Scaling and Stability

To evaluate performance consistency, we train both MINO-U and MINO-T with eight different random seeds on the Navier-Stokes and Cylinder-Flow datasets. The results, summarized in Table C.5, show that MINO-U consistently outperforms MINO-T across both datasets and metrics. We also note that performance variance across seeds is slightly higher on the Navier-Stokes task compared to the Cylinder-Flow task.

We next study data-scaling on Navier–Stokes benchmark by training on progres-

sively larger subsets of the full 30 k-sample dataset and evaluating on the same test split. Table C.6 shows steady performance gains as the training set grows, with improvements tapering off once roughly 18 k samples are reached.

C.7 Comparison with Transolver under Identical Settings

To ensure a fair comparison with Transolver, we reduce the latent dimension of MINO from 256 to 192, and use the same training configuration (200 epochs, batch size 48). This results in smaller models—MINO-U (S) (13.5M parameters) and MINO-T (S) (12.1M)—compared to Transolver (15.0M). Notably, our models are significantly faster, achieving a 3.1x speedup in training and a 3.5x speedup in inference while maintaining superior performances.

Table C.3: Architectural Comparison: MINO vs. GINO / UPT

Feature	MINO (Mesh-Informed Neural Operator)	GINO / UPT Frameworks
Primary Goal	Probabilistic Functional Generative Modeling (learning distributions via flow matching).	Deterministic Physics Simulation : GINO —solving PDEs on varying geometries; UPT —scalable temporal simulation (latent rollouts).
Encoder Strategy	Learned Resampling onto a Grid : a Graph Neural Operator (GNO) projects from N_{in} irregular points to a fixed, regular N_{node} grid.	GINO : GNO maps points to a regular grid and augments them with geometric features (signed-distance function). UPT : a GNN pools information from N_{in} points onto a smaller, fixed set of S “super-nodes.”
Latent Representation	Structured Feature Map (Grid) : a tensor of shape $(H \times W \times C)$, compatible to CNNs.	GINO : structured feature map (grid). UPT : unstructured set of vectors (tokens) of shape $(N_{\text{tokens}} \times D_{\text{hidden}})$.
Core Processor	SOTA diffusion-style U-Net (or Transformer).	GINO : Fourier Neural Operator (FNO) for efficient global processing on the latent grid. UPT : Transformer (Approximator) for modeling temporal transitions between latent tokens.
Decoder Mechanism	Cross-Attention with Input Bypass : Query (Q) comes from the original input function f_t and its positions; Key/Value (KV) come from the processed latent state. Prevents an information bottleneck and preserves high-frequency details.	No Input Bypass (Information Bottleneck) : GINO projects directly from latent grid to query points. UPT uses Perceiver-style cross-attention where Q is derived from <i>output positions only</i> . All information must be reconstructed solely from the compressed latent state.
Uncertainty	Inherent & Computationally Cheap : being generative, the model learns a distribution; Monte-Carlo dropout can provide epistemic uncertainty with a single trained network.	Computationally Expensive : uncertainty is not a primary target; estimating it usually requires training an ensemble of models (multiple runs with different seeds).

Table C.4: Ablation study on key architectural components of MINO across four benchmarks. We compare our standard MINO-T against two variants: one replacing the encoder’s cross-attention with self-attention, and another using a simpler position-only decoder query similar to prior work like UPT. Lower SWD/MMD scores are better; best results are in bold

Dataset →	Navier Stokes		Shallow Water		Darcy Flow		Cylinder Flow	
Variants ↓ Metric →	SWD	MMD	SWD	MMD	SWD	MMD	SWD	MMD
MINO-T	$4.0 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	$9.8 \cdot 10^{-3}$	$8.7 \cdot 10^{-3}$	$8.9 \cdot 10^{-2}$	$3.4 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
Self-Atten. Enc.	$5.2 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$2.0 \cdot 10^{-1}$	$7.6 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$
Position-Only Dec.	$5.7 \cdot 10^{-1}$	$4.5 \cdot 10^{-1}$	$8.4 \cdot 10^{-1}$	$5.7 \cdot 10^{-1}$	$8.3 \cdot 10^{-1}$	$4.4 \cdot 10^{-1}$	$7.2 \cdot 10^{-1}$	$5.3 \cdot 10^{-1}$

Table C.5: Performance comparison over eight random seeds. We report the mean $\pm$ standard deviation in parentheses for each metric. Lower is better.

Dataset →	Navier-Stokes		Cylinder Flow	
Model ↓ Metric →	SWD	MMD	SWD	MMD
MINO-T	$3.7 \cdot 10^{-2} (6.8 \cdot 10^{-3})$	$3.1 \cdot 10^{-2} (6.0 \cdot 10^{-3})$	$2.9 \cdot 10^{-2} (2.4 \cdot 10^{-3})$	$2.6 \cdot 10^{-2} (3.0 \cdot 10^{-3})$
MINO-U	$3.3 \cdot 10^{-2} (6.3 \cdot 10^{-3})$	$2.7 \cdot 10^{-2} (7.5 \cdot 10^{-3})$	$2.1 \cdot 10^{-2} (2.0 \cdot 10^{-3})$	$1.9 \cdot 10^{-2} (2.8 \cdot 10^{-3})$

Table C.6: Performance of MINO-U when trained on a subset of the Navier-Stokes dataset.

Metrics ↓ Samples →	6,000	12,000	18,000	24,000	30,000
SWD	$4.3 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$
MMD	$4.0 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$

Table C.7: Performance comparison with Transolver under identical settings. Best Performance in bold

Dataset →	Navier Stokes		Mesh GP		Cylinder Flow	
Model ↓ Metric →	SWD	MMD	SWD	MMD	SWD	MMD
Transolver	$5.3 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$9.1 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$
MINO-T (S)	$4.4 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$7.9 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
MINO-U (S)	$2.9 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$6.1 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$

APPENDIX FOR CHAPTER 5

D.1 Details of simulation dataset preparation

Computational testbed. Selecting an appropriate computational testbed is critical to ensuring the ability of the AI models to capture potential complexities of realistic wave propagation. It is thus emphasized that the simulation domain must exhibit heterogeneous geological features such as shallow sedimentary basins and sharp contrasts between different geologic units that result in abrupt changes in seismic wave velocity propagation and strongly influence spatial amplification patterns. The San Francisco Bay Area (SFBA) satisfies these criteria (Fig D.1), and additionally is a densely populated metropolitan region with significant infrastructure development and seismic activity resulting in a potential of high risk, thus has been significantly studied. For this study, we adopted the United States Geological Survey (USGS) SFBA community velocity model (version 21.1) Aagaard et al., 2021. This model was selected because it has undergone rigorous performance evaluation through high-resolution 3D physics-based simulations and recorded seismicity. A recent validation study Pinilla-Ramos et al., 2025 in the SFBA involved small-magnitude earthquakes and demonstrated that the velocity model performs particularly well in the frequency range of 0.25–3 Hz with practically unbiased simulated motions compared to recorded ground motion data and slight over-prediction of 25% in the frequency range 3-5Hz, successfully capturing wave propagation effects in near-surface soft materials and local basins. Moreover, using the same testbed, a significant simulation effort of large-magnitude Hayward-fault earthquake events has been recently conducted as a collaboration between the Pacific Earthquake Engineering Research (PEER) Center and Lawrence Berkeley National Laboratory (LBNL) McCallen et al., 2025. This work leverages the EQSIM framework and Exascale computing for earthquake simulation capabilities McCallen et al., 2024; McCallen et al., 2021a, and produced an open-access Simulated Ground Motion Database (SGMD) of surface motions in the SFBA.

Physics-based earthquake simulations. The physics-based ground motion simulations in this study are generated using the SW4 (Seismic Waves, 4th order) software. SW4 numerically solves the elastodynamic wave equations in displace-

ment formulation using a node-based finite difference approach, guaranteeing energy stability and achieving fourth-order accuracy in both space and time. The primary objective of these simulations is to capture the deterministic physics of wave propagation, including the effects of complex 3D geologic structures such as sedimentary basins, which strongly influence ground motion amplitude and duration. While the simulations provide a rigorous representation of linear wave propagation, they are subject to limitations. First, the modeling assumes a linear-elastic response of the geologic materials, neglecting potential nonlinear soil behavior. Second, simulated high-frequency content is limited by epistemic uncertainty of subsurface conditions, i.e., the lack of sufficient data on small-scale soil properties (on the order of tens of meters) to construct a velocity model capable of supporting accurate wave propagation at frequencies above 3-5 Hz. Simultaneously, a minimum shear-wave velocity cutoff (e.g., 250 m/s adopted here) is enforced because lower velocities correspond to shorter wavelengths, necessitating a finer computational grid that would drastically increase computational demands. This constraint is critical because computational cost scales inversely with the fourth power of the grid spacing (which is driven by the ratio of frequency to minimum shear-wave velocity), making high-resolution deterministic simulations computationally intensive. Despite these limitations, physics-based simulations provide high-fidelity data for medium- to long-period wave propagation. For the dataset generation in this study, the simulations are designed to resolve frequencies up to 1 Hz aiming to achieve manageable computational effort and robust simulated datasets of verified validity for the physical domain of interest. The following subsections describe the adopted computational domain, the seismic source characterization for small- and large-magnitude earthquake simulations and the simulation strategy for efficiently generating the ground motion datasets.

Computational domain and velocity model. The computational domain is shown in Fig D.1 and is identical to the one used in Pinilla-Ramos et al., 2025 which is a sub-domain of the detailed USGS v21.1 SFBA V Aagaard et al., 2021. The main modelling assumptions are listed in Figure 2, including a sketch describing the grid size and the implemented grid refinements in depth. A minimum shear-wave velocity of 250 m/s is adopted, and combined with a required accuracy for frequencies up to 1 Hz results in a minimum grid spacing of 31.25 m to ensure proper numerical accuracy ensuring 8 points per wavelength in all points within the mesh. The wave attenuation is modeled through a linear visco-elastic model characterized by the quality factors (Q_p and Q_s for compressional and shear waves,

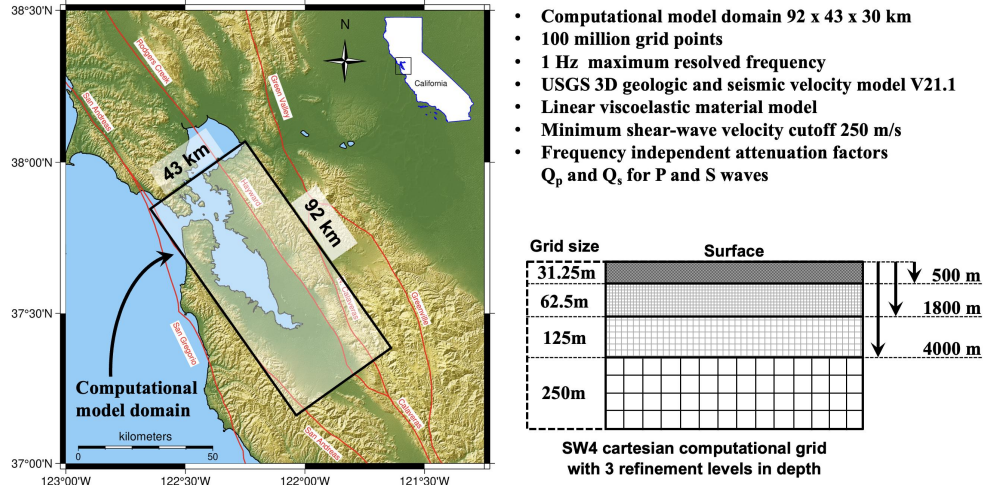


Figure D.1: Computational domain adopted for the SFBA and model parameters

respectively), which were obtained directly from the characterization included in the velocity model by the USGS.

The SFBA region features complex geologic structures that significantly influence wave propagation. In the northern area, the Hayward fault separates two distinct geological blocks Pinilla-Ramos et al., 2025 (cross-sections A-A, B-B, and C-C in Fig D.2). The western block consists of shallow soft marine sediments overlying sedimentary basins that extend down to 1 km depth (cross-sections B-B and C-C). In contrast, the eastern block consists mainly of sedimentary rocks and tends to be stiffer at the surface (with shear-wave velocities usually above 500 m/s). This comparison reverses at depths larger than 1 km where east of the Hayward fault consists of softer rocks compared to the western block. Structural complexity increases to the south where the Calaveras fault bounds the eastern block (cross-sections C-C, D-D, and E-E). Notably, the San Jose and Santa Clara regions in the south of the SFBA (cross-section E-E) exhibit deep, irregular sedimentary basins, which are expected to generate significant wave reverberations and increase ground motion duration.

