UC Irvine
UC Irvine Electronic Theses and Dissertations

Title
Infinite-Dimensional Generative Models through the Transport of Measure

Permalink
https://escholarship.org/uc/item/87f632kx

Author
Kerrigan, Gavin

Publication Date
2024

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Infinite-Dimensional Generative Models through the Transport of Measure

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Gavin Kerrigan

Dissertation Committee:
Distinguished Professor Padhraic Smyth, Chair
Associate Professor Stephan Mandt
Professor Erik Sudderth

2024

© 2024 Gavin Kerrigan

DEDICATION

To Carl Sagan, without whom this journey might never have begun.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

Above all, I am indebted to my advisor, Padhraic Smyth, without whom this journey might never have been completed. Padhraic's boundless support, in times of both triumph and tribulation, has been an essential ingredient in the creation of this dissertation. I am particularly grateful for the freedom, both intellectual and otherwise, with which he has entrusted me these last few years. Our countless conversations have had an immeasurable impact on my development as a researcher and beyond, and will not soon be forgotten.

I would also like to thank my committee members, Stephan Mandt and Erik Sudderth. I am especially grateful for Stephan's willingness to take a risk to offer me a role in organizing AISTATS, through which I learned a tremendous amount. In addition, I am particularly grateful for Erik's role in organizing the HPI Research Center at UCI, which provided invaluable opportunities to connect students and faculty both at home and abroad.

To my labmates, both present and former – Eric Nalisnick, Jihyun Park, Chris Galbraith, Disi Ji, Robby Logan, Alex Boyd, Rachel Longjohn, Markelle Kelly, Sam Showalter, Edgar Robles, Catarina Belém, Yuxin Chang, Giosuè Migliorini, Shang Wu, Coen Adler, Noah Benjamin, Saatvik Kher – I cannot thank you enough for making UCI a welcoming, exciting, and enjoyable place. I am also thankful to those I have been fortunate enough to collaborate with and learn from: Mark Steyvers, Heliodoro Tejeda, Dylan Slack, Jens Tuyls, Justin Ley, Prakhar Srivastava, Ruihan Yang, Gideon Dresdner, Jeremy McGibbon, Chris Bretherton, Clèment Guilloteau, Efi Foufoula-Georgiou, Neda Dolatabadi, and Kai Nelson.

More personally, I am eternally grateful for the support of my friends and family, both near and far. Finally, I sincerely thank Jax, who has unwittingly been an audience member to uncountably many practice talks, undercooked ideation sessions, Zoom calls, and out-loud bouts of debugging.

The work presented in this dissertation was supported by the National Science Foundation under the award 1900644, by the National Institutes of Health under award R01-LM013344, by the HPI Research Center in Machine Learning and Data Science at UC Irvine, and by a Qualcomm Faculty award.

VITA

Gavin Kerrigan

EDUCATION

Doctor of Philosophy in Computer Science **2024**
University of California, Irvine *Irvine, CA*

Bachelor of Science in Mathematics **2019**
The Pennsylvania State University *State College, PA*

RESEARCH EXPERIENCE

Graduate Research Assistant **2019–2024**
University of California, Irvine *Irvine, CA*

TEACHING EXPERIENCE

Instructor **Spring 2023**
CS 178: Machine Learning and Data Mining *University of California, Irvine*

Teaching Assistant **Fall 2022**
CS 178: Machine Learning and Data Mining *University of California, Irvine*

Mathematics Tutor **2017-2019**
Penn State Learning *The Pennsylvania State University*

ACADEMIC SERVICE

Workflow Chair **2024**
The 27th International Conference on AI and Statistics (AISTATS) *Valencia, Spain*

Session Chair **2024**
The 27th International Conference on AI and Statistics (AISTATS) *Valencia, Spain*

Academic Reviewing
JMLR, ICLR, CVPR, ICML, UAI, AAAI, AISTATS, NeurReps Workshop **2024**

NeurIPS, AISTATS, Deep Generative Models for Health, **2023**
Workshop on Topology, Algebra, and Geometry in Data Science,
Workshop on Diffusion Models
ICML, Workshop on Geometrical and Topological Representation Learning **2022**
Workshop on Human-Machine Collaboration and Teaming

REFEREED CONFERENCE PUBLICATIONS

* *denotes joint authorship*

Gavin Kerrigan, Giosuè Migliorini, and Padhraic Smyth. Dynamic Conditional Optimal Transport through Simulation-Free Flows. To appear in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Prakhar Srivastava, Ruihan Yang, Gavin Kerrigan, Gideon Dresdner, Jeremy McGibbon, Christopher Bretherton, and Stephan Mandt. Precipitation Downscaling with Spatiotemporal Video Diffusion. To appear in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Gavin Kerrigan, Giosuè Migliorini, and Padhraic Smyth. Functional Flow Matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 238:3934-3942, 2024 (Outstanding Student Paper Award).

Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion Generative Models in Infinite Dimensions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 206:9538-9563, 2023.

Gavin Kerrigan, Padhraic Smyth, and Mark Steyvers. Combining Human Predictions with Model Probabilities via Confusion Matrices and Calibration. In *Neural Information Processing Systems (NeurIPS)*, pp.44214434, 2021.

REFEREED JOURNAL PUBLICATIONS

Mark Steyvers, Heliodoro Tejeda, Gavin Kerrigan, Padhraic Smyth. Bayesian Modeling of Human-AI Complementarity. In *Proceedings of the National Academy of the Sciences (PNAS)*, 119(11):1-7, 2022.

REFEREED WORKSHOP PUBLICATIONS

Gavin Kerrigan*, Dylan Slack*, Jens Tuyls*. Differentially Private Language Models Benefit from Public Pre-Training. In *Proceedings of the Second Workshop on Privacy in NLP*, pp.39-45, 2020.

IN SUBMISSION

Gavin Kerrigan, Kai Nelson, and Padhraic Smyth. EventFlow: Forecasting Continuous-Time Event Data with Flow Matching. *Under review*, 2024.

Clement Guilloteau, Gavin Kerrigan, Kai Nelson, Giosuè Migliorini, Padhraic Smyth, Runze Li, and Efi Foufoula-Georgiou. A Generative Diffusion Model for Probabilistic Ensembles of Precipitation Maps Conditioned on Multisensor Satellite Observations. *Under review*, 2024.

ABSTRACT OF THE DISSERTATION

Infinite-Dimensional Generative Models through the Transport of Measure

By

Gavin Kerrigan

Doctor of Philosophy in Computer Science

University of California, Irvine, 2024

Distinguished Professor Padhraic Smyth, Chair

A pervasive and often tacit assumption in generative modeling is that our data distribution is finite-dimensional. These finite-dimensional distributions frequently arise from a discrete representation of some continuous underlying signal. Images, for instance, are represented as a finite collection of pixels, and generative models are typically built directly on top of this pixel-level representation.

However, our world is not made of pixels, and building faithful models of our world requires moving beyond this assumption. This is particularly true for data modalities like partial differential equations and time series, where the multi-scale or irregularly sampled nature of this data is a key feature. In this dissertation, we develop both the theory and methodology necessary to build generative models for infinite-dimensional data. In particular, we focus on a class of models which can be understood through the lens of measure transport.

We begin with an overview of this class of models and a review of the necessary mathematical background. We build upon this background to develop techniques for building diffusion and flow-based generative models for infinite-dimensional data. Our focus then shifts to conditional generation tasks and the application of optimal transport techniques within flow-based models. Finally, we apply flow-based techniques to forecast continuous-time event data before concluding with a discussion of several remaining challenges.

Chapter 1

Introduction

Can a machine compose a symphony? Paint a portrait? Author an ode to a loved one? Write code, prove a theorem, be a friend? A mere decade ago, the answer to these questions would likely have been a resounding *no*. Today, though, paradigm-changing advances in generative modeling have unequivocally narrowed the gap between machine and man.

At its core, generative modeling seeks to build artifacts which allow us to ingest and understand a collection of data, and to produce new data using the information thus gleaned. Contemporary methods are chiefly probabilistic, where a dataset is understood as a collection of samples from some probability distribution. The principal goal of generative modeling, then, is to obtain a faithful representation of this data generating distribution. Depending on the task at hand, this representation can be leveraged to draw new samples, evaluate existing data, or otherwise enhance our understanding of the underlying mechanisms which produced the data we observed.

Representations of the data we collect are, of course, discrete entities, and existing generative models are typically built directly on top of this discrete representation. However, many phenomena of interest, ranging from vision and audio to time series and fluid dynamics, are

most naturally thought of as a arising from a continuous underlying signal. For instance, an image, saved as a finite collection of pixels, is a discrete representation of the corresponding physical phenomenon. Such signals can often be thought of as functions living in some infinite-dimensional space.

In this dissertation, we aim to move beyond the assumption that our data is finite-dimensional and develop generative models which are able to both respect and leverage the underlying functional nature of many data sources. To do so, we develop methodological techniques for building generative models in infinite-dimensional spaces as well as the theoretical properties which serve as the foundation for such models.

This has several distinct advantages across a number of domains. For instance,

- **Time Series.** By considering a univariate time series as a function, one can build generative models to perform probabilistic forecasting of these models which allows one to incorporate data which is observed at irregular intervals, as well as forecast at arbitrary times.
- **PDEs.** Solutions to partial differential equations (PDEs) are functions. By treating these objects as functions, one may incorporate assumptions on the structure of the solutions, such as smoothness, providing a rigorous link between contemporary data-driven approaches and classical mathematical techniques.
- **Natural Images.** When generating natural images, it is typical to work at a single, fixed resolution. However, there may be scenarios in which one is interested in dynamically choosing the size of an image, e.g., reducing the size of an image when it is being generated on a mobile device in order to limit energy usage. By posing a generative model of images which is functional, one may generate images at an arbitrary resolution chosen at test-time.

- **Point Processes.** A fruitful point of view on point processes is to consider them as an infinite collection of joint distributions, with the nth distribution representing the event times given that exactly n events have occurred. By modeling this infinite-dimensional distribution, one is able to overcome limitations with existing autoregressive models, which are trained only to represent univariate one-step-ahead predictions.

1.1 Generative Models and Measure Transport

One of the most broad and successful classes of contemporary generative models are those based on the notion of *measure transport*, and this dissertation is focused on this setting. These models approximate the data distribution by transforming a simple reference distribution (e.g., a Gaussian) over many steps. This iterative process is naturally seen as occurring over time, and treating these models as dynamic processes is a fruitful point of view. In this section, we provide an informal and conceptual treatment of generative models through this lens, and we will see how this perspective allows us to both unify and generalize existing approaches.

1.1.1 Curves in the Space of Measures

Suppose we are interested in sampling from a probability measure μ_1 supported on $X = \mathbb{R}^d$, but we only have indirect access to this distribution. For instance, we often find ourselves in situations in which we only have i.i.d. samples from μ_1, or in which we only know the density of μ_1 up to a multiplicative constant.

When $X = \mathbb{R}$ and the CDF F of μ_1 is known, a classical technique is *inversion sampling*. To produce samples from μ_1, we begin by drawing a sample x_0 from the uniform measure μ_0 on $[0, 1]$. This source sample is transformed via the inverse CDF F^{-1} to obtain a data

sample, i.e., $x_1 = F^{-1}(x_0)$ is a sample from μ_1. From a global perspective, the distribution μ_1 is obtained via the *pushforward* $\mu_1 = F^{-1}_{\#}\mu_0$ of μ_0 along the inverse CDF.[1]

Can we play a similar game in higher dimensions and without knowing the CDF of μ_1? We might hope that we can transform a tractable source distribution μ_0 into the data distribution μ_1 through some learned mapping $T : X \to X$ with $\mu_1 \approx T_{\#}\mu_0$. A natural strategy is to learn a mapping T which minimizes a discrepancy measure of the form $D(\mu_1, T_{\#}\mu_0)$, such as the KL divergence, between the true data distribution μ_1 and our learned approximation $T_{\#}\mu_0$. Some models, such as GANs [Goodfellow et al., 2014], indeed carry out this plan.

Finding such a transformation T is challenging. Any model which directly transforms pure noise into, say, a realistic image, must be quite sophisticated. Contemporary generative models, such as diffusions [Song et al., 2021] and flows [Lipman et al., 2023], instead decompose this generative process into a sequence of easier-to-learn steps. By composing many small transformations, one hopes to learn a complex transformation while sidestepping the need to directly model it. Taking this perspective to its extreme, we might seek to learn *infinitesimal* transformations, which are composed not through a sequence of discrete steps, but rather via continuous dynamics.

To make this idea more precise, let us use $\mathbb{P}(X)$ to represent the space of probability measures over $X = \mathbb{R}^d$. This space will serve a geometric backdrop in which we may build generative models. We begin by fixing a reference measure $\mu_0 \in \mathbb{P}(X)$, which should be thought of as a point in the geometric space $\mathbb{P}(X)$. Our goal will be to design or otherwise learn a *curve* $(\mu_t)_{t\in[0,1]}$ in $\mathbb{P}(X)$ which (approximately) interpolates between our reference measure μ_0 and the data measure μ_1.

To describe the infinitesimal evolution of such a curve, we will need some notion of its *tangents*.

[1] Slightly more formally, if $T : X \to X$ is a given transformation, the pushforward measure $\nu = T_{\#}\mu$ is given by $\nu(A) = T_{\#}\mu(A) = \mu(T^{-1}(A))$. In terms of probability densities, this is the familiar change-of-variables formula. That is, if ν admits a density q and μ admits a density p, then $q(x) = p(T^{-1}(x))|DT^{-1}(x)|$.

It turns out that the correct notion is that of a time-dependent vector field $v_t : X \to X$ [Ambrosio et al., 2005]. To see why this is the case, suppose $x_0 \sim \mu_0$ is a sample from our reference distribution. If we begin at x_0 and flow this sample along the vector field v_t, then its position at any later time t is described by the *flow map* $\phi_t : X \to X$. That is, $x_t = \phi_t(x_0)$ is the solution of the ordinary differential equation

$$\mathrm{d}\phi_t(x_0) = v_t(\phi_t(x_0))\,\mathrm{d}t \qquad \phi_0(x_0) = x_0 \qquad t \in [0,1]. \tag{1.1}$$

By drawing many reference samples $x_0 \sim \mu_0$ and flowing each independently along v_t, we obtain their distribution μ_t at any later time $t \in [0,1]$ via the pushforward $\mu_t = [\phi_t]_{\#}\mu_0$.

In other words, the vector field v_t describes the *local* dynamics of our process, transporting individual samples, whereas the *global* dynamics of our process are captured by the path of measures $(\mu_t)_{t\in[0,1]}$. The key link between the local and global perspectives is the *continuity equation*

$$\partial_t \mu_t + \mathrm{div}(v_t \mu_t) = 0 \qquad t \in [0,1]. \tag{1.2}$$

That is, when a pair (v_t, μ_t) solve the continuity equation, they describe the same dynamics, where Equation (1.1) describes the transport of individual samples and Equation (1.2) describes the corresponding transport of measure.