Seismic source characterization The simulation of small-magnitude earthquake events (i.e., magnitudes less than 5) are considered following a double-couple point source using the Liu source-time function Liu et al., 2006 to model how the source releases energy over time. Being consistent with rupture dynamics, this slip rate function is not symmetric (as shown in Fig D.3; it is characterized by a large initial peak and a gradual amplitude decay that represents the healing process of the rupture. To simulate large-magnitude earthquake events, a planar fault rupture is modelled and initiates (at the hypocenter location) evolving with time during the

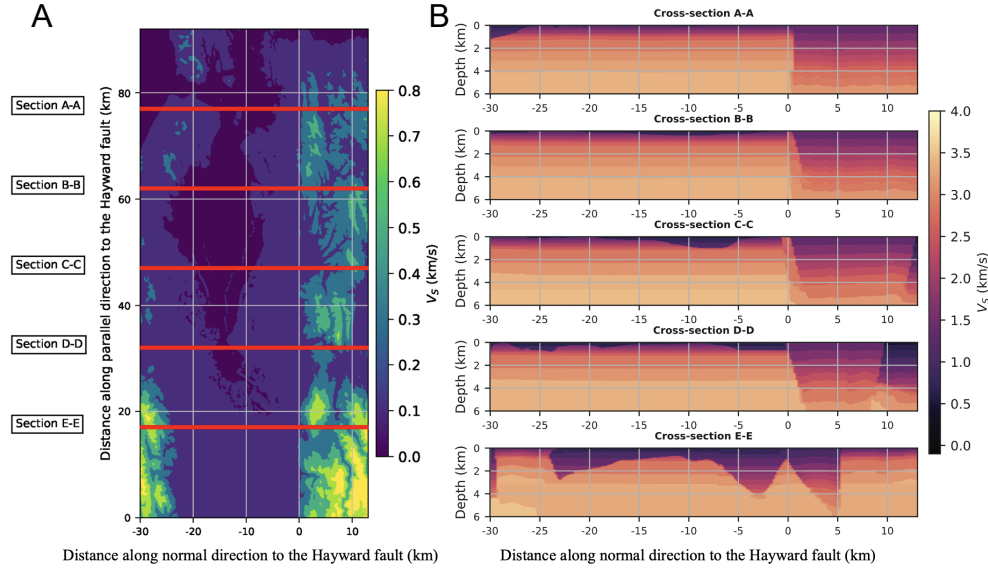


Figure D.2: (a) Map view of the free surface shear-wave velocity in the simulation domain, extracted from the United States Geological Survey velocity model, with the E-W red lines indicating the location of the velocity model cross-sections shown in (b); (b) Vertical cross-sections of the velocity model with the Hayward fault at 0 normal distance. (adopted from Pinilla-Ramos et al. 2025 with permission)

earthquake event. The rupture process for a given scenario is controlled by numerous seismological properties including the total slip, earthquake realizations utilized in the SFBA simulations are generated using the Graves and Pitarka (GP) kinematic rupture model Graves et al., 2016, with an enhanced slip rate representation Pitarka et al., 2022. The shape of the source time function varies with depth with a linear transition between crustal depths of 1 and 3km. This formulation allows the shape of the source time function to transition from “cosine-type,” with a relatively modest high-frequency content in the depth interval 0–1km, to “Kostrov-type” with a broadband frequency content, at depths greater than 3km.

Simulation database generation The database contains 2,100 small-magnitude earthquake events, with 2,000 intended for training of the AI algorithms and 100 for testing. All simulations have the same focal mechanism of dip=80°, strike=145° and rake=180° which has been identified as the one in the 2018 Berkeley 4.4-magnitude earthquake event reported in (Pinilla-Ramos et al., 2025), as well as the Liu source function and corner frequency of $f_c=1.4\text{Hz}$ suggested in the same study. The locations of the small-magnitude earthquake simulations are randomly positioned throughout the domain, covering its entire width, length and depth using the Latin Hypercube Sampling technique to ensure uniform spatial distribution. The simulations are performed on Perlmutter, a 70-PF (Peta flops) Graphics Process-

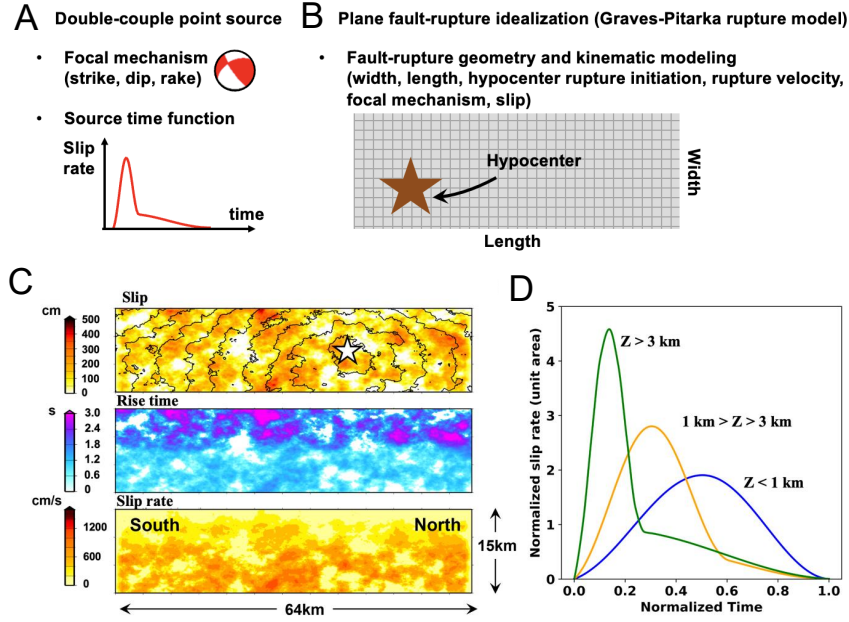


Figure D.3: (a) Small-magnitude seismic source characterization as a double-couple point-source function defined by its focal mechanism and source time function; (b) Large-magnitude seismic source characterization following the Graves-Pitarka rupture model; (c) Example of a pure stochastic rupture realization for the Hayward fault plane, shown through its total slip, rise time and slip rate ; (d) slip rate of points at various depths within the fault-plane as a function of time, normalized to show a unit area of total slip, developed within a normalized rise time of 1 second.

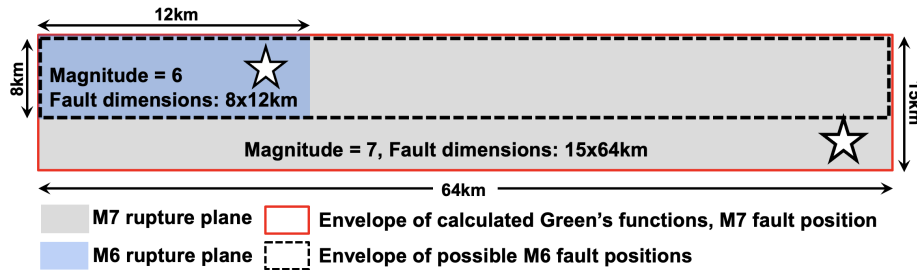


Figure D.4: Envelope of calculated Green's functions, rupture plane dimensions and positions for Magnitude 7 and Magnitude 6 earthquake events

ing Unit (GPU)-accelerated super-computer at the National Energy Research Super Computer Center at Lawrence Berkeley National Laboratory, with a total of 600 node-hours of GPU resources, where each node contains 4 NVIDIA A100 GPUs. Moreover, the database contains large-magnitude earthquake simulation, in particular 1,100 M7 and 2,100 M6 earthquake realizations, out of which 100 M7 and 100 M6 realizations are reserved for testing and the rest are reserved for training. To

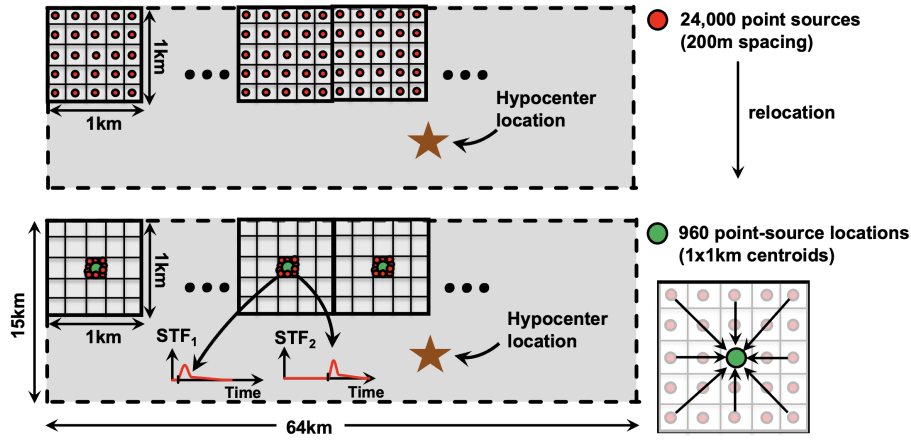


Figure D.5: Fault-plane discretization with point-sources at 200-m grid, repositioned in the corresponding centroids of 1x1km sub-faults

generate the source ruptures for each of those realizations, the Graves-Pitarka kinematic rupture model is deployed (Graves et al., 2016). The rupture plane dimensions are shown in Figure 6, and are governed by the magnitude of the seismic event, in particular 15x64 km for the M7 and 8x12km for the M6.

The straightforward approach in EQSIM (McCallen et al., 2021b) is to directly incorporate the extended source rupture files into sw4 and simulate the earthquake response. However, this could become cumbersome for generating thousands of realizations, and thus a different approach is investigated here to ensure low computational cost as well as scalability (i.e., the ability to generate more realizations if needed without the associated scaling of the computational cost). To this end, the representation theorem is deployed, allowing to use the Green's functions describing the source-to-site response (i.e., from each point of the finite source rupture to the surface stations). In Fig D.4, the envelope of the plane in which the Green's functions are calculated is shown in red, which corresponds to the M7 rupture plane shown as a gray-shaded area. The M6 rupture plane is shown as a blue-shaded area, and given its smaller dimensions compared to the M7 rupture plane, it can be positioned anywhere within the black-dashed envelope (i.e., as a sliding stencil at the top of the envelope where the Green's functions are calculated). For the M6 and M7 modelled magnitudes, the rupture variable parameters in these realizations are: 1) the hypocenter locations within the fault (shown in star markers in Fig D.4); 2) the slip distribution controlled by a random seed generator within the Graves-Pitarka kinematic rupture generator; and 3) the average rupture velocity. In particular, the

hypocenter horizontal location (HH) is $-25 \text{ km} < \text{HH} < 25 \text{ km}$ measured from the center of the M7 fault plane and $-3 \text{ km} < \text{HH} < 3 \text{ km}$ measured from the center of the M6 fault plane. The hypocenter depth (HD) is $7 \text{ km} < \text{HD} < 9 \text{ km}$ measured from the top of the M7 fault plane and $3 \text{ km} < \text{HD} < 6 \text{ km}$ measured from the top of the M6 fault plane. The rupture velocity (VRup) is $0.65 < \text{VRup} < 0.95$, expressed as a percentage of the average shear-wave velocity, similarly for both M7 and M6 ruptures.

In Fig D.5, the first row shows the 200-meter-spacing discretization of the rupture plane, as required for a proper rupture characterization of frequencies up to 1Hz, resulting in 24,000 point-sources. To reduce the volume of Green’s functions calculations, these point sources are then grouped into 5x5km sub-faults, and artificially considered to be located in their respective centroid that is shown in green at the second row of Figure 7. It is noted that during this modification, the respective source-time functions are preserved at the original resolution (24,000) to maintain the rupture model’s fidelity, and only the calculated Green’s functions are reduced to 960 (i.e., by a factor of 25). This allows for a significant reduction in computational demand – i.e., calculating 960 as opposed to 24,000 Green’s functions – requiring a total of 400 node-hours of GPU nodes on Perlmutter. After this dataset of Green’s functions is calculated, each realization defined by its source-rupture is calculated by convolving the 24,000 source time functions with their respective centroids, which additionally requires 1 node-hour of CPU-only nodes on Perlmutter, resulting in a total of 3,100 CPU-only node-hours, where each CPU-node contains two AMD EPYC 7763 (Milan) CPUs.