Interestingly, a given path of measures $(\mu_t)_{t\in[0,1]}$ can also be described via stochastic dynamics, where the evolution of an individual sample is now described by a stochastic differential equation of the form

$$\mathrm{d}x_t = v_t(x_t)\,\mathrm{d}t + \sqrt{2}\sigma_t(x_t)\,\mathrm{d}W_t \qquad x_0 \sim \mu_0 \qquad t \in [0,1] \tag{1.3}$$

where W_t is a Brownian motion on $X = \mathbb{R}^d$, $v : [0,1] \times X \to X$ is a drift vector field, and

$\sigma : [0,1] \times X \to \mathbb{R}$ is a scalar diffusion coefficient. In this case, the *Fokker-Planck-Kolmogorov* equation [Bogachev et al., 2022] plays the role of the continuity equation:

$$\partial_t \mu_t + \operatorname{div}(\mu_t v_t) = \Delta(g\mu_t) \tag{1.4}$$

where the Laplacian $\Delta(g\mu_t)$ accounts for the stochastic component of the local transport in Equation (1.3).

Overall, this interplay between the local and global perspective is a recurring theme in contemporary generative modeling, and many techniques are predicated on an analysis of the corresponding continuity equation.

1.1.2 Generative Models as Curves

So far, we have described a fairly abstract perspective on generative models, where our stated aim is to design an interpolant between a given reference measure μ_0 and the data measure μ_1. Broadly, there are two strategies for obtaining such an interpolant. In the first, we do not specify a particular path of measures, but rather allow the model to discover the path itself. In the second and more performant approach, we as practitioners explicitly define the path of measures.

In all cases, the object we parametrize in practice is the vector field which generates the corresponding path of measures. Upon learning this vector field, samples from the model distribution are drawn by sampling a source point $x_0 \sim \mu_0$ and numerically solving the corresponding (potentially stochastic) flow differential equation. Let us now make this procedure somewhat more concrete by focusing on several classes of generative models.

Normalizing Flows The earliest class of generative models which fit into this framework are normalizing flows [Papamakarios et al., 2019]. In particular, continuous normalizing flows [Chen et al., 2018] learn a parametric vector field $v_t^\theta : X \to X$ which implicitly defines a path of measures via the continuity equation. These models are trained via maximum likelihood, i.e. minimizing the KL divergence $\mathsf{KL}\left[\mu_1 \| [\phi_1]_\# \mu_0\right]$ between the true data distribution and the distribution $[\phi_1]_\# \mu_0$ obtained by flowing samples from μ_0 along the learned vector field v_t^θ.

Although this approach has the advantage of decomposing the generative process, it suffers from several limitations. Foremost, evaluating the KL divergence during training requires flowing samples along the learned vector field, which is computationally expensive. However, a more conceptual disadvantage is that we are asking the model to do the hard work of finding an interpolating path of measures $(\mu_t)_{t\in[0,1]}$. It is thus difficult to regularize the corresponding path of measures, e.g., encouraging the model to learn short paths which are cheap to simulate.

Diffusion Models On the other hand, diffusion models, particularly continuous-time variants [Song et al., 2021], define *a priori* a path of measures which we seek to learn. This is achieved by first specifying an SDE which interpolates between the data distribution μ_1 and a reference measure μ_0.[2]. This step requires no learning, as we are merely turning our data into noise. For instance, when μ_0 is a Gaussian, this SDE can be specified via an inhomogeneous Ornstein-Uhlenbeck process, which shrinks datapoints towards the origin while simultaneously adding Gaussian noise.

The *reverse process* is obtained by solving this SDE backwards in time, thereby giving us a path of measures interpolating from the reference measure μ_0 to the data measure μ_1. The corresponding time-reversed SDE, though, is intractable, and the drift vector field of this intractable SDE is modeled via a regression-style objective. Discrete-time diffusion models,

[2]We note that there is a caveat to this story – an interpolant is only achieved *asymptotically*, but the convergence towards μ_0 is exponentially fast [Pedrotti et al., 2024]

like DDPM [Ho et al., 2020], can be seen as modeling a fixed time-discretization of this scheme.

Flow Matching Similar to diffusion models, the recently introduced framework of flow matching[3] [Lipman et al., 2023, Albergo and Vanden-Eijnden, 2023, Liu et al., 2023] defines a path of measures *a priori*. However, the vector field which generates this path, in the sense of the continuity equation, is intractable. This vector field is again learned via a regression-style loss.

Compared with diffusion models, deterministic transport is typically used. Moreover, flow matching typically allows for greater flexibility in the design of the interpolating path than diffusion models. For instance, in flow matching, one can use arbitrary source distributions [Tong et al., 2024, Albergo et al., 2024], which can be difficult to achieve in diffusion models. Moreover, in flow matching, the path of measures can be more carefully designed. For instance, the performance of flow matching models can be improved by using *geodesics*, or shortest paths in $\mathbb{P}(X)$, as the desired interpolant. These geodesics arise naturally from optimal transport notions, which we study further in Chapter 5.

1.2 Structure and Contributions of the Dissertation

This dissertation presents several new techniques for constructing transport-based generative models in infinite-dimensional spaces, as detailed in the subsequent chapters. Carrying out this program requires overcoming both theoretical and methodological challenges.

Foremost, generative models in Euclidean spaces typically make an implicit assumption that all

[3]Several works contemporary with flow matching [Lipman et al., 2023], such as stochastic interpolants [Albergo and Vanden-Eijnden, 2023] and rectified flows [Liu et al., 2023], develop equivalent ideas. We generally use the terminology *flow matching* to refer to this class of models.

measures admit a density with respect to the Lebesgue measure, and subsequent constructions and analyses depend heavily on this assumption. However, in infinite dimensional spaces, there is no analogue of the Lebesgue measure [Eldredge, 2016], and thus we need to work directly with the underlying probability measures. A common theme throughout this dissertation is that infinite-dimensional probability measures often requires careful analyses to elucidate the conditions under which we may obtain a well-defined model.

More practically, generative models in function spaces frequently require us to learn mappings between infinite-dimensional spaces. Standard model architectures are not applicable in this setting, and we thus often require specialized models. A second common theme in this dissertation is the use of neural operators [Kovachki et al., 2021, Li et al., 2021] as a backbone for generative modeling tasks.

More specifically, the structure and contributions of this dissertation are as follows:

- Chapter 2 introduces technical background information which is essential in the developments of the later chapters, in addition to defining various pieces of notation used throughout the dissertation.

- Chapter 3 develops a discrete-time diffusion model for data measures supported in separable Hilbert spaces. This was the first work to develop function-space diffusion models, and was previously published in Kerrigan et al. [2023]. Some specific contributions of this chapter include the development of diffusion models using Gaussian process-valued noise, a functional ELBO used for training the model, techniques for approximating the ELBO using finite-dimensional quantities, and the identification of the Cameron-Martin space as playing a key role in function-space models.

- In Chapter 4, we develop techniques allowing us generalize the flow matching framework to infinite-dimensional spaces. The content of this chapter was previously published in Kerrigan et al. [2024a]. In comparison with Chapter 3, these models are continuous-time

and allow for deterministic sampling. Key contributions of this chapter include the development of the first flow-based generative model in function spaces, a rigorous theoretical framework which underpins these models, and a comprehensive evaluation against existing function-space generative models.

- Chapter 5 focuses on conditional generative models. In particular, recent work [Tong et al., 2024] shows that designing interpolants using optimal transport techniques can lead to strong gains in performance. However, prior to our work, the theory of *conditional* optimal transport was not sufficiently developed (even in the Euclidean setting) to justify these techniques for conditional generative modeling. Thus, we develop a theory of dynamic conditional optimal transport, and study the geometry of the resulting conditional Wasserstein space. Equipped with these tools, we propose a technique for conditional generation based on the flow matching framework. The results in this chapter again are applicable to infinite-dimensional settings.

  Some selected contributions include closed-form expressions for the conditional Wasserstein distance between Gaussian measures, identification of the conditional Wasserstein space as a geodesic space and a conditional analogue of the McCann interpolant theorem, a complete characterization of the absolutely continuous curves in the conditional Wasserstein space, a conditional analogue of the Benamou-Brenier theorem, and a comprehensive empirical evaluation of our proposed method.

- Chapter 6 represents somewhat of a departure from the previous chapters, which all focus on the Hilbert space setting. Instead, we shift our attention to *configuration spaces*. These infinite-dimensional spaces consist of counting measures, and are a natural framework for formalizing continuous-time event sequence data. Continuing our study of transport-based generative models, we develop a novel and highly performant methodology for generating and forecasting temporal point processes.

- Lastly, Chapter 7 concludes the dissertation and discusses several remaining challenges.

Chapter 2

Background

In this section, we provide a brief review of the mathematical background needed in this dissertation. Our aim is not to be comprehensive, but rather only to highlight the notions which will play a central role in the chapters which are to follow. We begin by defining some common notation, followed by a more detailed discussion.

We use $(X, \mathcal{A}, \omega)$ to represent a generic measure space. Frequently, $X \subseteq \mathbb{R}^d$ plays the role of a domain on which functions are defined. When X is a topological space, $\mathcal{B}(X)$ will denote its Borel σ-algebra. For a measurable space $(X, \mathcal{A})$, the set $\mathbb{P}(X)$ denotes the space of probability measures on $(X, \mathcal{A})$. Hilbert spaces are denoted by H, Y, U and, unless otherwise stated, are equipped with their Borel σ-algebras. Elements of these spaces are typically denoted $f, g, h \in H$ or $u \in U$ and $y \in Y$. We use e.g. $\pi^Y : Y \times U \to Y$ to denote the canonical projection maps $\pi^Y : (y, u) \mapsto y$.

The set of learnable parameters for a model is denoted by θ. We use δ to denote a unit-mass measure $\delta[x] \in \mathbb{P}(X)$ located at $x \in X$, or as an indicator, e.g., $\delta_{ij} = 1$ if and only if $i = j$. Other lowercase Greek letters (e.g. μ, ν, η, γ) typically denote measures. If $\eta \in \mathbb{P}(Y \times U)$ is a joint probability distribution, $\pi^Y_\# \eta$ represents its Y-marginal (and respectively for U).

2.1 Functional Analysis

The setting in which the majority of this work takes places is that of Hilbert spaces. While other spaces are certainly of interest, e.g., Banach spaces, this dissertation focuses on the Hilbert case, as the existence of an inner product is invaluable when performing calculations. We assume the reader is familiar with basic analytic notions. For comprehensive treatments, we refer to Folland [1999] and Rudin [1973]. See also Axler [2020] and Lax [2014] for more elementary treatments of this material.

Definition 1 (Hilbert Space [Axler, 2020, 8.21])**.**
A ***Hilbert Space*** *H is an inner product space which is complete under the induced norm.*

At an intuitive level, a Hilbert space is a natural generalization of the familiar Euclidean spaces $\mathbb{R}^d$ with the standard inner product. For our purposes, the key abstraction that Hilbert spaces enable is that Hilbert spaces may be infinite-dimensional. Many, but not all, spaces of functions are indeed Hilbert spaces.

It is worth singling out the *separable* class of Hilbert spaces, which are particularly friendly to work with.

Definition 2 (Separability [Folland, 1999, Ch. 4])**.**
A Hilbert space H is said to be ***separable*** *if it contains a countable dense subset.*

Separability is a key assumption, which informally restricts our Hilbert space from being too large. As the following theorem shows, separability is equivalent to the existence of a *countable* orthonormal basis. While even non-separable Hilbert spaces always admit bases [Folland, 1999, Prop. 5.28], countability will be an invaluable assumption when it comes to performing calculations in these spaces.

Definition 3 (Orthonormal Basis [Axler, 2020, Ch. 8])**.**
A subset $\{e_k\}_{k\in K}$ of H is said to be an ***orthonormal basis*** *for H if*

1. $\langle e_k, e_j \rangle = \delta_{ij}$ *for all* $i, j \in K$

2. *The closure of the linear span of* $\{e_k\}_{k \in K}$ *is* H*, i.e.* $\overline{span\{e_k\}_{k \in K}} = H$*.*

Recall that $\overline{\text{span}\{e_k\}_{k \in K}}$ consists of all sums of the form $\sum_{k \in K} \alpha_k e_k$ where $\alpha_k \in \mathbb{R}$ and $\sum_{k \in K} \alpha_k^2 < \infty$ [Axler, 2020, Prop. 8.58].

Theorem 4 ([Folland, 1999, Prop 5.29])**.**
A Hilbert space H *is separable if and only if it admits a countable orthonormal basis.*

If $f \in H$ is an element of a separable Hilbert space, we may express this vector via a basis expansion of the form $f = \sum_{k=1}^{\infty} \langle f, e_k \rangle e_k$ [Axler, 2020, Prop. 8.58]. While any two separable Hilbert spaces are isometrically isomorphic [Rudin, 1973, Chapter 12, Ex. 24], the choice of inner product may carry important consequences for computation and learning.

Examples We conclude this section with a few examples.

- The quintessential example of a Hilbert space is the Lebesgue space

$$L^2(X, \omega) = \left\{ f : X \to \mathbb{R} : \int f^2 \, \mathrm{d}\omega < \infty \right\} \tag{2.1}$$

  consisting of the square-integrable measurable functions $f : (X, \mathcal{A}, \omega) \to \mathbb{R}$, where $(X, \mathcal{A}, \omega)$ is an arbitrary measure space. Here, functions are identified up to sets of ω-measure zero. This space is equipped with the inner product and norm

$$\langle f, g \rangle = \int fg \, \mathrm{d}\omega \qquad\qquad \|f\|_2^2 = \langle f, f \rangle = \int f^2 \, \mathrm{d}\omega. \tag{2.2}$$

  Under mild additional assumptions on the underlying measure space, the space $L^2(X, \omega)$ is separable. For instance, $L^2(\mathbb{R}^d, \lambda)$ is separable, where λ is the Lebesgue measure [Folland, 1999, Pg. 187].

- The Lebesgue spaces

$$L^p(X,\omega) = \left\{ f : X \to \mathbb{R} : \int |f|^p \, \mathrm{d}\omega < \infty \right\} \tag{2.3}$$

for $1 \leq p < \infty$ are *not* Hilbert spaces when $p \neq 2$. This is because we equip $L^p(X,\omega)$ with the norm $\|f\|_p^p = \int |f|^p \, \mathrm{d}\omega$, which is not induced by any inner product [Folland, 1999, Pg. 187].

2.2 Measure and Probability

We again assume the reader is familiar with the basic notions of measure-theoretic probability, but we recall a few key notions which appear throughout this dissertation. We refer to Kallenberg [1997] and Durrett [2019] for a comprehensive introduction.

We first single out a condition which will play a recurring role throughout this dissertation. Namely, the *absolute continuity* of two measures provides a structural constraint on their supports. Under this assumption, the *Radon-Nikodym* theorem is an invaluable tool for showing the existence of densities.

Definition 5 (Absolute Continuity [Folland, 1999, Ch. 3.2])**.**
*We say that ν is **absolutely continuous with respect to** μ, denoted $\nu \ll \mu$, if $\mu(A) = 0$ implies that $\nu(A) = 0$ for any measurable set $A \in \mathcal{A}$.*

*We say that ν is **equivalent** to μ if $\mu \ll \nu$ and $\nu \ll \mu$. Similarly, ν and μ are said to be **singular** if neither measure is absolutely continuous with respect to the other.*

Theorem 6 (Radon-Nikodym [Axler, 2020, 9.36])**.**
Suppose that μ, ν are probability measures on $(X, \mathcal{A})$ and $\nu \ll \mu$. Then, there exists a

measurable $f : X \to [0, \infty)$ *such that*

$$\nu(A) = \int_A f \, \mathrm{d}\mu \tag{2.4}$$

for all measurable $A \in \mathcal{A}$*. The function* f *is unique up to* μ*-null sets.*

The function f furnished by Theorem (6) is often denoted $\mathrm{d}\nu / \mathrm{d}\mu$, and is called the **Radon-Nikodym derivative** of ν with respect to μ. Intuitively, the Radon-Nikodym derivative can be thought of as the density of ν with respect to the measure μ.

In fact, the common notion of a probability density function is indeed a Radon-Nikodym derivative. That is, when μ is the Lebesgue measure on $\mathbb{R}^d$ and ν is any given probability measure with $\nu \ll \mu$, the Radon-Nikodym derivative $\mathrm{d}\nu / \mathrm{d}\mu$ is precisely the standard probability density function associated with the measure ν. In this case, the absolute continuity restriction means that μ is in a sense diffuse. The Lebesgue (or uniform) measure thus plays a central role in probability theory, serving as a general-purpose reference measure through which one may calculate densities.

However, in an infinite dimensional space H, there does not exist an analogue of the Lebesgue measure. In particular, any such analogue should be translation invariant, in the sense that the measure of the set $A \in \mathcal{A}$ and $A - f$ coincide for any $f \in H$. As the following proposition shows, it is impossible to obtain such a measure which is compatible with the topology on H.

Proposition 7 ([Eldredge, 2016, Theorem 1.1])**.**
Suppose H *is a separable Hilbert space of infinite dimensions. Then, any translation-invariant Borel measure on* H *is either the zero measure, or assigns infinite measure to every open subset of* H*.*

As a consequence, any probabilistic approach on infinite-dimensional spaces necessarily requires us to work directly with the underlying probability measures, as we do not have

a canonical reference measure from which we may define densities. In Chapters 3 and 4, the existence of particular Radon-Nikodym derivatives of probability measures supported in separable Hilbert spaces will be of paramount importance for obtaining well-defined function space generative models.

Lastly, we discuss the notion of disintegrating a product measure. This provides us with a theoretical tool for conditioning joint probability measures, and we frequently make use of disintegration in the subsequent chapters. These notions are particularly important in Chapter 5, where we develop methods for conditional optimal transport.

We begin with the notion of a regular conditional measure, which places some technical conditions on what we hope to obtain when we condition a joint distribution.

Definition 8 (Regular Conditional Measures [Bogachev and Ruas, 2007, Def. 10.4.1])**.** *Let Y, U be separable Hilbert spaces and suppose $\eta \in \mathbb{P}(Y \times U)$ is a Borel probability measure on the product space $Y \times U$. A function $\mathcal{B}(U) \times Y \ni (B, y) \mapsto \eta^y(B) \in \mathbb{R}$ is called a* ***regular conditional measure*** *if*

- *For any fixed $y \in Y$, $\eta^y(-) \in \mathbb{P}(U)$ is a Borel probability measure over U.*
- *For any fixed $B \in \mathcal{B}(U)$, the mapping $y \mapsto \eta^y(B)$ is $\mathcal{B}(Y)$-measurable and integrable with respect to the Y-marginal $\pi^Y_\# \eta \in \mathbb{P}(Y)$.*
- *For any $B \in \mathcal{B}(Y \times U)$, we have that $\eta(B) = \int_Y \eta^y(B^y) \, \mathrm{d}\pi^Y_\# \eta(y)$, where $B^y = \{u \in U : (y, u) \in B\}$ is the y-slice of the set B.*

The following proposition shows that, for a given joint measure $\eta \in \mathbb{P}(Y \times U)$, we are guaranteed to have a corresponding family of regular conditional measures (η^y). Moreover, such a collection of regular conditional measures is essentially unique. This uniqueness allows us to avoid having to choose a particular family of conditional distributions when disintegrating a joint measure.

Proposition 9 (Disintegration of a Joint Measure [Bogachev and Ruas, 2007, Ch. 10.4])**.** *Let Y, U be separable Hilbert spaces and suppose $\eta \in \mathbb{P}(Y \times U)$. Then, the regular conditional measures of η exist, and moreover, they are essentially unique, in the sense that there exists $B \in \mathcal{B}(Y)$ with $\pi^Y_\# \eta(B) = 0$ and the measures η^y are unique for $y \in Y \setminus B$.*

2.2.1 Gaussian Measures

A particularly important class of probability measures are the Gaussian measures. Unlike the Lebesgue measure, Gaussian measures are readily constructed in infinite-dimensional spaces. In this dissertation, Gaussian measures play a central role both as reference distributions (in the sense discussed in Chapter 1) and as a source of noise injected into our generative models. Some material presented in this section originally appeared in Kerrigan et al. [2023], and we refer to Da Prato and Zabczyk [2014] and Bogachev [1998] for further details.

Definition 10 (Gaussian Measures [Kukush, 2020, Ch. 2])**.**
Let $(\Omega, \mathcal{B}, \mathbb{P})$ be a probability space. A measurable function $f : \Omega \to H$ is called a ***Gaussian random element (GRE)*** *if for any $g \in H$, the random variable $\omega \mapsto \langle g, f(\omega) \rangle$ has a (possibly degenerate) Gaussian distribution on $\mathbb{R}$. The pushforward of $\mathbb{P}$ along f, denoted $\mu_f = f_\# \mathbb{P}$, is a* ***Gaussian (probability) measure*** *on H.*

In other words, Gaussian random elements $f \sim \mu_f$ are random variables taking values in H whose one-dimensional projections are Gaussians. Note that Gaussian measures exactly coincide with the standard notion of Gaussian distributions in the special case of $H = \mathbb{R}^n$ equipped with the usual inner product, as a finite-dimensional random vector is Gaussian if and only if all of its one-dimensional projections are univariate Gaussians.

For every GRE $f \sim \mu_f$, there exists a unique mean element $m \in H$ given by

$$m = \int_H f \, \mathrm{d}\mu_f. \tag{2.5}$$

Similarly, there exists a unique linear covariance operator $C : H \to H$ given by

$$Cg = \int_H \langle g, f \rangle f \, \mathrm{d}\mu_f - \langle g, m \rangle m \qquad \forall g \in H. \tag{2.6}$$

In the finite-dimensional setting, C is simply the covariance matrix associated with our Gaussian distribution. The covariance operator C is symmetric, positive semidefinite, and compact. Moreover, C has finite trace, i.e. $\mathrm{tr}(C) = \mathbb{E}[||f||^2] < \infty$. Conversely, given any $m' \in H$ and any symmetric, positive semidefinite, trace-class linear operator $C' : H \to H$, there exists a Gaussian measure having mean m' and covariance operator C'. Thus, Gaussian measures are in one-to-one correspondence with their mean functions and covariance operators Da Prato and Zabczyk [2014, Chapter 2]. Hence, we may write $\mu_f = \mathcal{N}(m, C)$ for such a Gaussian measure. Interestingly, using the identity operator $C = \mathsf{Id}$ is inadmissible, as this is not a trace-class operator. This fact has important implications for building function-space models, as we will later see.

Note that for any $g \in H$, we have that $\langle g, f \rangle \sim \mathcal{N}(\langle g, m \rangle, \langle Cg, g \rangle)$ follows a Gaussian distribution on $\mathbb{R}$ with mean $\langle g, m \rangle \in \mathbb{R}$ and variance $\sigma^2 = \langle Cg, g \rangle \in \mathbb{R}_{\geq 0}$ [Wild et al., 2022].

We now prove a few basic lemmas regarding transformations of GREs. These proofs appeared in Kerrigan et al. [2023]. While these results are likely known to experts, we were unable to find precise statements in the literature, and include these proofs for the sake of completeness.

Lemma 1 (Affine Transformations of GREs)**.**
Let $f \sim \mathcal{N}(m, C)$ be a GRE on H. Then, for $\alpha \in \mathbb{R}$ and $g \in H$, we have that $\alpha f + g \sim \mathcal{N}(\alpha m + g, \alpha^2 C)$.

Proof. Fix any $h \in H$, and note that $\langle h, \alpha f + g \rangle = \alpha \langle h, f \rangle + \langle h, g \rangle$. Since $\langle h, f \rangle \sim \mathcal{N}(\langle h, m \rangle, \langle Ch, h \rangle)$, it follows that $\langle h, \alpha f + g \rangle$ must follow a Gaussian distribution on $\mathbb{R}$

with mean

$$\alpha\langle h, m\rangle + \langle h, g\rangle = \langle h, \alpha m + g\rangle \tag{2.7}$$

and variance

$$\alpha^2\langle Ch, h\rangle = \langle \alpha^2 Ch, h\rangle. \tag{2.8}$$

Thus, we have shown that $\alpha f + g$ is a GRE on H, as its inner product with arbitrary $h \in H$ is Gaussian on $\mathbb{R}$. Moreover, we have computed its mean and covariance operator as claimed. □

Lemma 2 (Sum of Independent GREs)**.**
If $f \sim \mathcal{N}(m_1, C_1)$ and $g \sim \mathcal{N}(m_2, C_2)$ are independent GREs on H, then $f + g \sim \mathcal{N}(m_1 + m_2, C_1 + C_2)$.

Proof. Let $z = f + g$. Write μ_f, μ_g, μ_z for the probability measures of f, g, z respectively. The Fourier transform of μ_f is given by

$$\widehat{\mu}_f(h) = \int_H \exp\left[i\langle h, f\rangle\right] \mathrm{d}\mu_f \qquad \forall h \in H, \tag{2.9}$$

and is given analogously for our other measures. By Bogachev [1998, A.3.17] and the subsequent discussion, a probability measure is uniquely determined by its Fourier transform. Moreover, we have that $\widehat{\mu}_z(h) = \widehat{\mu}_f(h)\widehat{\mu}_g(h)$ for every $h \in H$. Using the expression for the Fourier transform of a Gaussian measure given in Da Prato and Zabczyk [2014, Chapter 2], we see that that

$$\hat{mu}_z(h) = \exp\left[i\langle h, m_1\rangle - \frac{1}{2}\langle C_1 h, h\rangle\right] \exp\left[i\langle h, m_2\rangle - \frac{1}{2}\langle C_2 h, h\rangle\right] \tag{2.10}$$

$$= \exp\left[i\langle \lambda, m_1 + m_2\rangle - \frac{1}{2}\langle (C_1 + C_2)h, h\rangle\right] \tag{2.11}$$

which is precisely the Fourier transform of the measure $\mathcal{N}(m_1 + m_2, C_1 + C_2)$ as desired. $\square$

As previously discussed, the notion of absolute continuity and its implications for the existence of probability density functions are delicate topics in infinite-dimensional spaces. To illustrate this further, recall that by Proposition 7, infinite-dimensional spaces do not admit translation-invariant probability measures. While this precludes the notion of a uniform measure, we might hope for something weaker. For instance, suppose that $h \in H$ is some fixed element, and $T_h : H \to H$ is the translation map $T_h : g \mapsto g + h$. If μ is a given probability measure, is the translated measure $\mu_h = [T_h]_{\#}\mu$ equivalent to μ (in the sense of Definition 5)? In the finite-dimensional case, this is clearly true (at least when μ has full support). However, in infinite dimensions, the following proposition demonstrates that we again find ourselves in trouble.

Proposition 11 ([Bogachev and Smolyanov, 1990])**.**
Let $\mu \in \mathbb{P}(H)$ be any probability measure on a separable Hilbert space H. If μ is equivalent to its translation μ_h for every $h \in H$, then H is finite dimensional.

However, in the special case of a Gaussian measure $\mu = \mathcal{N}(m, C)$, we are able to identify a particular subspace $H_0 \subset H$ such that pushforwards of μ are equivalent under translations from elements in H_0. This space H_0 is the **Cameron-Martin space** associated with the measure μ. In fact, we may explicitly identify this space through the covariance operator C via $H_0 = C^{1/2}(H)$ [Da Prato and Zabczyk, 2014, Chapter 2].

The Cameron-Martin space allows us to obtain a complete characterization of absolute continuity through the Feldman-Hájek theorem. We present here the general case, and its implications for function-space generative modeling are discussed at length in the subsequent chapters. In particular, the consequences of this theorem are of central importance in Chapters 3 and 4.

Theorem 12 (Feldman-Hájek [Da Prato and Zabczyk, 2014, Ch. 2])**.**
*Let $\mu = \mathcal{N}(m_1, C_1)$ and $\nu = \mathcal{N}(m_2, C_2)$ be Gaussian measures on a separable Hilbert space H. We say that μ and ν are **equivalent** if they are mutually absolutely continuous, and **singular** if neither is absolutely continuous with respect to the other. The following statements hold.*

- *The Gaussian measures μ and ν are either equivalent or singular.*
- *They are equivalent if and only if*
  1. *Their Cameron-Martin spaces coincide, i.e., $C_1^{1/2}(H) = C_2^{1/2}(H) = H_0$.*
  2. *The difference in means $m_1 - m_2 \in H_0$ is an element of this Cameron-Martin space.*
  3. *The operator $(C_1^{-1/2}C_2^{1/2})(C_1^{-1/2}C_2^{1/2})^* - I$ is Hilbert-Schmidt on the closure $\overline{H}_0$.*
- *If μ and ν are equivalent and $C_1 = C_2 = C$, then ν-a.s. the Radon-Nikodym derivative $\mathrm{d}\mu/\,\mathrm{d}\nu$ is given by*

$$\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(f) = \exp\left[\langle C^{-1/2}(m_1 - m_2), C^{-1/2}(f - m_2)\rangle - \frac{1}{2}||C^{-1/2}(m_1 - m_2)||^2\right]. \quad (2.12)$$

We will see in the subsequent chapters that this theorem will enable us to calculate the density of one Gaussian measure with respect to another. Moreover, the Cameron-Martin space associated with a chosen Gaussian measure will play a central role in obtaining well-defined function-space generative models.

We conclude this section by noting that the requirements for Gaussian measures to be equivalent are quite strong. For instance, for a Gaussian measure $\mu = \mathcal{N}(m, C)$, the Cameron-Martin space $C^{1/2}(H) = H_0$ is actually of measure zero, i.e. $\mu(H_0) = 0$ [Eldredge, 2016, Prop. 3.11]. Thus, the first condition in Theorem 12 requires that the difference of means lives in a very particular subset of H.