Preprocessing of the simulation data. To simplify the learning setting and alleviate artifacts induced by absorbing boundaries, we crop the simulations to an 80×40 km subdomain sharing the same geometric center as the original computational domain (92×43). We further apply a sixth-order low-pass filter with cutoff frequency 1 Hz, downsample the spatial discretization from $[320, 160]$ to $[256, 128]$, and truncate the time series to 24 seconds sampled at (original) 4 Hz. The resulting three-component velocity wavefields therefore have shape $[3, 256, 128, 96]$. To alleviate dynamic-range variation across events of different magnitudes, each sample is divided by its sample-wise standard deviation, and the corresponding $\log_{10}$ standard deviation is appended as an additional constant channel. Consequently, each event is represented as an array of shape $[4, 256, 128, 96]$. The unnormalized three-channel representation and the normalized four-channel representation are fully equivalent,

since either can be recovered from the other.

D.2 Details of experimental setup

This section describes the experimental configuration and training protocol. All experiments were conducted on a single NVIDIA RTX A6000 Ada GPU (48 GB). All three components, namely the super-resolution neural operator (SNO), the autoencoding neural operator (AENO), and the conditional latent flow matching model, were trained on a total of 5,000 samples. Among them, only the latent flow matching model is stochastic and conditioned on event (magnitude, hypocenter), whereas SNO and AENO are deterministic operators.

Super-resolution neural operator (SNO). The SNO maps a low-resolution wavefield to a high-resolution wavefield. During training, the low-resolution input has shape $[4, 128, 96, 48]$ and the high-resolution target has shape $[4, 256, 128, 96]$. The fourth channel is a normalization channel that stores the per-sample standard deviation on a $\log_{10}$ scale. We implement SNO using a residual Fourier neural operator (rFNO) with 4 FNO layers. The model has 871 million parameters and a storage. We train it for 100 epochs with batch size 2 using AdamW and a cosine-annealing learning-rate schedule with an initial learning rate of 5×10^{-4} . The total training time is 58.5 hours.

Autoencoding neural operator (AENO). The AENO consists of an encoder operator and a decoder operator, each implemented as an rFNO with 6 layers. The model has 663 million parameters. We train it for 200 epochs with batch size 8 using AdamW and a cosine-annealing learning-rate schedule with an initial learning rate of 5×10^{-4} . The total training time is 36.3 hours.

Conditional latent flow matching. For conditional latent flow matching, we adopt a diffusion U-Net architecture following Tong et al., 2024. The model has 128 million parameters. We train it for 300 epochs with batch size 64 using AdamW and a cosine-annealing learning-rate schedule with an initial learning rate of 2×10^{-4} . The total training time is 5.8 hours.

D.3 Additional Results

Random reference stations for NCC. We provide additional NCC analyses using randomly selected reference stations for the representative Mw 4.4 point-source scenario shown in Fig. 5.3. These supplementary examples are intended to verify that the coherence patterns reported in the main text are not specific to a particular

Table D.1: Hyperparameters and training details of the SNO, AENO, and Flow Matching models.

Components	SNO	AENO	Flow Matching
Architecture	rFNO	rFNO	Diffusion U-Net
Layers	4	6+6	-
Batch Size	2	8	64
Epochs	100	200	300
Learning Rate	5×10^{-4}	5×10^{-4}	2×10^{-4}
Parameters	871 M	663 M	128 M
Learning rate	5e-4	5e-4	2e-4
Scheduler	Cosine Annealing	Cosine Annealing	Cosine Annealing
Training Time	58.5 h	36.3 h	5.8 h

reference location. Across different choices of reference station, the synthetic wavefields reproduce similar spatial patterns of peak NCC and, more importantly, similar travel-time gradients in the time-lag fields relative to the ground truth. This further supports the conclusion that GMFlow preserves physically consistent phase alignment and wave-propagation kinematics over the full spatial domain.

Evaluation on the Mw 6.0 finite-rupture source scenario. We also include additional evaluation results for the Mw 6.0 finite-rupture scenario in Fig. D.7. This case serves as an intermediate regime between the comparatively simple Mw 4.4 point-source events and the more challenging Mw 7.0 extended-rupture events discussed in the main text. The supplementary results show that GMFlow captures the principal characteristics of finite-rupture ground motion, including spatially varying amplitude patterns, coherent phase delays, and rupture-driven wavefield asymmetry. These results indicate that the model remains effective not only at the two ends of the magnitude range, but also in the intermediate finite-rupture regime.

Zero-shot super-resolution on finer spatial grids. Finally, we present zero-shot super-resolution results in Fig. D.9. In this experiment, the synthetic wavefields are evaluated on a finer spatial grid of $[3, 512, 256, 96]$, whereas the reference simulations are available at the default resolution of $[3, 256, 128, 96]$. Because GMFlow is formulated as a functional generative model with a super-resolution neural operator, the learned representation is not tied to a single output discretization. As a result, the model can be queried on denser spatial grids without retraining, while still producing wavefields that remain visually coherent and physically plausible.

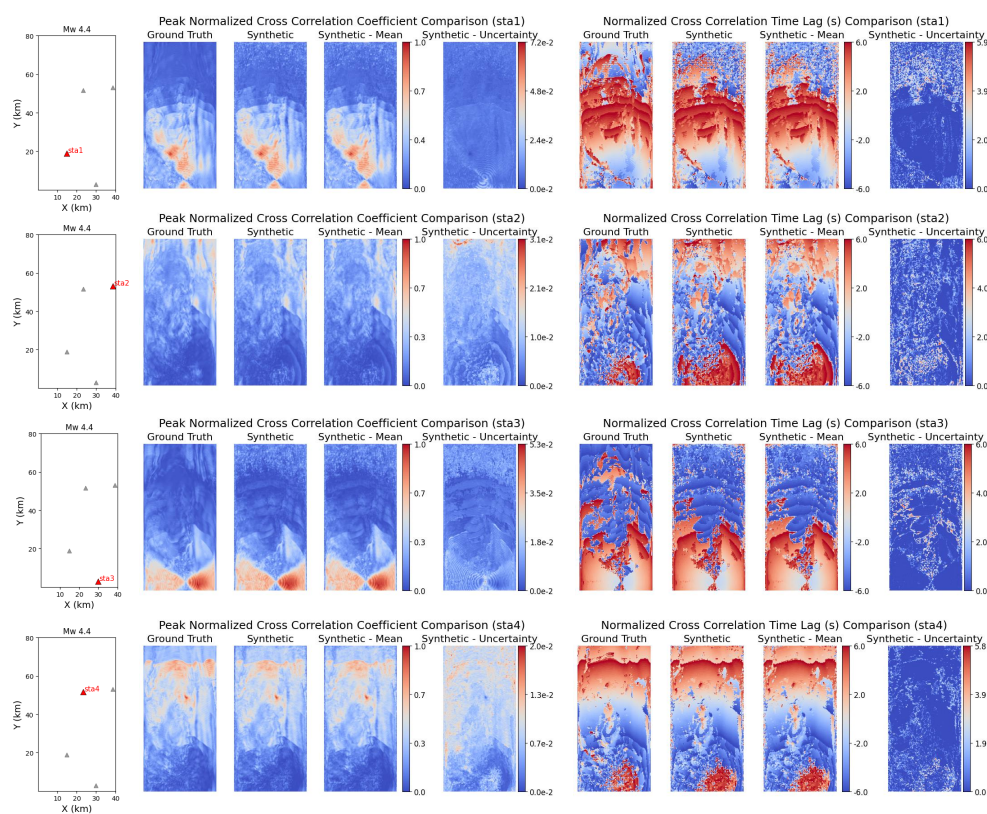


Figure D.6: NCC analysis using randomly selected reference stations for the representative M_w 4.4 event.

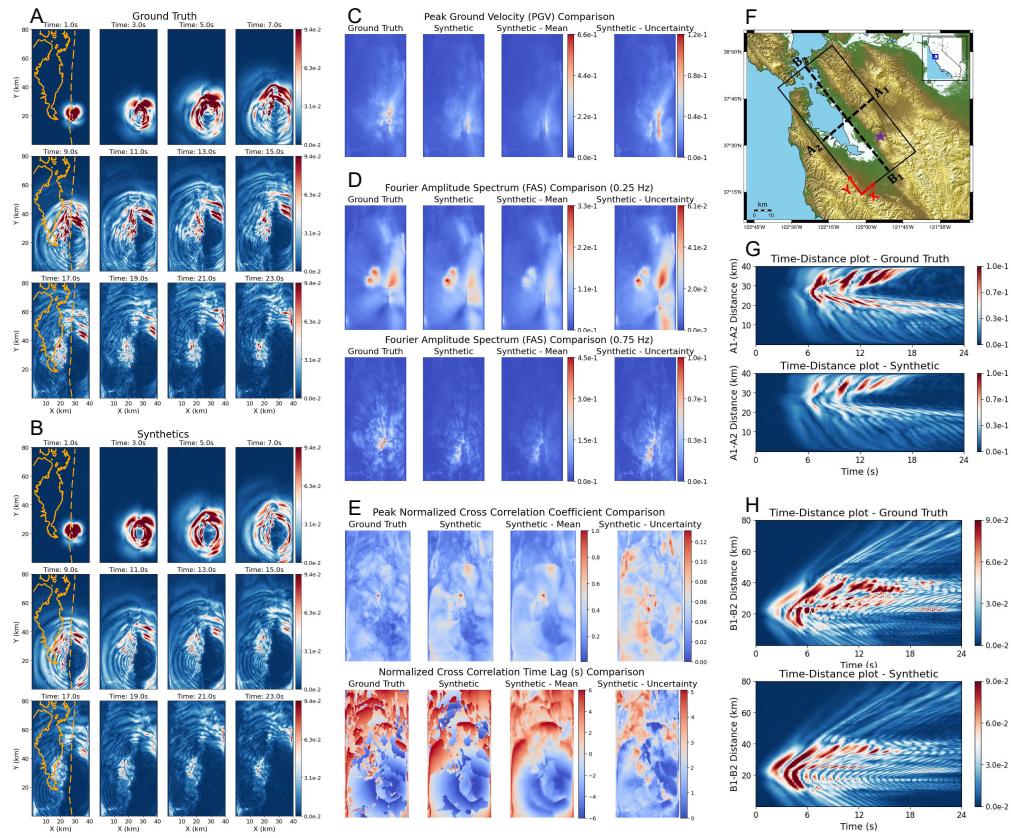


Figure D.7: Validation of GMFlow on a representative M_w 6.0 finite rupture source scenario.