2.3 Optimal Transport

For our last background section, we provide a brief and informal overview of optimal transport theory. The content of this section is particularly relevant to Chapter 5, where we develop a novel dynamic approach to *conditional* optimal transport. For more details, we refer to the standard references of Villani [2009], Santambrogio [2015], and Ambrosio et al. [2005] for theoretical aspects, and Peyré and Cuturi [2019] for computational aspects. Some material in this section originally appeared in Kerrigan et al. [2024b].

Suppose that $\mu, \nu \in \mathbb{P}(X)$ are two given probability measures. The goal of optimal transport is to transform the distribution μ into the distribution ν while incurring the smallest possible cost associated with this transformation. This cost is (at least in the standard setting) pointwise, meaning that we have fixed a cost function $c : X \times X \to \mathbb{R} \cup \{+\infty\}$ such that $c(x_0, x_1)$ measures the cost of moving one unit of mass from x_0 to x_1. Given that many generative models may be viewed as transforming a source distribution into the data distribution, it is perhaps unsurprising that optimal transport notions are frequently of use.

In the classical approach to optimal transport, the *Monge problem* [Monge, 1781] seeks to find a measurable transport map $T : X \to X$ minimizing the expected cost of transport, i.e. corresponding to the solution of the constrained optimization problem

$$\inf_{T} \left\{ \int_X c(x, T(x)) \, \mathrm{d}\mu(x) \mid T_{\#}\mu = \nu \right\}. \tag{2.13}$$

This optimization problem is challenging, though, as it is nonlinear in T. Moreover, the set of admissible transformations may be empty. For instance, if μ is a Dirac delta and ν admits a density, there will be no mapping $T : X \to X$ with $T_{\#}\mu = \nu$ since T can only relocate the Dirac delta.

Much later, the *Kantorovich problem* [Kantorovich, 1942] was introduced as a relaxation of the Monge problem which, conceptually, allows for our transportation plan to split mass from some source location across several target locations. In this setting, we seek an optimal coupling $\gamma \in \Pi(\eta, \nu)$, i.e. a probability distribution over $X \times X$ with marginals η, ν, which solves the constrained optimization problem

$$\inf_{\gamma} \left\{ \int_{X \times X} c(x_0, x_1) \, \mathrm{d}\gamma(x_0, x_1) \mid \gamma \in \Pi(\eta, \nu) \right\}. \tag{2.14}$$

Here, $\gamma(x_0, x_1)$ can be thought of as the amount of mass present at $x_0 \in X$ which is to be transported to $x_1 \in X$. Thus, the Kantorovich problem allows for mass to be split, and moreover, the optimization problem is now linear in γ. Unlike the Monge problem, solutions to the Kantorovich problem exist under quite mild conditions (e.g., the cost is lower semicontinuous and bounded from below [Ambrosio et al., 2013, Theorem 2.5]).

Here, we note that both of these approaches are static, in the sense that the transport problem depends only on the initial and terminal locations of some given mass. The seminal work of Benamou and Brenier [2000] instead interprets the optimal transport problem in a dynamic setting, where the source measure μ is transformed into the target measure ν over time. This point of view opens up deep connections with partial differential equations, geometry, and probability theory.

To elaborate, suppose X is now a normed space and that the cost is a power of the induced distance, i.e., $c(x_0, x_1) = |x_0 - x_1|^p$ for some fixed $1 \leq p < \infty$. We further let $\mathbb{P}_p(X) \subset \mathbb{P}(X)$ denote the subspace of probability measures having finite pth moment, and we assume $\mu, \nu \in \mathbb{P}_p(X)$. When the Kantorovich problem for μ, ν admits a finite solution, the cost of such an optimal coupling is the *p-Wasserstein distance*

$$W_p^p(\eta, \nu) = \min_{\gamma} \left\{ \int_{X \times X} |x_0 - x_1|^p \, \mathrm{d}\gamma(x_0, x_1) \mid \gamma \in \Pi(\eta, \nu) \right\} \tag{2.15}$$

which, as the name suggests, is a metric on the space $\mathbb{P}_p(X)$ [Ambrosio et al., 2005, Section 7.1] [Santambrogio, 2015, Section 5.1]. That is, optimal transport allows us to equip spaces of probability measures with a geometry.

The Benamou-Brenier theorem [Benamou and Brenier, 2000] shows us that this geometry can in fact be obtained in a dynamic sense, where the Wasserstein distance may be thought of as the length of a geodesic in $\mathbb{P}_p(X)$ which interpolates between μ and ν. In particular, the p-Wasserstein distance can be obtained by finding a time-dependent vector field transforming μ to ν across time $t \in [0, 1]$ with minimal energy:

$$W_p^p(\mu, \nu) = \min_{(\gamma_t, v_t)} \left\{ \int_0^1 \int_X |v_t(x)|^p \, \mathrm{d}\gamma_t(x) \, \mathrm{d}t \mid \gamma_0 = \eta, \gamma_1 = \nu, \partial_t \gamma_t + \mathrm{div}(v_t \gamma_t) = 0 \right\}. \quad (2.16)$$

Here, we constrain our minimization problem over the set of measures and vector fields (γ_t, v_t) interpolating between μ and ν, satisfying a continuity equation. The optimal vector field can be viewed as a tangent to the curve (γ_t), and the corresponding Riemannian-like structure is the focus of study of the Otto calculus [Otto, 2001]. Seeing optimal transport through this lens is highly compatible with our perspective on generative models as curves in the space of distributions, and the interplay between these two fields is rich.

In Chapter 5 of this dissertation, we study a generalization of optimal transport to the conditional setting [Hosseini et al., 2023, Carlier et al., 2016], where the goal is now to find a mapping $T : Y \times U \to U$ such that $T(y, -)$ transforms a specified conditional measure $\mu(- \mid y)$ into a target conditional measure $\nu(- \mid y)$. In other words, T must simultaneously solve a collection of optimal transport problems.

In particular, we study a dynamic formulation of COT, and one of our main results is to give a conditional analogue of the Benamou-Brenier theorem (see Theorem 33). As we will later see, conditional optimal transport (COT) is a natural tool for conditional generative

modeling. We note here that the study of COT is still in its early stages, with the general infinite-dimensional static case only recently being sufficiently understood [Hosseini et al., 2023].

Chapter 3

Diffusion Generative Models in Infinite Dimensions

The notions discussed in the previous chapter offer us a set of foundational tools for building probabilistic models on separable Hilbert spaces. Equipped with these tools, the first main contribution of this dissertation is to develop an infinite-dimensional discrete-time diffusion generative model.

Diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020] have recently emerged as a powerful class of generative models on a wide array of domains, ranging from images [Ho et al., 2020, Dhariwal and Nichol, 2021, Saharia et al., 2022, Ramesh et al., 2022a] and video [Ho et al., 2022, Yang et al., 2023] to molecular conformation [Xu et al., 2022]. At an intuitive level, these methods work by iteratively perturbing the data distribution towards a tractable prior via additive Gaussian noise, and generation is performed by learning to undo this transformation.

Existing methods largely assume that the data distribution of interest is supported on a finite-dimensional Euclidean space. However, in many domains, the underlying signal is

inherently *infinite-dimensional*, where the observed data can be seen as a collection of discrete observations of some underlying function. Such datasets are often dubbed *functional* [Ramsay and Silverman, 2008]. For instance, a time series dataset consisting of the temperature collected at a particular location every 24 hours can be seen as a uniform discretization of an underlying continuous-time temperature curve [Febrero-Bande and de la Fuente, 2012].

Although diffusion models have empirically demonstrated strong performance on some functional domains, such as audio signals [Kong et al., 2021, Chen et al., 2021] and time series [Rasul et al., 2021a, Tashiro et al., 2021, Alcaraz and Strodthoff, 2022], existing approaches work directly on an explicit discretization of the input space. It is thus unclear how existing methods relate to the underlying functions of interest. For instance, existing methods can not account for function-level assumptions about the data, such as continuity or smoothness constraints.

This chapter[1] develops a theoretical framework for diffusion generative modeling in separable Hilbert spaces. Our method operates by adding Gaussian process noise directly to our infinite-dimensional functions. We learn to reverse this process by performing variational inference in function space, in which we minimize the KL divergence between a known Gaussian measure and a variational family of Gaussian measures. See the discussion in Chapter 2 for additional background on Gaussian measures.

In addition to this framework, we propose practical methods for approximating functional KL divergences by discretizing the underlying operators and empirically verify our framework on several synthetic and real-world benchmarks. In our experiments, our diffusion models are implemented via neural networks that parametrize mappings between function spaces, i.e. neural operators [Li et al., 2021, 2020, Kovachki et al., 2021]. We propose methods that allow for both unconditional and conditional generation of function-valued data. Importantly, our

[1]The content of this chapter was previously published as *Diffusion Generative Models in Infinite Dimensions* (AISTATS 2023) [Kerrigan et al., 2023], with minor modifications.

approach allows us to work with arbitrary non-uniform discretizations, thereby allowing us to train on datasets where the observation set varies across functions. Moreover, we are able to query our generated functions at arbitrary input locations.

3.1 Related Work

We begin by reviewing a selection of related work, appearing before the publication of the developments presented in this chapter [Kerrigan et al., 2023].

Diffusion models are most typically applied to data living in a Euclidean space having a fixed, finite dimension (e.g., see Sohl-Dickstein et al. [2015], Ho et al. [2020], Dhariwal and Nichol [2021], Ho et al. [2020] amongst others). More recent work has extended these methods to Riemannian manifolds, but still with a finite-dimensional assumption [De Bortoli et al., 2022, Huang et al., 2022].

Most relevant to this chapter are diffusion models for signals, such as audio [Chen et al., 2021, Kong et al., 2021], time series [Rasul et al., 2021a, Tashiro et al., 2021, Alcaraz and Strodthoff, 2022], or neural processes [Dutordoir et al., 2023]. However, these approaches for functional data all perform diffusion modeling by employing standard finite-dimensional diffusion modeling on the discretized functions. Concurrent to the work appearing in this chapter, Biloš et al. [2022] proposed a diffusion model for temporal data, but do not take a function space perspective. As we will show in Section 3.4.3, existing approaches can be viewed as special cases within the general theoretical framework we develop.

Subsequent to our work in this chapter, Lim et al. [2023a] and Pidstrigach et al. [2023] in follow-up work proposed methodologies which are closely related and conceptually similar to our approach. As in our work, Lim et al. [2023a] and Pidstrigach et al. [2023] both perturbed the function space distribution corresponding to the data via a trace-class Gaussian

measure. Our work can be seen as extending the discrete time DDPM model [Ho et al., 2020] to function spaces, while the works of Lim et al. [2023a] and Pidstrigach et al. [2023] can be seen as extending score-matching techniques [Vincent, 2011, Song and Ermon, 2019] to function spaces. In particular, Lim et al. [2023a] developed techniques for function space score matching in discrete time, and Pidstrigach et al. [2023] developed function space score matching techniques from a continuous time perspective.

Beyond diffusion models, generative models of functions have been studied from the perspective of neural processes [Garnelo et al., 2018, Kim et al., 2019] or implicit neural representations [Dupont et al., 2022b,a]. In particular, generative models of functions based on neural operators have been proposed from a GAN approach [Rahman et al., 2022]. However, ours is the first work to combine diffusion models with neural operators.

Our approach is also broadly related to the general class of previous works that propose function-space perspectives in machine learning. In particular, such a point of view has proved useful for developing and understanding techniques used in Gaussian processes [Matthews et al., 2016, Wynne and Wild, 2022] and Bayesian deep learning [Sun et al., 2019, Wild et al., 2022, Rudner et al., 2021, Tran et al., 2022, Burt et al., 2021]. Our work extends this function-space perspective to diffusion models.

3.2 Notation and Background

We begin by setting up the notation for our problem and introducing the necessary background on Gaussian measures in Hilbert spaces, as well as their connection to the more familiar notion of Gaussian processes. In addition, we derive a closed-form expression for the KL divergence between Gaussian measures with equal covariance operators – this KL divergence plays a key role in our approach.

The notation in this chapter follows that outlined in Chapter 2, which we briefly recall here before proceeding. Let $(X, \mathcal{A}, \omega)$ be a measure space, with $X \subseteq \mathbb{R}^d$ being a subset of a Euclidean space. We use H to denote a separable Hilbert space equipped with its Borel σ-algebra. We typically take H to be a space of real-valued functions $f : X \to \mathbb{R}$ on the domain X, but our general framework is agnostic to the choice of H – see Section 3.4 for more details on this choice. The prototypical example is $X = [0, 1]$ with ω being the Lebesgue measure and $H = L^2(X, \omega)$ equipped with its usual inner product $\langle f, g \rangle_{L^2(X,\omega)} = \int_X fg \, \mathrm{d}\omega$.

We assume that we have a dataset of the form $\mathcal{D} = \{f^{(1)}, f^{(2)}, \ldots, f^{(n)}\}$, where each $f^{(j)} \in H$ is an i.i.d. draw from an unknown Borel probability measure μ_{data} on H. In practice, we typically only have noisy measurements of our functions on a finite subset of X. We let $\vec{x}^{(j)} = \{x^{(1j)}, \ldots, x^{(mj)}\} \subset X$ be a discrete subset of X with corresponding observations $\vec{y}^{(j)} = \{y^{(1j)}, \ldots, y^{(mj)}\}$, where $y^{(ij)} = f^{(j)}(x^{(ij)}) + \epsilon^{(ij)}$ is the output of the unknown jth function $f^{(j)}$ at the ith observation point and $\epsilon^{(ij)}$ represents i.i.d. observation noise. Generally, both the location $\vec{x}^{(j)}$ and number $m = m_j$ of observation points may vary across the functions in our dataset. The focus of this chapter is to develop the theory and practice behind building a diffusion generative model for sampling from the function-space probability measure μ_{data}.

3.2.1 The KL Divergence between Gaussian Measures

Our aim in this section is to study the KL divergence for Gaussian measures on Hilbert spaces. We first recall that a Gaussian measure ν over H is a distribution over random functions $f \sim \nu$ with $f \in H$ such that all one-dimensional projections for any given fixed $g \in H$ are normal, i.e. the scalar random variable $\langle g, f \rangle$ is Gaussian. See Chapter 2 for further details.

In our framework, we will perform variational inference in function space. However, one major challenge is that there is no analogue of the Lebesgue measure on infinite dimensional spaces [Eldredge, 2016], and so we must resort to a measure-theoretic definition of the KL

divergence. To that end, for arbitrary Borel probability measures μ, ν on H, we define

$$\mathsf{KL}\left[\mu \| \nu\right] = \begin{cases} \int_H \log\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right) \mathrm{d}\mu & \mu \ll \nu \\ +\infty & \text{otherwise.} \end{cases} \tag{3.1}$$

Here, $\mathrm{d}\mu / \mathrm{d}\nu : H \to \mathbb{R}$ is the Radon-Nikodym derivative of μ with respect to ν.

We now consider the special case that μ, ν are Gaussian measures on H with equal covariance operators. In this case, a version of the Feldman-Hájek Theorem gives us explicit control over the Radon-Nikodym derivative in terms of the parameters of μ and ν [Da Prato and Zabczyk, 2014, Theorem 2.23].

Theorem 13 (The Feldman-Hájek Theorem)**.**
Let $\mu = \mathcal{N}(m_1, C)$ and $\nu = \mathcal{N}(m_2, C)$ be Gaussian measures on H with equal covariance operators, and define $\Delta m = m_1 - m_2 \in H$. Then, μ and ν are equivalent (i.e. mutually absolutely continuous) if and only if $\Delta m \in C^{1/2}(H)$. In this case, for any $f \in H$, the Radon-Nikodym derivative $\mathrm{d}\nu / \mathrm{d}\mu$ is given by

$$\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(f) = \exp\left[\left\langle \Delta m, C^{-1}(f - m_2)\right\rangle - \frac{1}{2}\left\|C^{-1/2}\Delta m\right\|^2\right],$$

where C^{-1} is the pseudoinverse of C and $C^{-1/2}$ is the pseudoinverse of $C^{1/2}$.

As a straightforward consequence of the Feldman-Hájek theorem, we derive a closed-form expression for the KL divergence between Gaussian measures with equal covariance operators.

Proposition 14.
Let $\mu, \nu, \Delta m$ be defined as in Theorem 13. Then, if $\Delta m \in C^{1/2}(H)$,

$$\mathsf{KL}\left[\mu \| \nu\right] = \frac{1}{2}\left\langle \Delta m, C^{-1}\Delta m\right\rangle = \frac{1}{2}\left\|C^{-1/2}\Delta m\right\|^2. \tag{3.2}$$

Proof. As $\Delta m \in C^{1/2}(H)$, it follows from the Feldman-Hájek theorem that μ and ν are

equivalent. We now use the Radon-Nikodym expression from the Feldman-Hájek theorem to compute the KL divergence.

We have that

$$\mathsf{KL}\left[\mu\|\nu\right] = \int_H \log \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(f)\,\mathrm{d}\mu(f) \tag{3.3}$$

$$= -\frac{1}{2}\left\|C^{-1/2}\Delta m\right\|^2 + \int_H \langle C^{-1/2}\Delta m, C^{-1/2}(f-m_2)\rangle\,\mathrm{d}\mu(f). \tag{3.4}$$

We now analyze the integral term via a spectral decomposition. Let $\{(\lambda_j, e_j)\}_{j=1}^{\infty}$ be the eigenvalues and eigenvectors of C. Note that the eigenvectors of C form an orthonormal basis for H by the spectral theorem, as C is a self-adjoint compact operator. Then, we may evaluate the second integral as

$$\int_H \langle C^{-1/2}\Delta m, C^{-1/2}(f-m_2)\rangle\,\mathrm{d}\mu(f) \tag{3.5}$$

$$= \int_H \sum_{j=1}^{\infty} \langle \Delta m, e_j\rangle\langle f-m_2, e_j\rangle \lambda_j^{-1}\,\mathrm{d}\mu(f) \tag{3.6}$$

$$= \sum_{j=1}^{\infty} \lambda_j^{-1}\langle \Delta m, e_j\rangle \int_H \langle f-m_2, e_j\rangle\,\mathrm{d}\mu(f) \tag{3.7}$$

$$= \sum_{j=1}^{\infty} \lambda_j^{-1}\langle \Delta m, e_j\rangle^2 \tag{3.8}$$

$$= \langle C^{-1/2}\Delta m, C^{-1/2}\Delta m\rangle. \tag{3.9}$$

Combining this computation with the KL expression above completes the proof. □

In Section 3.3, we make use of this result in order to develop diffusion models in function space. In Section 3.4, we explore various practical methods for computing this functional KL divergence under various choices of the space H.

3.2.2 Gaussian Processes

Gaussian processes (GPs) [Williams and Rasmussen, 2006] are a popular class of models for specifying and learning distributions over functions. Formally, given a probability space $(\Omega, \mathcal{B}, \mathbb{P})$, a GP on X is a jointly measurable map $g : \Omega \times X \to \mathbb{R}$ whose finite dimensional marginal distributions are Gaussian.

In practice, a Gaussian process is typically specified by a mean function $m : X \to \mathbb{R}$ specifying $m(x) = \mathbb{E}[g(x)]$ and a kernel function $k : X \times X \to \mathbb{R}$ specifying the covariance structure of g via $k(x, x') = \mathbb{E}[(g(x) - m(x))(g(x') - m(x'))]$. We will write $g \sim \mathcal{GP}(m, k)$ for such a Gaussian process.

Gaussian processes give us a practical way of specifying Gaussian measures, as we only need to specify a mean function and a kernel. The kernel k plays an essential role in determining the sample path properties of a GP, such as continuity, differentiability, and periodicity [Williams and Rasmussen, 2006, Chapter 4]. In the case that the mean $m \in H$ is an element of H and k is chosen such that $g \in H$ with probability one, we may identify g with a GRE on H. For instance, suppose $H = L^2(X, \omega)$ and we specify a mean $m \in L^2(X, \omega)$ and kernel k with $\int_X k(x, x) \, \mathrm{d}\omega(x) < \infty$, then we may identify $\mathcal{GP}(m, k)$ with a Gaussian measure on $L^2(X, \omega)$. See Wild et al. [2022] and Section 3.4 for further details.

3.3 Diffusion Models in Function Space

Equipped with the necessary background, we now construct our diffusion generative model on H. Our construction mirrors that of DDPMs [Ho et al., 2020], with the key difference being that our diffusion process takes place in a space of infinite dimensions. We note that the constructions of Ho et al. [2020] rely heavily on properties of Gaussian densities in $\mathbb{R}^n$, and thus are not directly applicable to infinite-dimensional spaces as these spaces lack a

reference measure from which to define such densities [Eldredge, 2016]. Note further that $H = \mathbb{R}^n$ equipped with its usual inner product is a special case of our framework.

3.3.1 Forward Process

We begin by defining the *forward process*, a discrete-time Markov chain in H which iteratively perturbs our data distribution μ_{data} towards a fixed Gaussian measure $\mathcal{N}(m, C)$. In what follows, we will choose $m = 0$ for simplicity. The choice of covariance operator C is a hyperparameter which can be tuned.

We fix a finite number of timesteps $T \in \mathbb{Z}_{>0}$ and a variance schedule $\beta : \{1, 2, \ldots, T\} \to \mathbb{R}_{>0}$, where we write β_t for $\beta(t)$. For any $f_0 \in H$, we iteratively sample from the forward process via

$$f_t = \sqrt{1 - \beta_t} f_{t-1} + \sqrt{\beta_t} \xi_t \qquad t = 1, 2, \ldots, T \tag{3.10}$$

where $\xi_t \sim \mathcal{N}(0, C)$ are i.i.d. Gaussian random elements on H.

Given a fixed value of f_{t-1}, our forward process gives us conditional probability measures $\mu_{t|t-1}(- \mid f_{t-1})$. We will write μ_t for the marginal distribution on H obtained at time step t from this process, i.e.

$$\mu_t(-) = \int_H \mu_{t|t-1}(- \mid f_{t-1}) \, \mathrm{d}\mu_{t-1}(f_{t-1}) \tag{3.11}$$

where $\mu_0 = \mu_{\text{data}}$. The value of T and the variance schedule β are chosen such that the final distribution is approximately equal to our specified Gaussian measure, i.e. $\mu_T \approx \mathcal{N}(0, C)$.

In the following proposition, we derive expressions for several distributions related to our forward process.

Proposition 15.

Let $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$. *For the forward process defined in Equation* (3.10) *with* $m = 0$ *and fixed values of* f_0, f_{t-1}*:*

$$\mu_{t|t-1}(- \mid f_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} f_{t-1}, \beta_t C) \tag{3.12}$$

$$\mu_{t|0}(- \mid f_0) = \mathcal{N}(\sqrt{\alpha_t} f_0, (1-\alpha_t) C). \tag{3.13}$$

Proof. The first claim is a special case of Lemma (1). For the second claim, we proceed by induction on t. The case $t = 1$ is clear from Lemma (1). Now, suppose

$$u_{t-1} \mid u_0 \sim \mathcal{N}(\sqrt{\alpha_{t-1}} f_0, (1-\alpha_{t-1}) C). \tag{3.14}$$

By the definition of the forward process and our inductive assumption, we have that $f_t = \sqrt{1-\beta_t} f_{t-1} + \sqrt{\beta_t}\xi_t$ is the sum of two independent GREs: the first is

$$\sqrt{1-\beta_t} f_{t-1} \sim \mathcal{N}\left(\sqrt{\alpha_t} f_0, (1-\beta_t)(1-\alpha_{t-1}) C\right) \tag{3.15}$$

and the second is $\sqrt{\beta_t}\xi_t \sim \mathcal{N}(0, \beta_t C)$. By Lemma (2), we obtain the result, as

$$(1-\beta_t)(1-\alpha_{t-1}) + \beta_t = 1 - (1-\beta_t)\alpha_{t-1} = 1 - \alpha_t. \tag{3.16}$$

□

3.3.2 Reverse Process and Loss

Our generative model is then obtained by reversing the forward process, where we iteratively perturb the Gaussian measure $\mathcal{N}(0, C)$ towards the data distribution μ_0.

More specifically, to generate samples from our data distribution, we would like sample $f_T \sim \mathcal{N}(0, C)$ and iteratively sample $f_{t-1} \sim \mu_{t-1|t}(- \mid f_t)$ from the time-reversal of our forward process for $t = T-1, \ldots, 1$. However, while the posterior probability measure $\mu_{t-1|t}(- \mid f_t)$ is well-defined[2], it is intractable.

Most notably, using Bayes' rule here would require that the family of measures $\mu_{t|t-1}(- \mid f_{t-1})$ be simultaneously dominated by some fixed reference measure on H for every choice of f_{t-1}. As these measures are Gaussian, the Feldman-Hájek theorem tells us that this is not possible. Even if such technical difficulties were overcome (e.g. as in the Euclidean setting), computing Bayes' rule here would require computing an intractable normalization constant.

We instead take a variational approach, and approximate the posterior measures with a variational family of measures on H parametrized by $\theta \in \mathbb{R}^p$. In particular, we set $\nu_T^\theta = \mathcal{N}(0, C)$ and we approximate $\mu_{t-1|t}(- \mid f_t)$ by the Gaussian measure

$$\nu_{t-1|t}^\theta(- \mid f_t) = \mathcal{N}\left(m_t^\theta(f_t), C_t^\theta(f_t)\right). \tag{3.17}$$

Here, $m_t^\theta(f_t) = m_t^\theta(- \mid f_t) \in H$ is shorthand for a mean function in H and $C_t^\theta(f_t) = C_t^\theta(- \mid f_t) : H \to H$ is shorthand for a covariance operator. That is, the mean function and covariance operators depend on parameters θ as well as the timestep t and function $f_t \in H$.

Although the reverse-time measures are intractable, the following proposition states that the reverse-time measures are tractable when conditioned on a starting function $f_0 \in H$.

[2]This is because we assume H is separable, which implies that H is a Polish space. See Ghosal and Van der Vaart [2017, Chapter 1].

Proposition 16.

Let α_t be defined as in Proposition (15), and consider fixed values of $f_0, f_t \in H$. For $t = 2, 3, \ldots, T$, let $\widetilde{\beta}_t = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$ and let $\widetilde{m}_t(f_t, f_0) = \widetilde{m}_t(- \mid f_t, f_0) \in H$ be defined by

$$\widetilde{m}_t(f_t, f_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} f_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t} f_t. \tag{3.18}$$

Then, $\mu_{t-1|t,0}(- \mid f_t, f_0) = \mathcal{N}(\widetilde{m}_t(f_t, f_0), \widetilde{\beta}_t C)$.

Proof. By Proposition (15) and Lemma (1), we may write

$$f_{t-1} = \sqrt{\alpha_{t-1}} f_0 + \sqrt{1-\alpha_{t-1}}\xi \qquad \xi \sim \mathcal{N}(0, C) \tag{3.19}$$

and by construction we have

$$f_t = \sqrt{1-\beta_t} f_{t-1} + \sqrt{\beta_t}\xi' \qquad \xi' \sim \mathcal{N}(0, C) \tag{3.20}$$

where $\xi, \xi' \sim \mathcal{N}(0, C)$ are independent GREs. Our strategy is to manipulate these expressions to obtain a reparametrized expression for f_{t-1}. By Equation (3.19),

$$\beta_t \sqrt{\alpha_{t-1}} f_0 = \beta_t \left[f_{t-1} - \sqrt{1-\gamma_{t-1}}\xi \right], \tag{3.21}$$

and similarly by Equation (3.20),

$$(1-\alpha_{t-1})\sqrt{1-\beta_t} f_t = (1-\alpha_{t-1}) \left[(1-\beta_t) f_{t-1} + \sqrt{\beta_t}\sqrt{1-\beta_t}\xi' \right]. \tag{3.22}$$

Upon summing Equations (3.21)-(3.22) and isolating the f_{t-1} terms,

$$\begin{aligned} \left(\beta_t + (1-\alpha_{t-1})(1-\beta_t)\right) f_{t-1} = \beta_t \sqrt{\alpha_{t-1}} f_0 + (1-\alpha_{t-1})\sqrt{1-\beta_t} f_t \\ + \beta_t \sqrt{1-\alpha_{t-1}}\xi - (1-\alpha_{t-1})\sqrt{\beta_t}\sqrt{1-\beta_t}\xi'. \end{aligned} \tag{3.23}$$

On the LHS, we have

$$(\beta_t + (1-\alpha_{t-1})(1-\beta_t))f_{t-1} = (\beta_t + (1-\beta_t) - (1-\beta_t)(\alpha_{t-1}))f_{t-1} = (1-\alpha_t)f_{t-1} \tag{3.24}$$

thereby allowing us to obtain

$$f_{t-1} = \frac{\beta_t\sqrt{\alpha_{t-1}}}{1-\alpha_t}f_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}f_t + \frac{\beta_t\sqrt{1-\alpha_{t-1}}}{1-\alpha_t}\xi - \frac{(1-\alpha_{t-1})\sqrt{\beta_t}\sqrt{1-\beta_t}}{1-\alpha_t}\xi'. \tag{3.25}$$

We now analyze the noise terms (i.e. only those terms depending on ξ, ξ'). By Lemmas (1)-(2) and the independence of ξ, ξ', the sum of the noise terms follows a mean zero Gaussian measure with covariance

$$\left(\frac{\beta_t\sqrt{1-\alpha_{t-1}}}{1-\alpha_t}\right)^2 + \left(\frac{(1-\alpha_{t-1})\sqrt{\beta_t}\sqrt{1-\beta_t})}{1-\alpha_t}\right)^2$$
$$= \left(\frac{\beta_t(1-\alpha_{t-1})}{1-\alpha_t}\right)\left(\frac{\beta_t + (1-\beta_t)(1-\alpha_{t-1})}{1-\alpha_t}\right) = \frac{\beta_t(1-\alpha_{t-1})}{1-\alpha_t}$$

where the last line follows from the calculation in Equation (3.24). Thus, we see that $f_{t-1} \mid f_t, f_0$ follows a Gaussian measure with the claimed mean and covariance. □

We now tie our function-space Markov chain back to our observed data in order to obtain a loss function. Recall that our observations $\vec{y} \subset \mathbb{R}$ are assumed to be a vector of noisy observations of a function $f_0 \in H$ at some finite collection of points $\vec{x} \subset X$. We thus set the likelihood of our observed data to be $q^\theta(\vec{y} \mid \vec{x}, f_0) = \mathcal{N}(\vec{y}; f_0(\vec{x}), \sigma^2 I)$ where $\sigma^2 \in \mathbb{R}_{\geq 0}$ is some fixed constant. Note that q^θ is a Gaussian density on a finite dimensional space.