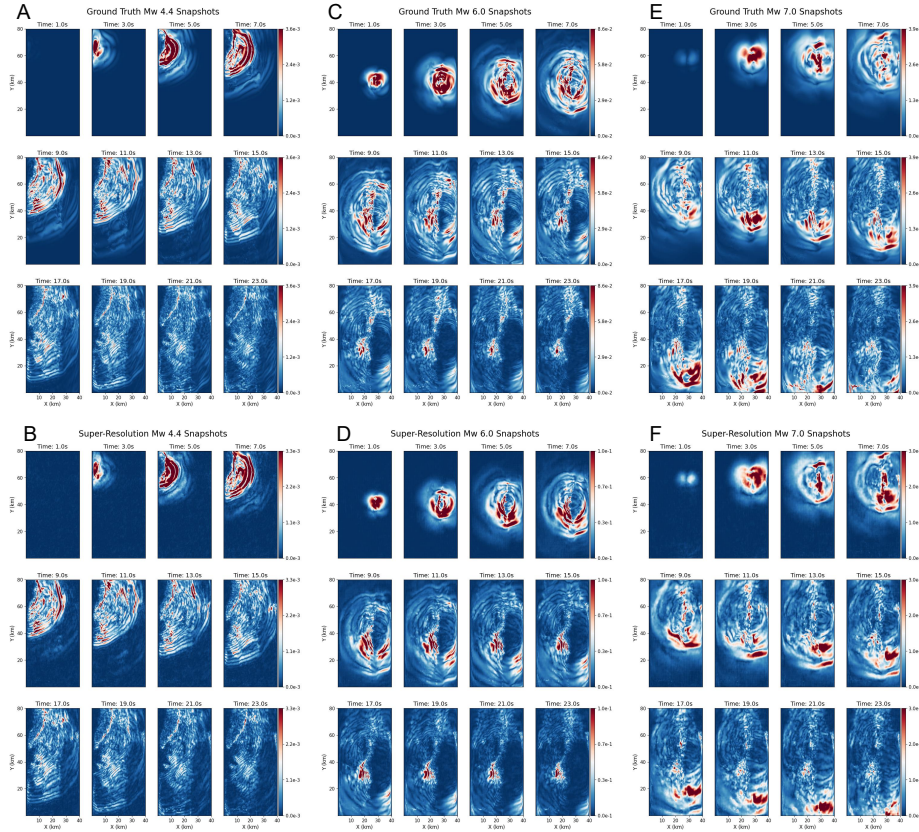


Figure D.8: **Zero-shot super-resolution.** Ground-truth wavefield snapshots are provided at spatial resolution 256×128 , whereas the synthetic samples are super-resolved to spatial resolution 512×256 . (A, C, E) Ground-truth snapshots for representative M_w 4.4, M_w 6, and M_w 7 events, respectively. (B, D, F) Corresponding zero-shot super-resolved synthetic snapshots for the same events.

D.4 Ablation Study

In this section, we perform an ablation study focusing on two factors: (i) the capacity of the latent flow matching model (16M vs. 128M parameters), and (ii) the cutoff frequency used in the *Physics-Aligned Subspace*.

Configuration of flow matching. We choose the modified 3D diffusion-UNet architecture as the backbone for the flow matching Tong et al., 2024. For the smaller 16M model, we use the following configuration:

```
model_config = {
    'hidden_channels': 64,
    'num_heads': 4,
    'attention_res': '8',
}
```

For the larger 128M model, we use:

```
model_config = {
    'hidden_channels': 96,
    'num_res_blocks': 2,
    'num_heads': 8,
    'attention_res': '8,4',
    'channel_mult': (1, 2, 3, 4)
}
```

Evaluation metrics. We reproduce both the statistical distribution analysis and the residual analysis. For the statistical distribution comparison, we use the 1D Wasserstein distance. Given two one-dimensional probability mass functions p_1 and p_2 , the first Wasserstein distance is defined as

$$l_1(p_1, p_2) = \inf_{\pi \in \Pi(p_1, p_2)} \int_{\mathbb{R} \times \mathbb{R}} |y_1 - y_2| d\pi(y_1, y_2), \quad (\text{D.1})$$

where $\Pi(p_1, p_2)$ denotes the set of all joint distributions on $\mathbb{R} \times \mathbb{R}$ whose marginals are p_1 and p_2 , respectively. We quantify distributional agreement by repeating the same procedure used in Fig. 5.6A–B. For the residual analysis, we compare the mean residual curves using the root mean squared error (RMSE) as the evaluation metric.

As shown in Table D.2, lower values indicate better agreement with the target statistics. Increasing model capacity consistently improves performance across nearly all metrics, with the clearest gains observed for the more challenging finite-rupture cases. Likewise, Table D.3 suggests that higher cutoff frequencies, particularly $f_c = 0.75$ and 1.0 Hz, are more effective for capturing large finite-rupture events. This trend is further corroborated by the full residual spectra for different f_c values shown in Fig. D.9.

Table D.2: Model-size ablation for latent flow matching. Quantitative comparison of the 16M and 128M CFM models using PGV, FAS-based Wasserstein distance at multiple frequencies, and RMSE of the mean residual for M_w 4.4, M_w 6.0, and M_w 7.0 events. The lower the better.

Datasets	Model ↓ Metric →	PGV	FAS-0.25Hz	FAS-0.5Hz	FAS-0.75Hz	FAS-0.96Hz	residual mean
Mw 4.4	CFM - 16M	$7.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$5.7 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-1}$
	CFM - 128M	$5.2 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$8.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$
Mw 6.0	CFM - 16M	$6.5 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$9.9 \cdot 10^{-2}$	$2.7 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$	$4.4 \cdot 10^{-1}$
	CFM - 128M	$3.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$
Mw 7.0	CFM - 16M	$1.4 \cdot 10^{-1}$	$7.6 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$3.8 \cdot 10^{-1}$	$4.7 \cdot 10^{-1}$	$6.4 \cdot 10^{-1}$
	CFM - 128M	$8.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$	$4.3 \cdot 10^{-1}$	$5.6 \cdot 10^{-1}$

Table D.3: Cutoff-frequency ablation for the Physics-Aligned Subspace. Quantitative comparison of models using different cutoff frequencies $f_c \in \{0.5, 0.6, 0.75, 1.0\}$ Hz across representative M_w 4.4, M_w 6.0, and M_w 7.0 events. Metrics include PGV, FAS-based Wasserstein distance at multiple frequencies, and RMSE of the mean residual. The best and second-best results in each column are highlighted in black and brown, respectively.

Datasets	Model ↓ Metric →	PGV	FAS-0.25Hz	FAS-0.5Hz	FAS-0.75Hz	FAS-0.96Hz	residual mean
Mw 4.4	fc=0.5hz	$5.7 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$9.6 \cdot 10^{-2}$	$5.9 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$
	fc=0.6hz	$6.2 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$7.5 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$
	fc=0.75hz	$5.2 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$8.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$
	fc=1.0hz	$5.9 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	$9.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$
Mw 6.0	fc=0.5hz	$5.7 \cdot 10^{-2}$	$5.7 \cdot 10^{-3}$	$8.8 \cdot 10^{-2}$	$2.7 \cdot 10^{-1}$	$1.8 \cdot 10^{-1}$	$3.5 \cdot 10^{-1}$
	fc=0.6hz	$7.6 \cdot 10^{-2}$	$7.8 \cdot 10^{-3}$	$1.2 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$	$4.1 \cdot 10^{-1}$
	fc=0.75hz	$3.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$
	fc=1.0hz	$5.0 \cdot 10^{-2}$	$9.9 \cdot 10^{-3}$	$7.7 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$3.0 \cdot 10^{-1}$
Mw 7.0	fc=0.5hz	$1.3 \cdot 10^{-1}$	$7.6 \cdot 10^{-2}$	$2.0 \cdot 10^{-1}$	$4.2 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$5.8 \cdot 10^{-1}$
	fc=0.6hz	$1.4 \cdot 10^{-1}$	$9.2 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$4.0 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$
	fc=0.75hz	$8.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$	$4.2 \cdot 10^{-1}$	$5.6 \cdot 10^{-1}$
	fc=1.0hz	$8.2 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$	$4.3 \cdot 10^{-1}$	$5.2 \cdot 10^{-1}$

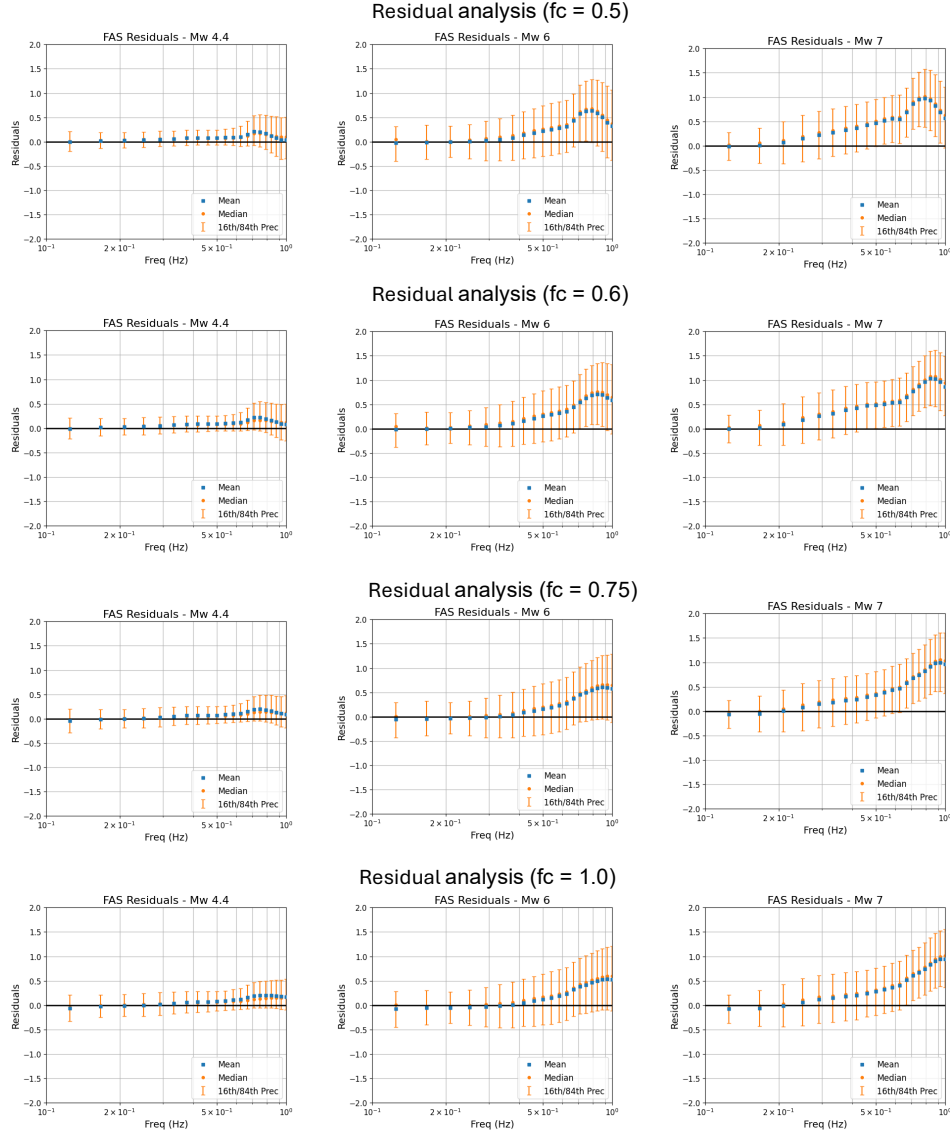


Figure D.9: Frequency-dependent spectral residuals for different cutoff frequency f_c

Spectral Bias Calibration

The systematic underestimation of high-frequency content for large finite-rupture events is likely due, at least in part, to limited training coverage in this regime. Our model is trained on only 5,000 samples, which appears sufficient for point-source events, whose wavefields are governed by comparatively simpler propagation effects. In contrast, large-magnitude finite ruptures exhibit richer and more heterogeneous high-frequency structure due to rupture complexity, including extended-source radiation and directivity, and accurately capturing this variability likely requires substantially more training samples.

To mitigate this limitation within the current framework, we introduce an optional post-processing bias correction based on the mean spectral residual shown in Fig. 5.7. Empirically, the residual is approximately spatially uniform, suggesting that a single frequency-dependent amplitude correction can be applied to each synthetic event. In practice, the calibration is defined in log-spectral space and applied equivalently as a frequency-dependent multiplicative scaling of the spectrum. This adjustment yields an approximately unbiased spectrum in expectation, as illustrated in Fig. D.13. Because the correction modifies amplitudes only, it preserves the predicted phase structure and spatiotemporal coherence while improving FAS agreement and magnitude scaling. Importantly, this correction is only a practical mitigation and does not replace the need for improved generative modeling of high-frequency rupture and scattering effects.

For the representative magnitudes M_w 4.4, 6.0, and 7.0, we estimate three corresponding bias functions from the mean spectral residual in log space. To correct a synthetic realization with magnitude between M_w 4.4 and 6.0, or between 6.0 and 7.0, we linearly interpolate between the two neighboring bias functions to obtain a magnitude-dependent correction curve. This interpolated bias function is then applied as a frequency-dependent multiplicative scaling to the synthetic spectrum.

Additional results for this calibration procedure are provided in this section. As shown in Table D.4, the calibrated synthetic samples substantially reduce spectral bias relative to the uncalibrated results. As further illustrated in Fig. D.10, the calibration does not alter the phase information, and the corrected synthetics remain both visually and physically plausible. Moreover, Fig. D.12 shows that, after calibration, the magnitude scaling is much better aligned with the ground truth. Overall, these results confirm that the proposed spectral calibration effectively reduces systematic amplitude bias while preserving the predicted phase structure and spatiotemporal coherence.

Legend for movie files

The movie files compare generated ground motions with the corresponding simulation data. **Movie 1**, **Movie 2**, and **Movie 3** show data-versus-synthetic comparisons for the M_w 4.4, M_w 6, and M_w 7 scenarios, respectively. **Movies 4**, **5**, and **6** show data-versus-spectrally calibrated synthetic comparisons for the M_w 4.4, M_w 6, and M_w 7 scenarios, respectively.

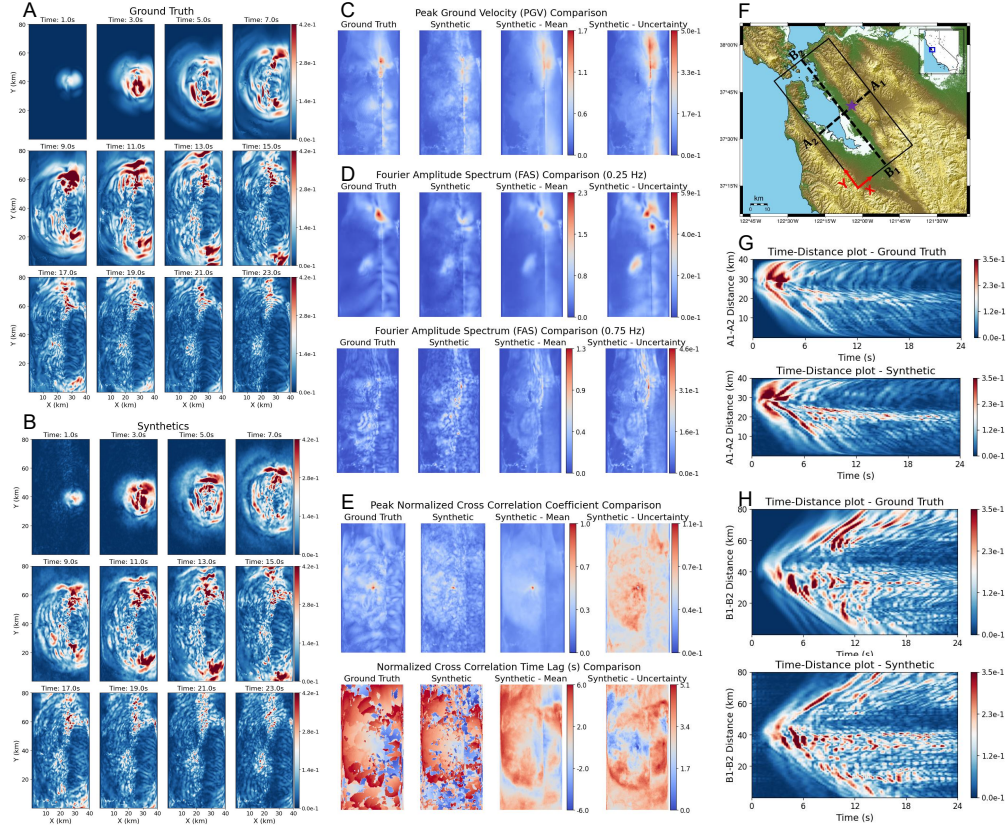


Figure D.10: Evaluation of spectral-calibrated GMFlow on a representative M_w 7.0 point-source scenario.

Table D.4: Effect of post-processing spectral calibration. Quantitative comparison of GMFlow and GMFlow-calibrated for representative M_w 4.4, M_w 6.0, and M_w 7.0 events using PGV, FAS-based Wasserstein distance at selected frequencies, and RMSE of the mean residual. The lower, the better.

Datasets	Model ↓ Metric →	PGV	FAS-0.25Hz	FAS-0.5Hz	FAS-0.75Hz	FAS-0.96Hz	residual mean
Mw 4.4	GMFlow-calibrated	$2.1 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$	$5.0 \cdot 10^{-4}$
	GMFlow	$5.2 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$8.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$
Mw 6.0	GMFlow-calibrated	$1.4 \cdot 10^{-2}$	$9.6 \cdot 10^{-3}$	$2.3 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$9.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-3}$
	GMFlow	$3.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$
Mw 7.0	GMFlow-calibrated	$1.8 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$7.8 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$
	GMFlow	$8.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$	$4.3 \cdot 10^{-1}$	$5.6 \cdot 10^{-1}$

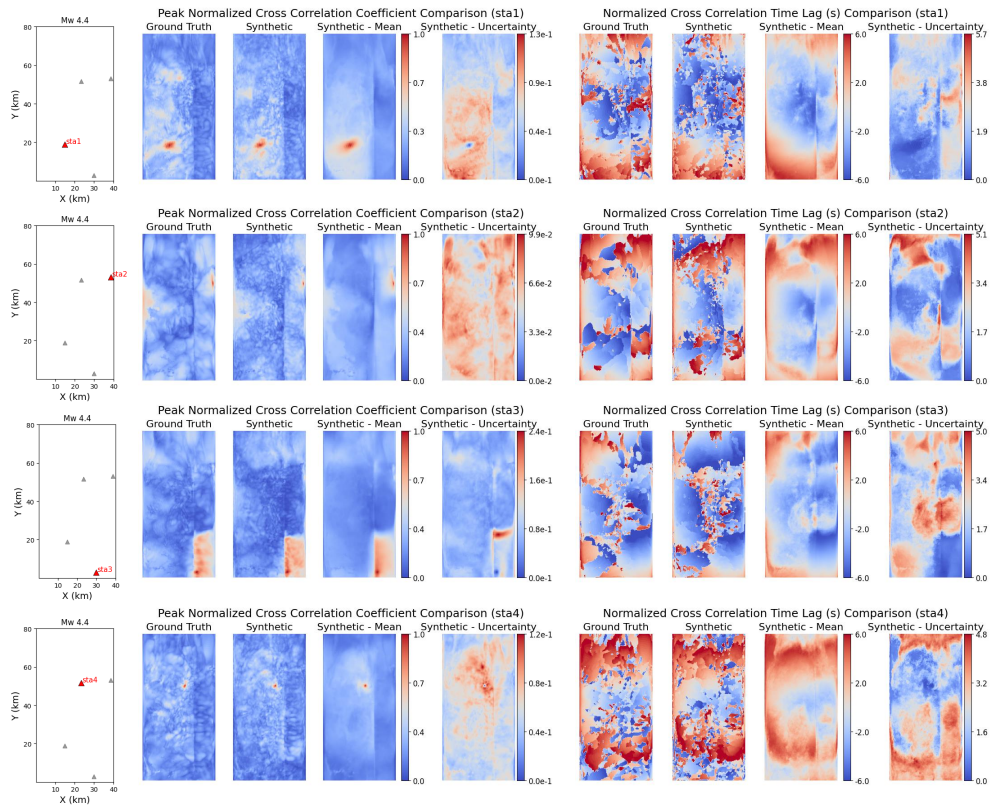


Figure D.11: NCC analysis using randomly selected reference stations for the spectral-calibrated representative M_w 7 event.

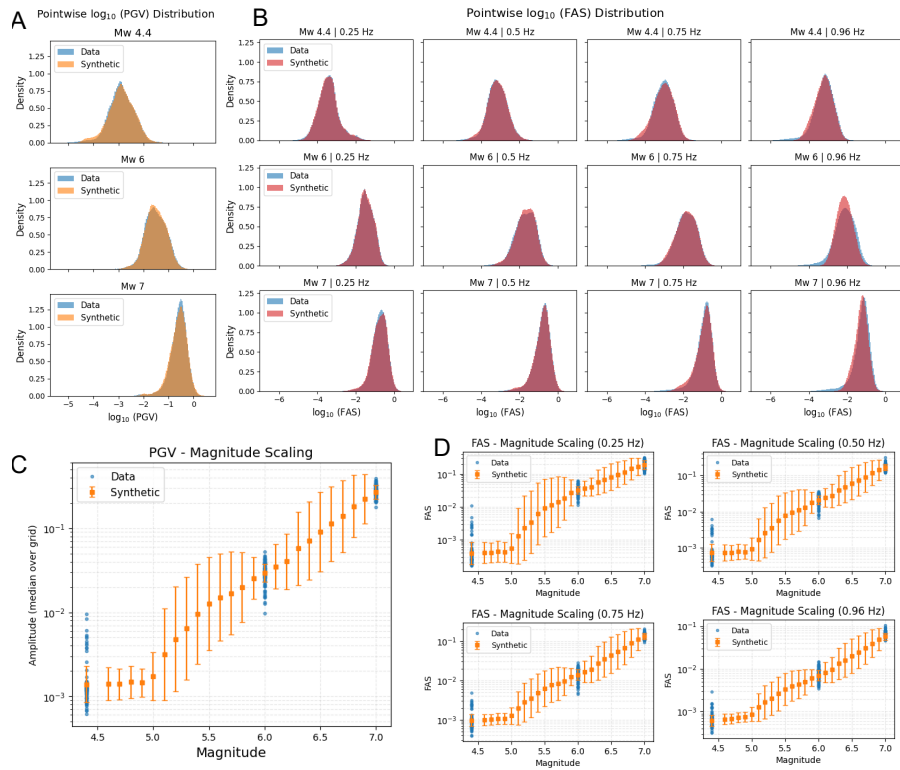


Figure D.12: Statistical distribution and magnitude scaling laws of spectral-calibrated generated ground motions

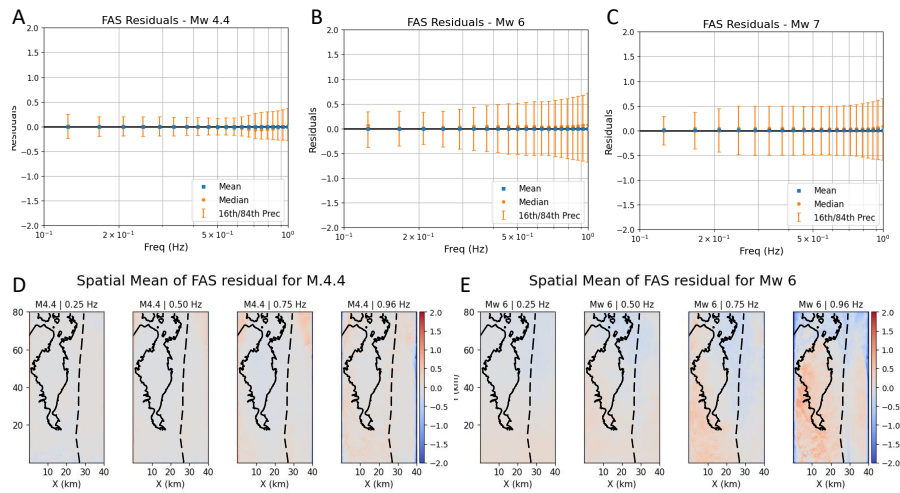


Figure D.13: Frequency-dependent spectral residual with calibrated synthetics.