In the following proposition, we obtain a variational lower bound on the log-likelihood of our observations. This will serve as our loss function, which we seek to maximize over θ. Although this lower bound is analogous to the standard DDPM lower bound [Ho et al., 2020],

the proof is non-trivial as we must work directly with the underlying probability measures rather than their densities.

Proposition 17.

The marginal likelihood of $\vec{y}$ given $\vec{x}$ is lower bounded by

$$\log\ q^\theta(\vec{y} \mid \vec{x}) \geq \tag{3.26}$$
$$\mathbb{E}_\mu \left[\log q(\vec{y} \mid \vec{x}, f_0) - \mathsf{KL}\left[\mu_T(- \mid \vec{x}, \vec{y}) \| \nu_T^\theta(-)\right] - \sum_{t=1}^{T} \mathsf{KL}\left[\mu_{t-1|t}(- \mid f_t, \vec{x}, \vec{y}) \| \nu_{t-1|t}^\theta(- \mid f_t))\right] \right].$$

Proof. First, we apply the usual functional ELBO [Wild et al., 2022, Matthews et al., 2016, Sun et al., 2019], treating $f_{0:T}$ as latent variables and using the assumption that the reverse-time chain is Markov to obtain

$$\log q^\theta(\vec{y} \mid \vec{x}) \geq \mathbb{E}_\mu \left[\log q^\theta(\vec{y} \mid \vec{x}, f_0)\right] - \mathsf{KL}\left[\mu(\,\mathrm{d}f_{0:T} \mid \vec{x}, \vec{y}) \| \nu^\theta(\,\mathrm{d}f_{0:T})\right]. \tag{3.27}$$

By the chain rule for KL divergences [Dupuis and Ellis, 2011], we may condition on f_T to obtain

$$\begin{aligned} \log q^\theta(\vec{y} \mid \vec{x}) \geq & \mathbb{E}_\mu \left[\log q^\theta(\vec{y} \mid \vec{x}, f_0)\right] - \mathsf{KL}\left[\mu_T(\,\mathrm{d}f_T \mid \vec{x}, \vec{y}) \| \nu^\theta(\,\mathrm{d}f_T)\right] \\ & - \mathbb{E}_\mu \left[\mathsf{KL}\left[\mu(\,\mathrm{d}f_{0:T-1} \mid \vec{x}, \vec{y}, f_T) \| \nu^\theta(\,\mathrm{d}f_{0:T} \mid f_T)\right]\right]. \end{aligned} \tag{3.28}$$

Repeatedly applying the KL divergence chain rule to condition on $f_{T-1}, f_{T-2}, \ldots, f_1$ and using the Markov assumption yields

$$\log q^\theta(\vec{y} \mid \vec{x}) \geq \mathbb{E}_\mu \left[\log q^\theta(\vec{y} \mid \vec{x}, f_0)\right] - \mathsf{KL}\left[\mu_T(\,\mathrm{d}f_T \mid \vec{x}, \vec{y}) \| \nu^\theta(\,\mathrm{d}f_T)\right] \tag{3.29}$$
$$- \sum_{t=1}^{T} \mathbb{E}_\mu \left[\mathsf{KL}\left[\mu(\,\mathrm{d}f_{t-1} \mid f_t, \vec{x}, \vec{y}) \| \nu^\theta(\,\mathrm{d}f_{t-1} \mid f_t)\right]\right]. \tag{3.30}$$

□

Since we assume ν_T^θ has no trainable parameters, we may ignore the term $\mathsf{KL}\left[\mu_T(- \mid \vec{x}, \vec{y}) \| \nu_T^\theta(-)\right]$ during training.

Mean and Covariance Parametrization We now make several further choices for our variational family. First, we analyze the terms

$$L_{t-1} = \mathsf{KL}\left[\mu_{t-1|t}(- \mid f_t, f_0) \| \nu_{t-1|t}^\theta(- \mid f_t))\right]. \tag{3.31}$$

Note that the first measure $\mu_{t-1|t}(- \mid f_t, f_0)$ is Gaussian by Proposition (16), and the second measure $\nu_{t-1|t}^\theta(- \mid f_t)$ is Gaussian by assumption. A more general form of the Feldman-Hájek theorem (see Chapter 2) places strict requirements on the corresponding covariance operators in order to obtain a finite KL divergence. In particular, the term L_{t-1} will be infinite if

$$\widetilde{\beta}_t^{-1}\left(C^{-1/2}C_t^\theta(f_t)^{1/2}\right)\left(C^{-1/2}C_t^\theta(f_t)^{1/2}\right)^* - I \tag{3.32}$$

is not a Hilbert-Schmidt operator on the closure of $C^{1/2}(H)$. For instance, even the seemingly innocuous choice of $C_t^\theta(f_t) = \alpha\widetilde{\beta}_t C$ for any non-negative $\alpha \neq 1$ will result in an infinite KL divergence. Thus, motivated by necessity, we will choose $C_t^\theta(f_t) = \widetilde{\beta}_t C$.

Under this choice of $C_t^\theta(f_t)$, a consequence of Propositions (14) and (16) is that

$$L_{t-1} = \frac{1}{2\widetilde{\beta}_t}\left\|C^{-1/2}(\widetilde{m}_t(f_t, f_0) - m_t^\theta(u_t))\right\|^2. \tag{3.33}$$

Similar to DDPM [Ho et al., 2020], we further choose to parametrize the mean function via

$$m_t^\theta(f_t) = \frac{1}{\sqrt{1-\beta_t}}\left(f_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\xi_t^\theta(f_t)\right) \tag{3.34}$$

where $\xi_t^\theta(f_t) \in H$ is the output of a model parametrized by θ which takes in (t, f_t) as inputs and has function-valued outputs. In other words, our model is a parametrized mapping $\xi^\theta : \{1, 2, \ldots, T\} \times H \to H$ specified via $(t, f_t) \mapsto \xi_t^\theta(- \mid f_t)$. Under this choice, we have that

$$L_{t-1} = \lambda_t \left\| C^{-1/2}(\xi_t - \xi_t^\theta(f_t)) \right\|^2 \tag{3.35}$$

where $\lambda_t = \beta_t^2/(2\widetilde{\beta}_t(1-\beta_t)(1-\alpha_t)) \in \mathbb{R}$ is a time-dependent constant. In light of Proposition (14), we see that L_{t-1} is (up to a multiplicative constant) the KL divergence between two Gaussian measures on H having covariance operators C and respective means $\xi_t, \xi_t^\theta(f_t)$. As is standard in diffusion generative modeling, we drop the constant λ_t when training in order to obtain a re-weighted variational lower bound [Ho et al., 2020] for improved quality.

In Section 3.5, we provide a practical instantiation of the mapping ξ_t^θ via neural operators [Li et al., 2021, 2020, Kovachki et al., 2021].

Following our work, Lim et al. [2023a] noted that the parametrization of the loss given in Equations 3.34 and 3.35 can result in an infinite quantity when the dimension of H is infinite. However, it is straightforward to remedy this by considering an alternative parametrization, where the model directly predicts a rescaled version of f_0 rather than predicting ξ_t, e.g., see Appendix E and Appendix I of Lim et al. [2023a] for additional details. In our experiments in this paper we used the parametrization given in Equations 3.34 and 3.35, and note that the corresponding quantities are only infinite in the limit corresponding to a discretization size of zero.

3.4 Function Spaces and KL Approximations

We have thus far described our framework in terms of abstract Gaussian measures on Hilbert spaces. We can obtain a concrete instantiation of our framework by choosing an appropriate space of functions to work on, as well as a choice of Gaussian measure which specifies our forward process.

In this section, we explore two choices for H: the space of square-integrable functions $L^2(X, \omega)$ and the Sobolev spaces $H^k(X, \omega) = W^{k,2}(X, \omega)$. We derive practical methods for estimating the KL divergence between Gaussian measures in these spaces, which is necessary for evaluating the terms in our loss function given in Equation (3.35).

To compute the functional KL divergence in Proposition (14), we derive discrete approximations of both the inverse covariance operator C^{-1} and the associated inner product. Suppose that m_1 and m_2 are known on a common discretization $\vec{x} = \{x^{(1)}, \ldots, x^{(n)}\} \subset X$ which is drawn from the measure ω on X. For any function $f : X \to \mathbb{R}$, we write $f(\vec{x}) \in \mathbb{R}^n$ to represent the vector corresponding to evaluating f at the points contained in $\vec{x}$. We assume further that our Gaussian measure $\xi \sim \mathcal{GP}(0, k)$ is specified by a mean-zero Gaussian process with kernel k, with appropriate restrictions on k such that $\xi \in H$ (see Section 3.2.2). In Section 3.5.4, we explore estimating these KL divergences with spectral methods, but find that it is sensitive to the discretization size, even when the eigenfunctions are analytically known.

3.4.1 Square-Integrable Functions

We first consider the space $H = L^2(X, \omega)$ of measurable, square-integrable functions $f : X \to \mathbb{R}$ equipped with the inner product $\langle f, g \rangle_{L^2(X,\omega)} = \int_X fg \, \mathrm{d}\omega$. For many applications, this is a natural choice of function space as square integrability is a relatively weak

assumption. Moreover, $p = 2$ is the unique choice such that the Banach space $L^p(X, \omega)$ is also a Hilbert space, and the associated inner product structure is a useful tool for performing calculations.

In $L^2(X, \omega)$, the covariance operator associated with our kernel function k can be explicitly described via

$$[Cg](x) = \int_X k(x, x')g(x') \, \mathrm{d}\omega(x') \qquad \forall g \in H. \tag{3.36}$$

Indeed, recall that that for a Gaussian measure $f \sim \mu_f = \mathcal{N}(m, C)$ on a separable Hilbert space H, the covariance operator $C : H \to H$ is defined via

$$Cg = \int_H \langle g, f \rangle f \, \mathrm{d}\mu_f - \langle g, m \rangle m \qquad \forall g \in H. \tag{3.37}$$

Thus, Equation (3.36) can be derived from Equation (3.37) via

$$\begin{aligned}
[Cg](x) &= \int_H \langle g, f \rangle_{L^2(X,\omega)} f(x) \, \mathrm{d}\mu_f - \langle g, m \rangle_{L^2(X,\omega)} m(x) && (3.38)\\
&= \int_H \left[\int_X g(x')f(x') \, \mathrm{d}\omega(x') \right] f(x) \, \mathrm{d}\mu_f - \langle g, m \rangle_{L^2(X,\omega)} m(x) && (3.39)\\
&= \int_X g(x') \left[\int_H f(x)f(x') \, \mathrm{d}\mu_f \right] \mathrm{d}\omega(x') - \langle g, m \rangle_{L^2(X,\omega)} m(x) && (3.40)\\
&= \int_X g(x') \left[k(x, x') + m(x)m(x') \right] \mathrm{d}\omega(x') - \int_X g(x')m(x')m(x) \, \mathrm{d}\omega(x') && (3.41)\\
&= \int_X g(x')k(x, x') \, \mathrm{d}\omega(x') && (3.42)
\end{aligned}$$

where we apply Fubini's theorem in the third equality.

To discretize this, let $K_{\vec{x}\vec{x}} \in \mathbb{R}^{n \times n}$ be the covariance matrix specified by k and evaluated on $\vec{x}$, i.e. the (i, j)th entry of $K_{\vec{x}\vec{x}}$ is given by $k(x_i, x_j)$. Then, upon replacing ω with the

empirical measure specified by $\vec{x}$, we have $[Cg](\vec{x}) \approx n^{-1} K_{\vec{x}\vec{x}} g(\vec{x}) \in \mathbb{R}^n$, so that the (scaled) covariance matrix K is a discrete approximation of the covariance operator C which may inverted. Replacing ω once more with the empirical measure specified by $\vec{x}$, we then have

$$\mathsf{KL}\left[\mathcal{N}(m_1, C) \| \mathcal{N}(m_2, C)\right] \approx \frac{1}{2} \Delta m(\vec{x})^T K_{\vec{x}\vec{x}}^{-1} \Delta m(\vec{x}). \tag{3.43}$$

Interestingly, this is precisely the KL divergence between two finite-dimensional Gaussians with equal covariance matrices $K_{\vec{x}\vec{x}}$ and means $m_1(\vec{x}), m_2(\vec{x})$.

Furthermore, we note that Sun et al. [2019] prove that the KL divergence between two stochastic processes is the supremum of the KL divergences between their finite-dimensional marginals. Our approximation in Equation (3.43) is increasing under refinements of the observation set $\vec{x}$, and thus is a lower bound on the true KL divergence.

Proposition 18.

Equation (3.43) *is strictly increasing under refinements of the observation set* $\vec{x}$*. In particular, if* $\vec{z} \subset \vec{x}$*, then*

$$\Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) \leq \Delta m(\vec{x})^T K_{\vec{x}\vec{x}}^{-1} \Delta m(\vec{x}). \tag{3.44}$$

Proof. Set $\vec{z} = \{z^{(1)}, \ldots, z^{(n)}\} \subset X$. If suffices to check the case that $\vec{x} = x \cup \vec{z}$ is increased by a single point $x \in X$.

Let $K_{\vec{z}\vec{z}} \in \mathbb{R}^{n \times n}$ be the covariance matrix corresponding to $\vec{z}$, and let $K_{\vec{x}\vec{x}} \in \mathbb{R}^{(n+1)\times(n+1)}$ be the covariance matrix corresponding to $\vec{x}$, i.e. in both cases the covariance matrix is given by evaluating the kernel k at all combinations of points in $\vec{z}$ or $\vec{x}$. Let $k_{\vec{z}}(x) \in \mathbb{R}^n$ be the covariance between the points of $\vec{z}$ and our new point x. Lastly, let $\Delta m(\vec{z}) \in \mathbb{R}^n$ be any vector and be $\Delta m(x) \in \mathbb{R}$ any scalar. We will write $\Delta m(\vec{x}) = [\Delta m(\vec{z}), \Delta m(x)]^T \in \mathbb{R}^{n+1}$ for the vector extending $\Delta m(\vec{z})$ by the single entry $\Delta m(x)$.