BIBLIOGRAPHY

- Aagaard, Brad T and Evan T Hirakawa (2021). “San Francisco Bay region 3D seismic velocity model v21. 1”. In: *US Geological Survey (USGS) Data Release*, p. 1310.
- Abdal, Rameen, Peihao Zhu, Niloy J Mitra, and Peter Wonka (2021). “Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows”. In: *ACM Transactions on Graphics (ToG)* 40.3, pp. 1–21. ISSN: 0730-0301.
- Abrahamson, Norman, Nicholas Gregor, and Kofi Addo (2016). “BC Hydro ground motion prediction equations for subduction earthquakes”. In: *Earthquake Spectra* 32.1, pp. 23–44. ISSN: 8755-2930.
- Abrahamson, Norman A, Walter J Silva, and Ronnie Kamai (2014). “Summary of the ASK14 ground motion relation for active crustal regions”. In: *Earthquake Spectra* 30.3, pp. 1025–1055. ISSN: 8755-2930.
- Aki, Keiiti and Paul Richards (2002). *Quantitative seismology*. MIT Press. ISBN: 1-891389-63-7.
- Albergo, Michael S and Eric Vanden-Eijnden (2022). “Building normalizing flows with stochastic interpolants”. In: *arXiv preprint arXiv:2209.15571*.
- Alkin, Benedikt, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter (2024). “Universal physics transformers: A framework for efficiently scaling neural operators”. In: *Advances in Neural Information Processing Systems* 37, pp. 25152–25194.
- Ambrosio, Luigi, Nicola Gigli, and Giuseppe Savaré (2005). *Gradient flows: in metric spaces and in the space of probability measures*. Springer.
- Archuleta, Ralph J (1984). “A faulting model for the 1979 Imperial Valley earthquake”. In: *Journal of Geophysical Research: Solid Earth* 89 (B6), pp. 4559–4585. ISSN: 0148-0227.
- Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). “Wasserstein generative adversarial networks”. In: *International conference on machine learning*. Pmlr, pp. 214–223. ISBN: 2640-3498.
- Azizzadenesheli, Kamyar, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar (2024). “Neural operators for accelerating scientific simulations and design”. In: *Nature Reviews Physics* 6.5, pp. 320–328. ISSN: 2522-5820.
- Baglio, Marco, Alessandro Cardoni, Gian Paolo Cimellaro, and Norman Abrahamson (2023). “Generating ground motions using the Fourier amplitude spectrum”. In: *Earthquake Engineering & Structural Dynamics* 52.15, pp. 4884–4899. ISSN: 0098-8847.

- Bahrampouri, Mahdi, Adrian Rodriguez-Marek, Shrey Shahi, and Haitham Dawood (2021). “An updated database for ground motion parameters for KiK-net records”. In: *Earthquake Spectra* 37.1, pp. 505–522. ISSN: 8755-2930.
- Baker, Jack, Brendon Bradley, and Peter Stafford (2021). *Seismic hazard and risk analysis*. Cambridge University Press. ISBN: 1-108-60490-0.
- Baker, Jack W (2011). “Conditional mean spectrum: Tool for ground-motion selection”. In: *Journal of structural engineering* 137.3, pp. 322–331. ISSN: 0733-9445.
- Bayless, Jeff and Norman A Abrahamson (2019). “An empirical model for the interfrequency correlation of epsilon for Fourier amplitude spectra”. In: *Bulletin of the Seismological Society of America* 109.3, pp. 1058–1070. ISSN: 0037-1106.
- Benamou, Jean-David and Yann Brenier (2000). “A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem”. In: *Numerische Mathematik* 84.3, pp. 375–393. ISSN: 0029-599X.
- Bhosale, AS, Robin Davis, and Pradip Sarkar (2017). “Vertical irregularity of buildings: regularity index versus seismic risk”. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 3.3, p. 04017001. ISSN: 2376-7642.
- Bogachev, Vladimir Igorevich (1998). *Gaussian measures*. Issue: 62. American Mathematical Soc. ISBN: 0-8218-1054-5.
- Bonneel, Nicolas, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister (2015). “Sliced and radon wasserstein barycenters of measures”. In: *Journal of Mathematical Imaging and Vision* 51.1, pp. 22–45. ISSN: 0924-9907.
- Boore, David M (1983). “Stochastic simulation of high-frequency ground motions based on seismological models of the radiated spectra”. In: *Bulletin of the Seismological Society of America* 73.6, pp. 1865–1894. ISSN: 1943-3573.
- (2003). “Simulation of ground motion using the stochastic method”. In: *Pure and applied geophysics* 160.3, pp. 635–676. ISSN: 0033-4553.
- (2010). “Orientation-independent, nongeometric-mean measures of seismic intensity from two horizontal components of motion”. In: *Bulletin of the Seismological Society of America* 100.4, pp. 1830–1835. ISSN: 1943-3573.
- Boore, David M, Jennie Watson-Lamprey, and Norman A Abrahamson (2006). “Orientation-independent measures of ground motion”. In: *Bulletin of the seismological Society of America* 96.4, pp. 1502–1511. ISSN: 1943-3573.
- Brémaud, Pierre (2020). *Probability theory and stochastic processes*. Springer. ISBN: 3-030-40182-0.
- Burks, Lynne S and Jack W Baker (2014). “Validation of ground-motion simulations through simple proxies for the response of engineered systems”. In: *Bulletin of the Seismological Society of America* 104.4, pp. 1930–1946. ISSN: 1943-3573.

- Causse, Mathieu, Fabrice Cotton, Cécile Cornou, and Pierre-Yves Bard (2008). “Calibrating median and uncertainty estimates for a practical use of empirical Green’s functions technique”. In: *Bulletin of the Seismological Society of America* 98.1, pp. 344–353. ISSN: 1943-3573.
- Chen, Ricky TQ, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud (2018). “Neural ordinary differential equations”. In: *Advances in neural information processing systems* 31.
- Cheng, Chaoran, Boran Han, Danielle Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W Mahoney, and Bernie Wang (2025). “Gradient-free generation for hard-constrained systems”. In: *International Conference on Learning Representations*. Vol. 2025, pp. 100510–100539.
- Chizat, Lenaïc, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard (2018). “Unbalanced optimal transport: Dynamic and Kantorovich formulations”. In: *Journal of Functional Analysis* 274.11, pp. 3090–3123. ISSN: 0022-1236.
- Dabaghi, Mayssa and Armen Der Kiureghian (2017). “Stochastic model for simulation of near-fault ground motions”. In: *Earthquake Engineering & Structural Dynamics* 46.6, pp. 963–984. ISSN: 0098-8847.
- Damianou, Andreas and Neil D Lawrence (2013). “Deep gaussian processes”. In: *Artificial intelligence and statistics*. PMLR, pp. 207–215.
- Di Giovanni, Francesco, James Rowbottom, Benjamin P Chamberlain, Thomas Markovich, and Michael M Bronstein (2022). “Understanding convolution on graphs via energies”. In: *arXiv preprint arXiv:2206.10991*.
- Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). “Density estimation using real nvp”. In: *arXiv preprint arXiv:1605.08803*.
- Dong, Yihe, Jean-Baptiste Cordonnier, and Andreas Loukas (2021). “Attention is not all you need: Pure attention loses rank doubly exponentially with depth”. In: *International conference on machine learning*. PMLR, pp. 2793–2803. ISBN: 2640-3498.
- Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly (2020). “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929*.
- Douglas, John and Hideo Aochi (2008). “A survey of techniques for predicting earthquake ground motions for engineering purposes”. In: *Surveys in geophysics* 29.3, pp. 187–220. ISSN: 0169-3298.
- Dupont, Emilien, Yee Whye Teh, and Arnaud Doucet (2021). “Generative models as distributions of functions”. In: *arXiv preprint arXiv:2102.04776*.
- Dutordoir, Vincent, Alan Saul, Zoubin Ghahramani, and Fergus Simpson (2023). “Neural diffusion processes”. In: *International Conference on Machine Learning*. PMLR, pp. 8990–9012. ISBN: 2640-3498.

- Esfahani, Reza DD, Fabrice Cotton, Matthias Ohrnberger, and Frank Scherbaum (2023). “TFCGAN: Nonstationary ground-motion simulation in the time–frequency domain using conditional generative adversarial network (CGAN) and phase retrieval methods”. In: *Bulletin of the Seismological Society of America* 113.1, pp. 453–467. ISSN: 0037-1106.
- Fang, Yuxin, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang (2023). “Unleashing vanilla vision transformer with masked image modeling for object detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6244–6253.
- Flamary, Rémi, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fstras, and Nemo Fournier (2021). “Pot: Python optimal transport”. In: *Journal of Machine Learning Research* 22.78, pp. 1–8. ISSN: 1533-7928.
- Florez, Manuel A, Michaelangelo Caporale, Pakpoom Buabthong, Zachary E Ross, Domniki Asimaki, and Men-Andrin Meier (2022). “Data-driven synthesis of broadband earthquake ground motions using artificial intelligence”. In: *Bulletin of the Seismological Society of America* 112.4, pp. 1979–1996. ISSN: 0037-1106.
- Frankel, Arthur (2009). “A constant stress-drop model for producing broadband synthetic seismograms: Comparison with the Next Generation Attenuation relations”. In: *Bulletin of the Seismological Society of America* 99.2, pp. 664–680. ISSN: 1943-3573.
- Gagne, David John, Hannah M Christensen, Aneesh C Subramanian, and Adam H Monahan (2020). “Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz’96 model”. In: *Journal of Advances in Modeling Earth Systems* 12.3, e2019MS001896. ISSN: 1942-2466.
- Garnelo, Marta, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh (2018). “Neural processes”. In: *arXiv preprint arXiv:1807.01622*.
- Gatti, Filippo and Didier Clouteau (2020). “Towards blending physics-based numerical simulations and seismic databases using generative adversarial network”. In: *Computer Methods in Applied Mechanics and Engineering* 372, p. 113421. ISSN: 0045-7825.
- Gilmer, Justin, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl (2017). “Neural message passing for quantum chemistry”. In: *International conference on machine learning*. Pmlr, pp. 1263–1272. ISBN: 2640-3498.
- Goodfellow, Ian (2016). “Deep learning”. In.
- Goodfellow, Ian J, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). “Generative adversarial nets”. In: *Advances in neural information processing systems* 27.

- Gordon, Jonathan, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner (2019). “Convolutional conditional neural processes”. In: *arXiv preprint arXiv:1910.13556*.
- Goulet, Christine A, Jennie Watson-Lamprey, Jack Baker, Curt Haselton, and Nico Luco (2008). “Assessment of ground motion selection and modification (GMSM) methods for non-linear dynamic analyses of structures”. In: *Geotechnical Earthquake Engineering and Soil Dynamics IV*, pp. 1–10.
- Grathwohl, Will, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud (2018). “Ffjord: Free-form continuous dynamics for scalable reversible generative models”. In: *arXiv preprint arXiv:1810.01367*.
- Graves, Robert and Arben Pitarka (2015). “Refinements to the Graves and Pitarka (2010) broadband ground-motion simulation method”. In: *Seismological Research Letters* 86.1, pp. 75–80. ISSN: 1938-2057.
- (2016). “Kinematic ground-motion simulations on rough faults including effects of 3D stochastic velocity perturbations”. In: *Bulletin of the Seismological Society of America* 106.5, pp. 2136–2153. ISSN: 1943-3573.
- Graves, Robert W and Arben Pitarka (2010). “Broadband ground-motion simulation using a hybrid approach”. In: *Bulletin of the Seismological Society of America* 100.5, pp. 2095–2123. ISSN: 0037-1106.
- Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola (2012). “A kernel two-sample test”. In: *The journal of machine learning research* 13.1, pp. 723–773. ISSN: 1532-4435.
- Guidotti, Roberto, Hana Chmielewski, Vipin Unnikrishnan, Paolo Gardoni, Therese McAllister, and John van de Lindt (2016). “Modeling the resilience of critical infrastructure: The role of network dependencies”. In: *Sustainable and resilient infrastructure* 1.3, pp. 153–168. ISSN: 2378-9689.
- Han, Ruiyu (2023). “Sliced Wasserstein distance between probability measures on Hilbert spaces”. In: *arXiv preprint arXiv:2307.05802*.
- Han, Xu, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Li-Ping Liu (2022). “Predicting physics in mesh-reduced space with temporal attention”. In: *arXiv preprint arXiv:2201.09113*.
- Hanks, Thomas C and Robin K McGuire (1981). “The character of high-frequency strong ground motion”. In: *Bulletin of the Seismological Society of America* 71.6, pp. 2071–2095. ISSN: 1943-3573.
- He, Yingqing, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen (2022). “Latent video diffusion models for high-fidelity long video generation”. In: *arXiv preprint arXiv:2211.13221*.
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33, pp. 6840–6851.

- Ho, Jonathan, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet (2022). “Video diffusion models”. In: *Advances in neural information processing systems* 35, pp. 8633–8646.
- Huang, Rongjie, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao (2023). “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models”. In: International Conference on Machine Learning. PMLR, pp. 13916–13932. ISBN: 2640-3498.
- Hutchinson, Michael F (1989). “A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines”. In: *Communications in Statistics-Simulation and Computation* 18.3, pp. 1059–1076. ISSN: 0361-0918.
- Jaegle, Andrew, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, and Evan Shelhamer (2021). “Perceiver io: A general architecture for structured inputs & outputs”. In: *arXiv preprint arXiv:2107.14795*.
- Jankowiak, Martin, Geoff Pleiss, and Jacob Gardner (2020). “Deep sigma point processes”. In: Conference on uncertainty in artificial intelligence. PMLR, pp. 789–798. ISBN: 2640-3498.
- Jin, Yang, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin (2025). “Pyramidal flow matching for efficient video generative modeling”. In: International Conference on Learning Representations. Vol. 2025, pp. 23378–23402.
- Kantorovich, L and G Rubinshtein (1958). “On a space of totally additive functions, vestn”. In: *Vestn Lening. Univ* 13, pp. 52–59.
- Katsanos, Evangelos I, Anastasios G Sextos, and George D Manolis (2010). “Selection of earthquake ground motion records: A state-of-the-art review from a structural engineering perspective”. In: *Soil dynamics and earthquake engineering* 30.4, pp. 157–169. ISSN: 0267-7261.
- Kerrigan, Gavin, Justin Ley, and Padhraic Smyth (2022). “Diffusion generative models in infinite dimensions”. In: *arXiv preprint arXiv:2212.00886*.
- Kerrigan, Gavin, Giosue Migliorini, and Padhraic Smyth (2023). “Functional flow matching”. In: *arXiv preprint arXiv:2305.17209*.
- (2024). “Dynamic conditional optimal transport through simulation-free flows”. In: *Advances in Neural Information Processing Systems* 37, pp. 93602–93642.
- Kim, Hyunjik, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh (2019). “Attentive neural processes”. In: *arXiv preprint arXiv:1901.05761*.
- Kolmogorov, Andreï Nikolaevich (2018). *Foundations of the theory of probability: Second English Edition*. Courier Dover Publications. ISBN: 0-486-82979-0.

- Kong, Qingkai, Daniel T Trugman, Zachary E Ross, Michael J Bianco, Brendan J Meade, and Peter Gerstoft (2019). “Machine learning in seismology: Turning data into insights”. In: *Seismological research letters* 90.1, pp. 3–14. ISSN: 0895-0695.
- Konno, Katsuaki and Tatsuo Ohmachi (1998). “Ground-motion characteristics estimated from spectral ratio between horizontal and vertical components of microtremor”. In: *Bulletin of the Seismological Society of America* 88.1, pp. 228–241. ISSN: 1943-3573.
- Kottke, Albert and Stickler Bot (n.d.). “arkottke/pyrotd v0. 5.4”. In: *Zenodo* (). URL: <https://zenodo.org/record/1322849>.
- Kottke, Albert R, Norman A Abrahamson, David M Boore, Yousef Bozorgnia, Christine A Goulet, Justin Hollenback, Tadahiro Kishida, Olga-Joan Ktenidou, Ellen M Rathje, and Walter J Silva (2021). “Selection of random vibration theory procedures for the NGA-East project and ground-motion modeling”. In: *Earthquake Spectra* 37.1, pp. 1420–1439. ISSN: 8755-2930.
- Kou, Peng, Feng Gao, and Xiaohong Guan (2013). “Sparse online warped Gaussian process for wind power probabilistic forecasting”. In: *Applied energy* 108, pp. 410–428. ISSN: 0306-2619.
- Kovachki, Nikola, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2023). “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.
- Lavrentiadis, Grigorios, Yongfei Wang, Norman A Abrahamson, Yousef Bozorgnia, and Christine A Goulet (2025). “A seismologically consistent surface rupture length model for unbounded and width-limited events”. In: *Earthquake Spectra* 41.4, pp. 2859–2879. ISSN: 8755-2930.
- Li, Chunyuan, Changyou Chen, David Carlson, and Lawrence Carin (2016). “Pre-conditioned stochastic gradient Langevin dynamics for deep neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. ISBN: 2374-3468.
- Li, Qimai, Zhichao Han, and Xiao-Ming Wu (2018). “Deeper insights into graph convolutional networks for semi-supervised learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. ISBN: 2374-3468.
- Li, Tianhong and Kaiming He (2025). “Back to basics: Let denoising generative models denoise”. In: *arXiv preprint arXiv:2511.13720*.
- Li, Yuanming, Bonhwa Ku, Shou Zhang, Jae-Kwang Ahn, and Hanseok Ko (2020a). “Seismic data augmentation based on conditional generative adversarial networks”. In: *Sensors* 20.23, p. 6850. ISSN: 1424-8220.

- Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2020b). “Fourier neural operator for parametric partial differential equations”. In: *arXiv preprint arXiv:2010.08895*.
- (2020c). “Neural operator: Graph kernel network for partial differential equations”. In: *arXiv preprint arXiv:2003.03485*.
- Li, Zongyi, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, and Kamyar Azizzadenesheli (2023). “Geometry-informed neural operator for large-scale 3d pdes”. In: *Advances in Neural Information Processing Systems* 36, pp. 35836–35854.
- Lim, Jae Hyun, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, and Jan Kautz (2025). “Score-based diffusion models in function space”. In: *Journal of Machine Learning Research* 26.158, pp. 1–62. ISSN: 1533-7928.
- Lipman, Yaron, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le (2022). “Flow matching for generative modeling”. In: *arXiv preprint arXiv:2210.02747*.
- Liu, Haitao, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai (2020). “When Gaussian process meets big data: A review of scalable GPs”. In: *IEEE transactions on neural networks and learning systems* 31.11, pp. 4405–4423. ISSN: 2162-237X.
- Liu, Haohe, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley (2023). “Audioldm: Text-to-audio generation with latent diffusion models”. In: *arXiv preprint arXiv:2301.12503*.
- Liu, Pengcheng, Ralph J Archuleta, and Stephen H Hartzell (2006). “Prediction of broadband ground-motion time histories: Hybrid low/high-frequency method with correlated random source parameters”. In: *Bulletin of the Seismological Society of America* 96.6, pp. 2118–2130. ISSN: 1943-3573.
- Liu, Qihao, Xi Yin, Alan Yuille, Andrew Brown, and Mannat Singh (2025). “Flowing from words to pixels: A noise-free framework for cross-modality evolution”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2755–2765.
- Liu, Xingchao, Chengyue Gong, and Qiang Liu (2022). “Flow straight and fast: Learning to generate and transfer data with rectified flow”. In: *arXiv preprint arXiv:2209.03003*.
- Liu-Schiaffini, Miguel, Julius Berner, Boris Bonev, Thorsten Kurth, Kamyar Azizzadenesheli, and Anima Anandkumar (2024). “Neural operators with localized integral and differential kernels”. In: *arXiv preprint arXiv:2402.16845*.

- Lu, Cheng, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu (2022). “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps”. In: *Advances in neural information processing systems* 35, pp. 5775–5787.
- Luco, Nicolas and Paolo Bazzurro (2007). “Does amplitude scaling of ground motion records result in biased nonlinear structural drift responses?” In: *Earthquake Engineering & Structural Dynamics* 36.13, pp. 1813–1835. ISSN: 0098-8847.
- Lyu, Dongwei, Rie Nakata, Pu Ren, Michael W Mahoney, Arben Pitarka, Nori Nakata, and N Benjamin Erichson (2025). “Rapid wavefield forecasting for earthquake early warning via deep sequence to sequence learning”. In: *Nature Communications*. ISSN: 2041-1723.
- Maechling, Philip J, Fabio Silva, Scott Callaghan, and Thomas H Jordan (2015). “SCEC broadband platform: System architecture and software implementation”. In: *Seismological Research Letters* 86.1, pp. 27–38. ISSN: 1938-2057.
- Marano, Giuseppe Carlo, Marco Martino Rosso, Angelo Aloisio, and Giansalvo Cirrincione (2024). “Generative adversarial networks review in earthquake-related engineering fields”. In: *Bulletin of Earthquake Engineering* 22.7, pp. 3511–3562. ISSN: 1570-761X.
- Maroñas, Juan, Oliver Hamelijnck, Jeremias Knoblauch, and Theodoros Damoulas (2021). “Transforming Gaussian processes with normalizing flows”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1081–1089. ISBN: 2640-3498.
- Matinfar, Mehrshad, Naser Khaji, and Goodarz Ahmadi (2023). “Deep convolutional generative adversarial networks for the generation of numerous artificial spectrum-compatible earthquake accelerograms using a limited number of ground motion records”. In: *Computer-Aided Civil and Infrastructure Engineering* 38.2, pp. 225–240. ISSN: 1093-9687.
- McCallen, David, Anders Petersson, Arthur Rodgers, Arben Pitarka, Mamun Miah, Floriana Petrone, Bjorn Sjogreen, Norman Abrahamson, and Houjun Tang (2021a). “EQSIM—A multidisciplinary framework for fault-to-structure earthquake simulations on exascale computers part I: Computational models and workflow”. In: *Earthquake Spectra* 37.2, pp. 707–735. ISSN: 8755-2930.
- McCallen, David, Floriana Petrone, Mamun Miah, Arben Pitarka, Arthur Rodgers, and Norman Abrahamson (2021b). “EQSIM—A multidisciplinary framework for fault-to-structure earthquake simulations on exascale computers, part II: Regional simulations of building response”. In: *Earthquake Spectra* 37.2, pp. 736–761. ISSN: 8755-2930.
- McCallen, David, Arben Pitarka, Houjun Tang, Rie Nakata, Khalid M Mosalam, Floriana Petrone, Selim Günay, and Claudio Perez (2025). “An open-access simulated earthquake ground-motion database for an M7 Hayward Fault earthquake

- in the San Francisco Bay Region”. In: *Earthquake Spectra* 41.3, pp. 2560–2597. ISSN: 8755-2930.
- McCallen, David, Arben Pitarka, Houjun Tang, Ramesh Pankajakshan, N Anders Petersson, Mamun Miah, and Junfei Huang (2024). “Regional-scale fault-to-structure earthquake simulations with the EQSIM framework: Workflow maturation and computational performance on GPU-accelerated exascale platforms”. In: *Earthquake Spectra* 40.3, pp. 1615–1652. ISSN: 8755-2930.
- Miller, M and J Baker (2015). “Ground-motion intensity and damage map selection for probabilistic infrastructure network risk assessment using optimization”. In: *Earthquake Engineering & Structural Dynamics* 44.7, pp. 1139–1156. ISSN: 0098-8847.
- Mostowsky, Peter, Vincent Dutordoir, Iskander Azangulov, Noémie Jaquier, Michael John Hutchinson, Aditya Ravuri, Leonel Rozo, Alexander Terenin, and Viacheslav Borovitskiy (2025). “The GeometricKernels package: Heat and Matérn kernels for geometric learning on manifolds, meshes, and graphs”. In: *Journal of Machine Learning Research* 26.276, pp. 1–14. ISSN: 1533-7928.
- Naeim, Farzad and Marshall Lew (1995). “On the use of design spectrum compatible time histories”. In: *Earthquake spectra* 11.1, pp. 111–127. ISSN: 8755-2930.
- National Research Council (2014). “Lessons Learned from the Fukushima Nuclear Accident for Improving Safety of US Nuclear Plants”. In.
- National Research Institute for Earth Science and Disaster Resilience (2019). “Nied k-net, kik-net”. In: *National Research Institute for Earth Science and Disaster Resilience*.
- Palgunadi, Kadek Hendrawan, Andreas Bergmeister, Andrea Bosisio, Laura Ermert, Maria Koroni, Nathanaël Perraudin, Simon Dirmeier, and Men-Andrin Meier (2024). “High resolution seismic waveform generation using denoising diffusion”. In: *arXiv preprint arXiv:2410.19343*.
- Paolucci, Roberto, Chiara Smerzini, and Manuela Vanini (2021). “BB-SPEEDset: A validated dataset of broadband near-source earthquake ground motions from 3D physics-based numerical simulations”. In: *Bulletin of the Seismological Society of America* 111.5, pp. 2527–2545. ISSN: 0037-1106.
- Papamakarios, George, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan (2021). “Normalizing flows for probabilistic modeling and inference”. In: *Journal of Machine Learning Research* 22.57, pp. 1–64. ISSN: 1533-7928.
- Pathak, Jaideep, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, and Kamyar Azizzadenesheli (2022). “Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators”. In: *arXiv preprint arXiv:2202.11214*.