Then, we have that

$$K_{\vec{x}\vec{x}} = \begin{bmatrix} K_{\vec{z}\vec{z}} & k_{\vec{z}}(x) \\ k_{\vec{z}}(x)^T & k(x,x) \end{bmatrix}, \tag{3.45}$$

i.e. the extended covariance matrix corresponding to $\vec{x}$ can be written as a block matrix containing the covariance matrix for $\vec{z}$. Our goal is to show that

$$\Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) \leq \Delta m(\vec{x})^T K_{\vec{x}\vec{x}}^{-1} \Delta(\vec{x}). \tag{3.46}$$

Using the block matrix inversion formula (see e.g. Williams and Rasmussen [2006, Appendix A.3]), we may express $K_{\vec{x}\vec{x}}^{-1}$ as

$$K_{\vec{x}\vec{x}}^{-1} = \begin{bmatrix} K_{\vec{z}\vec{z}}^{-1} + K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} & -K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M \\ -M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} & M \end{bmatrix} \tag{3.47}$$

where

$$M = \left(k(x,x) - k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x)\right)^{-1} \in \mathbb{R}. \tag{3.48}$$

Note that M is exactly the posterior variance at $x \in X$ of a GP with covariance function k [Williams and Rasmussen, 2006, Eqn. 2.26]. In particular, we must have $M \geq 0$.

We now proceed to directly compute the quadratic form on the right-hand side of Equation

(3.46). We have:

$$[\Delta m(\vec{z}), \Delta m(x)]\, K_{\vec{x}\vec{x}}^{-1} \begin{bmatrix} \Delta m(\vec{z}) \\ \Delta m(x) \end{bmatrix} \tag{3.49}$$

$$= \left\langle \begin{bmatrix} \Delta m(\vec{z})^T \left(K_{\vec{z}\vec{z}}^{-1} + K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} \right) - \Delta m(x) M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} \\ -\Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M + \Delta m(x) M \end{bmatrix}, \begin{bmatrix} \Delta m(\vec{z}) \\ \Delta m(x) \end{bmatrix} \right\rangle \tag{3.50}$$

$$= \Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) + \Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) \tag{3.51}$$

$$- \Delta m(x) M k_{\vec{z}}(x)^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) - \Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) M \Delta m(x) + \Delta m(x)^2 M \tag{3.52}$$

$$= \Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} \Delta m(\vec{z}) + M \left(\Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) - \Delta m(x) \right)^2. \tag{3.53}$$

We now plug Equation (3.53) back into Equation (3.46). Noting the first term in (3.53) is precisely the LHS of (3.46), we only need to check

$$0 \leq M \left(\Delta m(\vec{z})^T K_{\vec{z}\vec{z}}^{-1} k_{\vec{z}}(x) - \Delta m(x) \right)^2. \tag{3.54}$$

However, note that we already observed that $M \geq 0$, and the other term is the square of a scalar, whence it is positive. □

3.4.2 Sobolev Spaces

A second choice of function spaces that have many practical applications are the Sobolev spaces $H^k(X, \omega)$ consisting of functions in $L^2(X, \omega)$ whose mixed partial derivatives of order at most k exist (in a weak sense) and are also in $L^2(X, \omega)$ [Evans, 2010, Chapter 5]. Of particular interest is the setting where $X \subseteq \mathbb{R}$ and $k = 1$, where the inner product is given

by $\langle f, g\rangle_{H^1(X,\omega)} = \langle f, g\rangle_{L^2(X,\omega)} + \langle \partial_x f, \partial_x g\rangle_{L^2(X,\omega)}$.

When the Gaussian process associated with the kernel function k lies in H^1 with probability one, the corresponding covariance operator can be expressed as

$$[Cg](x) = \int_X k(x, x')\,\mathrm{d}\omega(x') + \int_X \left[\partial_{x'} k(x, x')\right]\left[\partial_{x'} g(x')\right]\,\mathrm{d}\omega(x'). \tag{3.55}$$

To derive this, note that the mean element is not dependent on the inner product – it is merely an arbitrary element $m \in H$. Now, from Equation (3.37),

$$\begin{aligned}
[Cg](x) &= \int_H \langle g, f\rangle_{H^1(X,\omega)} f(x)\,\mathrm{d}\mu_f - \langle g, m\rangle_{H^1(X,\omega)} && (3.56)\\
&= \int_H \left[\langle g, f\rangle_{L^2(X,\omega)} + \langle \partial_{x'} g(x'), \partial_{x'} f(x')\rangle_{L^2(X,\omega)}\right] f(x)\,\mathrm{d}\mu_f - \langle g, m\rangle_{H^1(X,\omega)} m(x) && (3.57)\\
&= \int_X k(x, x') g(x')\,\mathrm{d}\omega(x') + \int_H \langle \partial_{x'} g(x'), \partial_{x'} f(x')\rangle_{L^2(X,\omega)} f(x)\,\mathrm{d}\mu_f && (3.58)\\
&\qquad - \langle \partial_{x'} g(x'), \partial_{x'} m(x')\rangle_{L^2(X,\omega)} m(x)\\
&= \int_X k(x, x') g(x')\,\mathrm{d}\omega(x') + \int_X \partial_{x'} g(x') \mathbb{E}[f(x)\partial_{x'} f(x')]\,\mathrm{d}\omega(x') && (3.59)\\
&\qquad - \langle \partial_{x'} g(x'), \partial_{x'} m(x')\rangle_{L^2(X,\omega)} m(x)\\
&= \int_X k(x, x') g(x')\,\mathrm{d}\omega(x') + \int_X \partial_{x'} g(x') \left(\partial_{x'} k(x, x') + m(x)\partial_{x'} m(x')\right)\,\mathrm{d}\omega(x') && (3.60)\\
&\qquad - \langle \partial_{x'} g(x'), \partial_{x'} m(x')\rangle_{L^2(X,\omega)} m(x)\\
&= \int_X k(x, x') g(x')\,\mathrm{d}\mu(x') + \int_X \partial_{x'} k(x, x') \partial_{x'} g(x, x')\,\mathrm{d}\omega(x'). && (3.61)
\end{aligned}$$

The third equality follows from the corresponding $L^2(X, \omega)$ calculation. The fifth equality follows from the fact that if $f \sim \mathcal{GP}(m, k)$ is differentiable with probability one, then $\partial_{x'} f$ is also a Gaussian process with mean $\partial_{x'} m$ [Williams and Rasmussen, 2006, Papoulis and Pillai, 2002], and moreover the covariance between f and its derivative is given by differentiating

the kernel:

$$\mathrm{Cov}(f(x), \partial_{x'} f(x')) = \mathbb{E}\left[(f(x) - m(x))\left(\partial_{x'} F(x') - \partial_{x'} m(x')\right)\right] = \partial_{x'} k(x, x'). \tag{3.62}$$

See e.g. Williams and Rasmussen [2006, Chapter 9.4].

Our discretization in this setting follows closely that of our techniques for the space $L^2(X, \omega)$, with the additional necessity of employing a discrete differential operator. To that end, let $D \in \mathbb{R}^{n \times n}$ be any discrete approximation to the first-order differentiation operator. In practice we use a discretization based on finite-difference equations. Let $K'_{\vec{x}\vec{x}} \in \mathbb{R}^{n \times n}$ be the covariance matrix corresponding to the differeniated kernel $\partial_{x'} k(x, x')$. That is, the (i, j)th entry of $K'_{\vec{x}\vec{x}}$ is given by $\frac{\partial}{\partial x'} k(x_i, x_j)$. Then, the covariance operator C can be discretized via $[Cg](\vec{x}) \approx n^{-1}\left[K_{\vec{x}\vec{x}} + K'_{\vec{x}\vec{x}} D\right] g(\vec{x}) \in \mathbb{R}^n$, and moreover,

$$\mathsf{KL}\left[\mathcal{N}(m_1, C) \| \mathcal{N}(m_2, C)\right] \approx \frac{1}{2} \Delta m(\vec{x})^T \left[I + D^T D\right] \left[K_{\vec{x}\vec{x}} + K'_{\vec{x}\vec{x}} D\right]^{-1} \Delta m(\vec{x}).$$

Although the covariance operator C is guaranteed to be positive semidefinite in theory, discretizing this operator often results in a non-PSD matrix approximation which may cause training to diverge. In practice, we project the matrix $[I + D^T D][K_{\vec{x}\vec{x}} + K'_{\vec{x}\vec{x}} D]^{-1}$ to the nearest symmetric PSD matrix (in terms of the Frobenius norm) [Higham, 1988, Cheng and Higham, 1998]. In particular, we apply the methods of Cheng and Higham [1998], Higham [1988] to find

$$\widetilde{A} = \arg\min_B \left\{ ||B - A||_F : B \text{ is symmetric, PSD} \right\} \tag{3.63}$$

i.e. the closest symmetric PSD matrix to A in terms of the Frobenius norm. This has a unique solution, which can be computed in a straightforward manner. We briefly review this method here for the sake of completeness. First, set $C = \frac{1}{2}(A + A^T)$ to be the symmetric part

of A. Then, compute the usual spectral decomposition $C = Q\text{diag}(\lambda_i)Q^T$ where Q is a matrix containing the eigenvectors of C with corresponding eigenvalues $\{\lambda_i\}$. Set $\tau_i = \max(0, \lambda_i)$. Then, $\widetilde{A} = Q\text{diag}(\tau_i)Q^T$ is our desired projection.

3.4.3 Existing Methods in Terms of Our Theory

In terms of our methodology, existing methods [Kong et al., 2021, Chen et al., 2021, Tashiro et al., 2021, Dutordoir et al., 2023] can be viewed as operating in the space $H = L^2(X, \omega)$, with the discretization employed in Equation (3.43). In all of these methods, the forward process is defined via a white noise prior. However, such a prior can *not* be seen as a Gaussian measure. In particular, the white noise process is not jointly measurable [Kallianpur, 2013, Example 1.2.5], and thus one is unable to consider the corresponding sample paths as elements of some function space. A GRE corresponding to this prior would have infinite variance, as the corresponding covariance operator would be the identity operator. Nonetheless, despite these foundational concerns, existing methods show strong empirical performance. Explaining this performance from a functional point of view, for example through the theory of generalized functions [Grubb, 2008], is an interesting challenge for future work.

3.5 Experiments

In this section, we perform several experiments in order to illustrate how our theoretical framework can be implemented as a practical estimation methodology. In all experiments, we parametrize $\xi_t^\theta(f_t)$ via a graph neural operator (GNO) [Li et al., 2020, Kovachki et al., 2021]. See Appendix A.1 for our model configurations and hyperparameter settings. Our models are trained by minimizing the reweighted negative ELBO as described in Section 3.3. In all plots, our functional diffusion model is denoted *FuncDiff*. Pseudocode and additional

details for all of our algorithms is available in Appendix A.3.[3]

A key property of the GNO is the ability to condition on arbitrary discretizations of X. This allows us to train our models on functions that are observed at different points, as well as to condition on arbitrary function observations when performing conditional generation. Moreover, as neural operators parametrize mappings between function spaces, we are able to query our model at arbitrary input locations. Thus, our model is not tied to any particular discretization.

Datasets We use both a synthetic and a real-world dataset to illustrate our approach, with results on additional real-world datasets in Appendix A.4. Our synthetic dataset is a mixture of Gaussian processes ($MoGP$) with a squared-exponential kernel with variance $\sigma^2 = 0.4$ and length scale $\ell = 0.1$, where the first mixture component has mean $m_1 = 10x - 5$ and the second has mean $m_2 = -10x + 5$. These functions are observed on a uniform discretization of $[0, 1] \subset \mathbb{R}$. We use 64 observation points unless otherwise specified. Our real-world dataset ($AEMET$) is a well-known dataset in the functional data analysis literature. This dataset consists of 73 curves, where each curve is the mean daily temperature at a particular Spanish weather station, so that each curve has a total of 365 discrete observations [Febrero-Bande and de la Fuente, 2012]. See Figure 3.1 for an illustration of these datasets.

3.5.1 Unconditional Generation

In this experiment, we sample curves unconditionally from our trained model. In Figure 3.1, the generated curves closely match the training data in terms of perceptual qualities. We additionally compute the pointwise mean, pointwise variance, and mean autocorrelation of both the real and generated curves. The summary statistics of the generated data closely

[3]Code for all of our experiments is available at `https://github.com/GavinKerrigan/functional_diffusion`

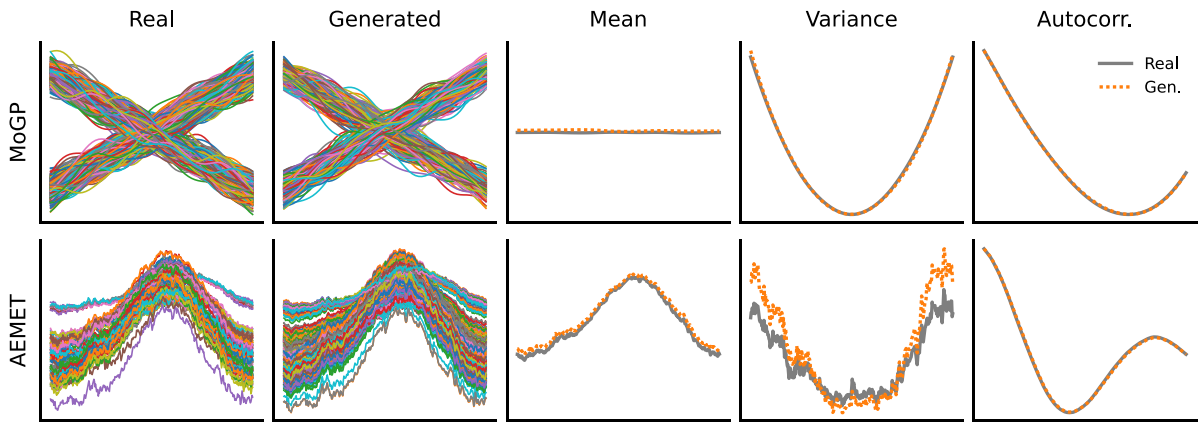


Figure 3.1: Unconditional function generation on a synthetic (MoGP) and real-world (AEMET) dataset. For each dataset, a GNO model was trained on the plotted functions (first column), and a total of 500 functions were sampled from the model (second column). The generated curves closely match the training curves in both perceptual quality and pointwise statistics.

match those of the real data, indicating that the model has successfully learned to sample from the functional distribution. See Appendix A.4 for a comparison to a simple baseline based on functional PCA [Ramsay and Silverman, 2008, Chapter 6] and additional datasets.