- Peebles, William and Saining Xie (Dec. 19, 2022). *Scalable Diffusion Models with Transformers*. arXiv.org. URL: <https://arxiv.org/abs/2212.09748v2> (visited on 05/15/2026).
- Pidstrigach, Jakiw, Youssef Marzouk, Sebastian Reich, and Sven Wang (2024). “Infinite-dimensional diffusion models”. In: *Journal of Machine Learning Research* 25.414, pp. 1–52. ISSN: 1533-7928.
- Pinilla-Ramos, Camilo, Arben Pitarka, David McCallen M. EERI, and Rie Nakata (2025). “Performance evaluation of the USGS velocity model for the San Francisco Bay Area”. In: *Earthquake Spectra* 41.1, pp. 457–494. ISSN: 8755-2930.
- Pitarka, Arben, Robert Graves, Kojiro Irikura, Ken Miyakoshi, Changjiang Wu, Hiroshi Kawase, Arthur Rodgers, and David McCallen (2022). “Refinements to the Graves–Pitarka kinematic rupture generator, including a dynamically consistent slip-rate function, applied to the 2019 M w 7.1 Ridgecrest earthquake”. In: *Bulletin of the Seismological Society of America* 112.1, pp. 287–306. ISSN: 0037-1106.
- Pitarka, Arben, Hong Kie Thio, Paul Somerville, and Luis Fabian Bonilla (2013). “Broadband ground-motion simulation of an intraslab earthquake and nonlinear site response: 2010 Ferndale, California, earthquake case study”. In: *Seismological Research Letters* 84.5, pp. 785–795. ISSN: 1938-2057.
- Price, Ilan, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, and Peter Battaglia (2025). “Probabilistic weather forecasting with machine learning”. In: *Nature* 637.8044, pp. 84–90. ISSN: 0028-0836.
- Rahman, Md Ashiqur, Manuel A. Florez, Anima Anandkumar, Zachary E. Ross, and Kamyar Azizzadenesheli (2022). “Generative Adversarial Neural Operators”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=X1VzbBU6xZ>.
- Rahman, Md Ashiqur, Zachary E. Ross, and Kamyar Azizzadenesheli (2023). “U-NO: U-shaped Neural Operators”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=j3oQF9coJd>.
- Rezaeian, Sanaz and Armen Der Kiureghian (2012). “Simulation of orthogonal horizontal ground motion components for specified earthquake and site characteristics”. In: *Earthquake Engineering & Structural Dynamics* 41.2, pp. 335–353. ISSN: 0098-8847.
- Rodgers, Arthur J, Arben Pitarka, Ramesh Pankajakshan, Bjorn Sjögreen, and N Anders Petersson (2020). “Regional-scale 3D ground-motion simulations of M w 7 earthquakes on the Hayward Fault, northern California resolving frequencies 0–10 Hz and including site-response corrections”. In: *Bulletin of the Seismological Society of America* 110.6, pp. 2862–2881. ISSN: 0037-1106.

- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022). “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.
- Sabetta, Fabio, Antonio Pugliese, Gabriele Fiorentino, Giovanni Lanzano, and Lucia Luzi (2021). “Simulation of non-stationary stochastic ground motions based on recent Italian earthquakes”. In: *Bulletin of Earthquake Engineering* 19.9, pp. 3287–3315. ISSN: 1570-761X.
- Salimbeni, Hugh and Marc Deisenroth (2017). “Doubly stochastic variational inference for deep Gaussian processes”. In: *Advances in neural information processing systems* 30.
- Scholz, Christopher H (2019). *The mechanics of earthquakes and faulting*. Cambridge university press. ISBN: 1-107-16348-X.
- Schwerdtner, Paul, Frederick Law, Qing Wang, Cenk Gazen, Yi-Fan Chen, Matthias Ihme, and Benjamin Peherstorfer (2024). “Uncertainty quantification in coupled wildfire–atmosphere simulations at scale”. In: *PNAS nexus* 3.12, p. e554. ISSN: 2752-6542.
- Seidman, Jacob H, Georgios Kissas, George J Pappas, and Paris Perdikaris (2023). “Variational autoencoding neural operators”. In: *arXiv preprint arXiv:2302.10351*.
- Shi, Jian (2019). *Improving site response analysis for earthquake ground motion modeling*. California Institute of Technology. ISBN: 979-8-3798-5774-5.
- Shi, Yaozhong, Angela F. Gao, Zachary E. Ross, and Kamyar Azizzadenesheli (2024a). “Universal Functional Regression with Neural Operator Flows”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=rHL329Xa3X>.
- Shi, Yaozhong, Grigorios Lavrentiadis, Domniki Asimaki, Zachary E Ross, and Kamyar Azizzadenesheli (2024b). “Broadband ground-motion synthesis via generative adversarial neural operators: Development and validation”. In: *Bulletin of the Seismological Society of America* 114.4, pp. 2151–2171. ISSN: 0037-1106. DOI: <https://doi.org/10.1785/0120230207>.
- Shi, Yaozhong, Grigorios Lavrentiadis, Konstantinos Tsalouchidis, Zachary E Ross, David McCallen, Caifeng Zou, Kamyar Azizzadenesheli, and Domniki Asimaki (2026a). “Large-Scale 3D Ground-Motion Synthesis with Physics-Inspired Latent Operator Flow Matching”. In: *arXiv preprint arXiv:2603.17403 (Under review)*.

- Shi, Yaozhong, Zachary Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2026b). “Stochastic process learning via operator flow matching”. In: *Advances in Neural Information Processing Systems* 38, pp. 38186–38226.
- Shi, Yaozhong, Zachary E. Ross, Domniki Asimaki, and Kamyar Azizzadenesheli (2025). “Mesh-Informed Neural Operator : A Transformer Generative Approach”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=K8qAuRfv0G>.
- Sitzmann, Vincent, Eric Chan, Richard Tucker, Noah Snively, and Gordon Wetzstein (2020). “Metasdf: Meta-learning signed distance functions”. In: *Advances in Neural Information Processing Systems* 33, pp. 10136–10147.
- Skilling, John (1989). “The eigenvalues of mega-dimensional matrices”. In: *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*. Springer, pp. 455–466.
- Song, Yang, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole (2020). “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456*.
- Stafford, Peter J (2017). “Interfrequency correlations among Fourier spectral ordinates and implications for stochastic ground-motion simulation”. In: *Bulletin of the Seismological Society of America* 107.6, pp. 2774–2791. ISSN: 0037-1106.
- State of California (n.d.). *How It Works – California Earthquake Early Warning*. URL: <https://www.earthquake.ca.gov/how-it-works/>.
- Sun, Hongyu, Zachary E Ross, Weiqiang Zhu, and Kamyar Azizzadenesheli (2023). “Phase neural operator for multi-station picking of seismic arrivals”. In: *Geophysical Research Letters* 50.24, e2023GL106434. ISSN: 0094-8276.
- Swiler, Laura P, Mamikon Gulian, Ari L Frankel, Cosmin Safta, and John D Jakeman (2020). “A survey of constrained Gaussian process regression: Approaches and implementation challenges”. In: *Journal of Machine Learning for Modeling and Computing* 1.2. ISSN: 2689-3967.
- Takamoto, Makoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert (2022). “Pdebench: An extensive benchmark for scientific machine learning”. In: *Advances in neural information processing systems* 35, pp. 1596–1611.
- Tavakoli, Behrooz and Shahram Pezeshk (2005). “Empirical-stochastic ground-motion prediction for eastern North America”. In: *Bulletin of the Seismological Society of America* 95.6, pp. 2283–2296. ISSN: 1943-3573.
- Tong, Alexander, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrod Rector-Brooks, Guy Wolf, and Yoshua Bengio (2024). “Improving and generalizing flow-based generative models with minibatch optimal transport”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=CD9Snc73AW>.

- Touhami, Sara, Filippo Gatti, Fernando Lopez-Caballero, Régis Cottureau, Lúcio de Abreu Corrêa, Ludovic Aubry, and Didier Clouteau (2022). “SEM3D: A 3D high-fidelity numerical earthquake simulator for broadband (0–10 Hz) seismic response prediction at a regional scale”. In: *Geosciences* 12.3, p. 112. ISSN: 2076-3263.
- Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). “Instance normalization: The missing ingredient for fast stylization”. In: *arXiv preprint arXiv:1607.08022*.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.
- Wald, David J and Thomas H Heaton (1994). “Spatial and temporal distribution of slip for the 1992 Landers, California, earthquake”. In: *Bulletin of the Seismological Society of America* 84.3, pp. 668–691. ISSN: 1943-3573.
- Wan, Team, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, and Jianxiao Yang (2025). “Wan: Open and advanced large-scale video generative models”. In: *arXiv preprint arXiv:2503.20314*.
- Wang, Sifan, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J Pappas, and Paris Perdikaris (2024a). “Cvit: Continuous vision transformer for operator learning”. In: *arXiv preprint arXiv:2405.13998*.
- Wang, Tian and Chuang Wang (2024b). “Latent neural operator for solving forward and inverse pde problems”. In: *Advances in Neural Information Processing Systems* 37, pp. 33085–33107.
- Wang, Tianren, Teng Zhang, and Brian Lovell (2021a). “Faces a la carte: Text-to-face generation via attribute disentanglement”. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3380–3388.
- Wang, Tiantong, Daniel Trugman, and Youzuo Lin (2021b). “SeismoGen: Seismic waveform synthesis using GAN with application to seismic data augmentation”. In: *Journal of Geophysical Research: Solid Earth* 126.4, e2020JB020077. ISSN: 2169-9313.
- Wang, Tiantong, Zhongping Zhang, and Youzuo Li (2019). “EarthquakeGen: Earthquake generator using generative adversarial networks”. In: *SEG International Exposition and Annual Meeting*. Seg, D043S119R004.
- Welling, Max and Yee W Teh (2011). “Bayesian learning via stochastic gradient Langevin dynamics”. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688.
- Wen, Gege, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson (2022). “U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow”. In: *Advances in Water Resources* 163, p. 104180. ISSN: 0309-1708.

- Williams, Christopher KI and Carl Edward Rasmussen (2006). *Gaussian processes for machine learning*. Vol. 2. Issue: 3. MIT press Cambridge, MA.
- Wu, Haixu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long (2024). “Transolver: A fast transformer solver for pdes on general geometries”. In: *arXiv preprint arXiv:2402.02366*.
- Wu, Yuxin and Kaiming He (2018). “Group normalization”. In: Proceedings of the European conference on computer vision (ECCV), pp. 3–19.
- Yang, Yan, Angela F Gao, Kamyar Azizzadenesheli, Robert W Clayton, and Zachary E Ross (2023). “Rapid seismic waveform modeling and inversion with neural operators”. In: *IEEE Transactions on Geoscience and Remote Sensing* 61, pp. 1–12. ISSN: 0196-2892.
- Yang, Yan, Angela F Gao, Jorge C Castellanos, Zachary E Ross, Kamyar Azizzadenesheli, and Robert W Clayton (2021). “Seismic wave propagation and inversion with neural operators”. In: *The Seismic Record* 1.3, pp. 126–134. ISSN: 2694-4006.
- Yao, Jiachen, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Animashree Anandkumar (2026). “Guided diffusion sampling on function spaces with applications to pdes”. In: *Advances in Neural Information Processing Systems* 38, pp. 127057–127094.
- Yu, Wenbo, Anirbit Ghosh, Tobias Sebastian Finn, Rossella Arcucci, Marc Bocquet, and Sibor Cheng (2025). “A probabilistic approach to wildfire spread prediction using a denoising diffusion surrogate model”. In: *arXiv preprint arXiv:2507.00761*.
- Zhao, Qingqing, David B Lindell, and Gordon Wetzstein (2022). “Learning to solve pde-constrained inverse problems with graph networks”. In: *arXiv preprint arXiv:2206.00711*.