3.5.2 Conditional Generation

Our proposed approach for conditional generation is an extension of the ILVR method [Choi et al., 2021] to functional data. This method works by perturbing conditioning information via the forward process, and during generation we set the values of the generated function at the conditioning locations to these perturbed values. In particular note that we are able to condition a pre-trained unconditional model on arbitrary function observations. Thus, this method may potentially be applied to a wide array of tasks, such as extrapolation, upsampling, or data imputation.

In Figure 3.2, we demonstrate this by conditioning our generation on a known segment of the function. We see that our method is able to leverage the learned functional distribution

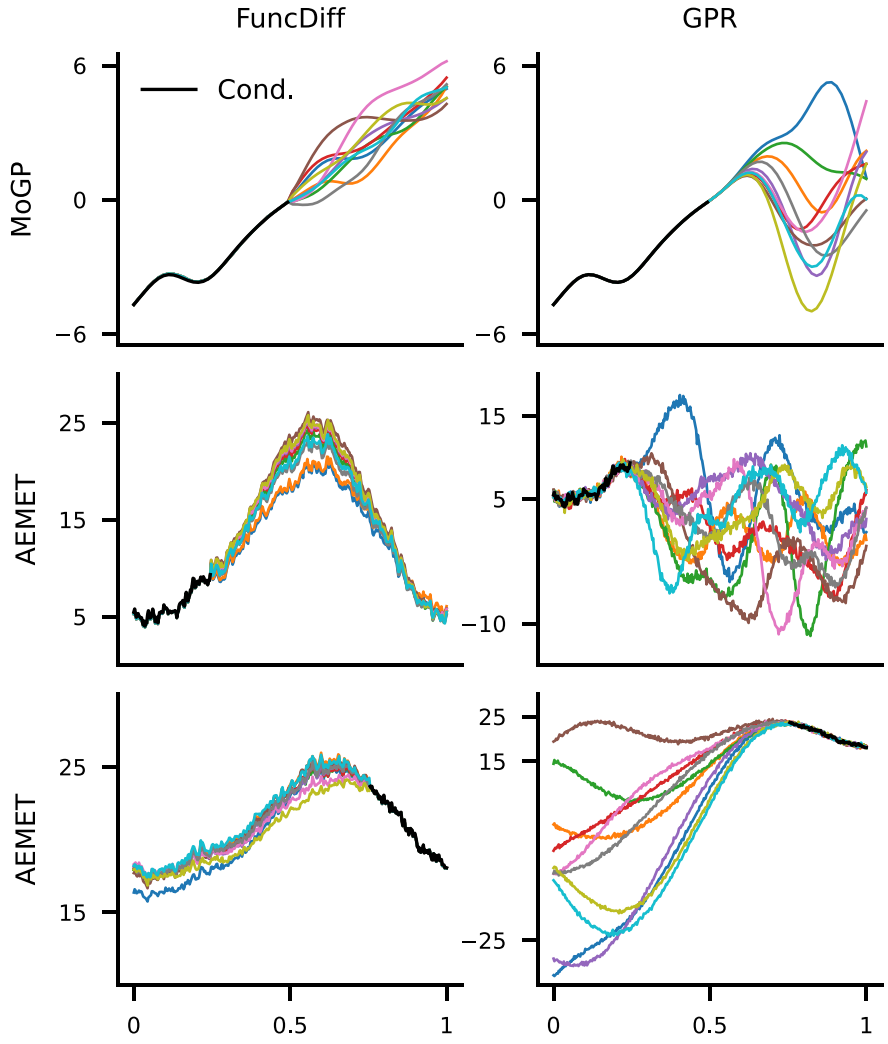


Figure 3.2: Conditional samples of our model (FuncDiff) are compared against Gaussian process regression (GPR). In each plot, both models are conditioned on the black curves.

in order to accurately extrapolate the given conditioning information. We compare to a Gaussian process regression (*GPR*) baseline, where we fit a Gaussian process only to the conditioning information. Unsurprisngly, the GPR method is not able to accurately extrapolate the conditioning information, as it has no additional information regarding the underlying functional distribution.

Moreover, our conditioning method allows us to do *soft conditioning*, where the diffusion process is not conditioned on the observed values for some number of the final diffusions steps.

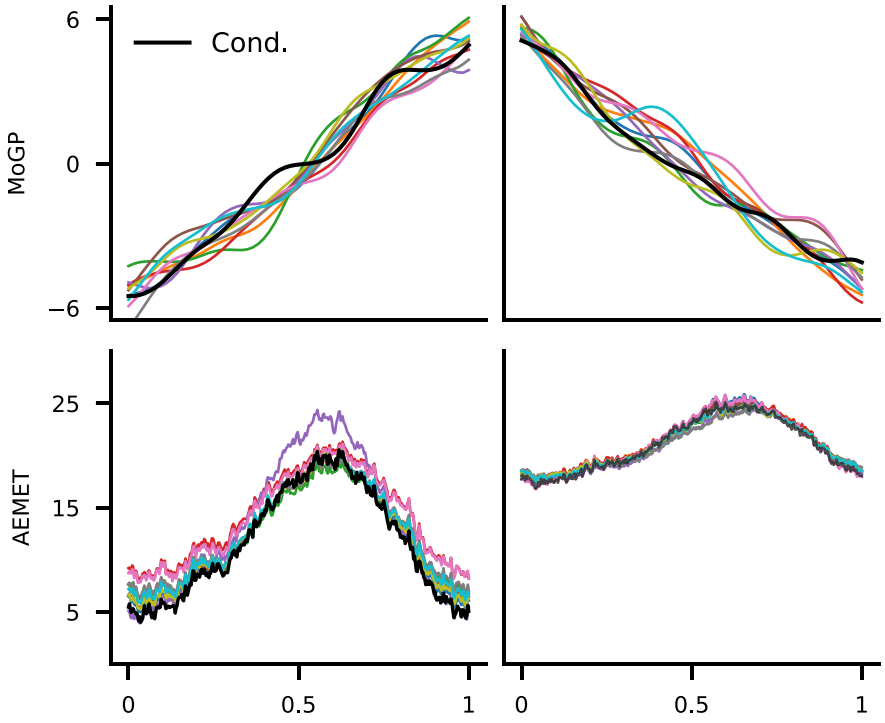


Figure 3.3: An illustration of our soft conditioning method. We condition the generative process on the black curves for all but the final 150 diffusion steps. This allows us to generate functions that are qualitatively similar to the given conditioning information (in black), such that the generated function values do not necessarily exactly match those of the conditioning information.

This allows us to generate curves that are similar to a given observation, but not exactly matching. For example, this can be used to select a particular mode to sample from in a multimodal dataset. We demonstrate this in Figure 3.3. As a potential future application, soft conditioning could be applied as a data augmentation method for functional data.

3.5.3 Function Spaces

Lastly, we experiment with the choice of function space. In particular, we compare the use of the $L^2(X, \omega)$ inner product against the use of the $H^1(X, \omega)$ inner product. Intuitively, the derivative term in Sobolev inner product will penalize generated functions that are not

Table 3.1: Mean smoothness of generated functions as measured by the standard deviation of the function derivatives, averaged across 500 samples. Using the Sobolev norm over the L^2 norm can significantly increase the smoothness of generated functions, while not harming performance if the generated functions are already sufficiently smooth.

Dataset	$L^2(X, \omega)$	$H^1(X, \omega)$
Linear	0.753	0.203
MoGP	24.73	24.74

smooth. In Table 2, we measure the smoothness of generated curves by computing the mean standard deviation of the derivatives of said curves. We find empirically that using the Sobolev loss can result in significantly smoother generations when the underlying functional dataset is highly regular. As smoothness is not a desirable property for the AEMET dataset, we include here a dataset consisting of linear functions (*Linear*) instead. See Appendix A.4 for more on this dataset. We use the Matérn kernel with $\nu = 3/2$ when working with the the Sobolev norm as this kernel has differentiable sample paths.

3.5.4 Spectral Loss

In our previous experiments, we approximate the functional KL divergence by discretizing the underlying operators. In this section, we experiment with an alternative approach based on the spectrum of the covariance operator. We focus here on the setting $H = L^2(X, \mu)$ with $X = [0, 1]$ equipped with the Lebesgue measure $\omega = \mathrm{d}x$. Consider a Gaussian measure on H with covariance operator C. Since C is self-adjoint and compact, the spectral theorem tells us that the eigenfunctions of C form an orthonormal basis of H. We denote the eigenvalues and eigenfunctions of C by $\{(\lambda_j, e_j)\}_{j=1}^{\infty}$. We then have that [Da Prato and Zabczyk, 2014,

Remark 2.24]

$$\mathsf{KL}\left[\mathcal{N}(m_1, C) \,\middle\|\, \mathcal{N}(m_2, C)\right] = \frac{1}{2}\langle m_1 - m_2, C^{-1}(m_1 - m_2)\rangle_{L^2(X,\omega)} \tag{3.64}$$

$$= \frac{1}{2}\sum_{j=1}^{\infty} \lambda_j^{-1}\langle m_1 - m_2, e_j\rangle^2_{L^2(X,\omega)} \tag{3.65}$$

$$\approx \frac{1}{2}\sum_{j=1}^{J} \lambda_j^{-1}\langle m_1 - m_2, e_j\rangle^2_{L^2(X,\omega)}. \tag{3.66}$$

Thus, an alternative method for approximating the KL divergence between Gaussian measures with equal covariance operators is to truncate the above sum at some specified number of terms J. For some choices of C, the eigenvalues and eigenfunctions are analytically known – for example, see Williams and Rasmussen [2006, Chapter 4] for the squared-exponential kernel, and see Le Maître and Knio [2010, Chapter 2] or Burt [2018, Section 2.5] for the exponential kernel.

In Figure 3.4, we compare this spectral approach to the discrete approach proposed in Section (3.4). In particular, we specify C via a Gaussian process with a Matérn kernel with $\nu = 1/2$, unit variance, and lengthscale $\ell = 0.1$. This is done to match the settings in our other experiments. Moreover, the eigenvalues and eigenfunctions are analytically available in this case [Le Maître and Knio, 2010, Burt, 2018]. In each row of Figure 3.4, we specify particular functions for m_1 and m_2. We vary the discretization size (i.e. the number of function observations) on the horizontal axis for discretization sizes of $10, 50, 100, 300$, and plot the estimated KL divergence between $\mathcal{N}(m_1, C)$ and $\mathcal{N}(m_2, C)$ on the vertical axis.

We observe that the discrete approximation to the KL divergence (in blue) is monotonically increasing, as was proved in Proposition (18). However, we see that the spectral approximation is sensitive to both the number of terms in the series expansion and the discretization size. In particular, when using $J = 10$ terms, the spectral approximation underestimates the true KL divergence. In contrast, when $J \geq 50$, we see that the spectral approximation

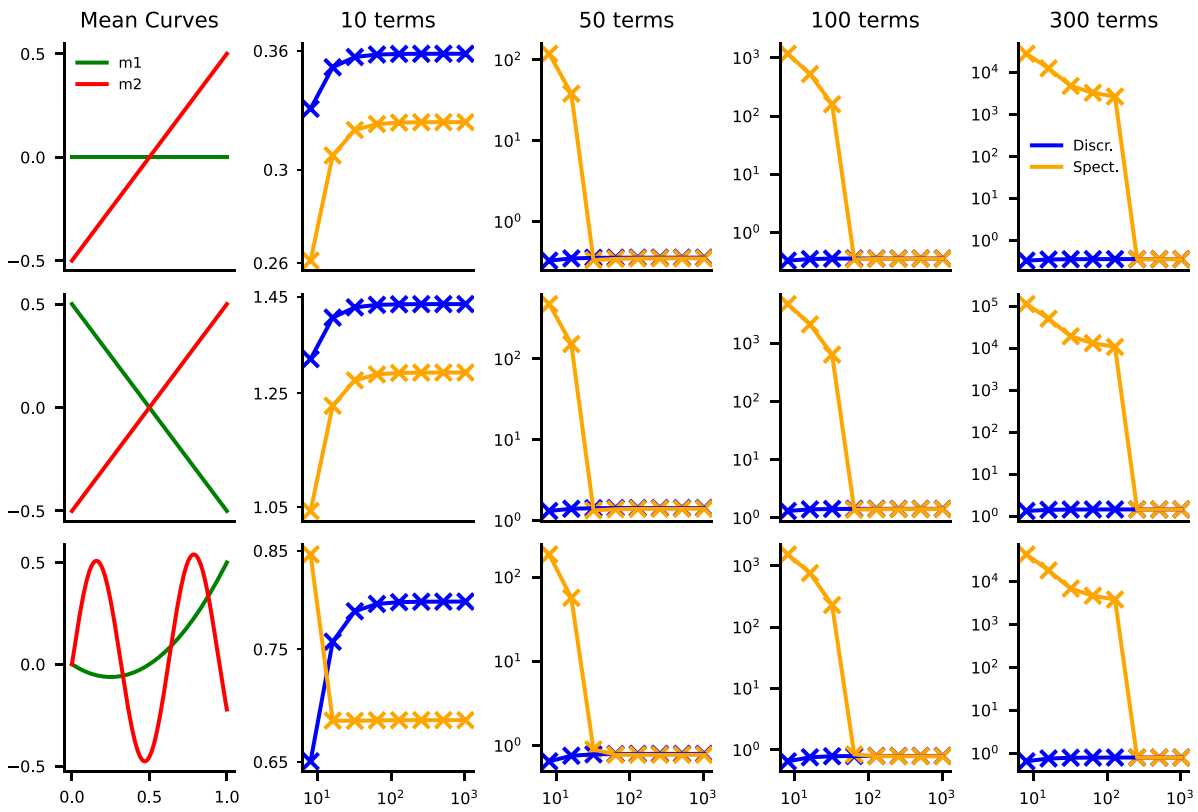


Figure 3.4: Various synthetic functions (first column) and estimates of the KL divergence between Gaussian measures with these means, having covariance operator given by an exponential kernel. For columns 2-5, the horizontal axis corresponds to discretization size (i.e. number of function observations), and the vertical axis corresponds to the corresponding estimated KL divergence. The discrete method (in blue) has KL estimates that are monotonically increasing (see also Proposition (18)), but the spectral method (in orange) is sensitive to the choice of terms in the series expansion as well as the discretization size.

overestimates the true KL divergence by several orders of magnitude if the discretization of X is not sufficiently fine. This effect worsens as we increase the number of terms J. We conjecture that this is because the eigenfunctions e_j are sinusoidal in this case, and thus without a sufficiently fine discretization of X, the inner product in the spectral approximation is a poor numerical estimate of the true inner product.

3.6 Conclusion

We propose a framework for diffusion generative modeling in infinite-dimensional spaces and develop practical techniques for realizing this framework on real-world data. Enabled by our framework, future functional diffusion models may be able move beyond the typical L^2-space assumption in order to incorporate informative prior information.

Chapter 4

Functional Flow Matching

In Chapter 3, we developed the foundations for discrete-time function space diffusion models. However, the discrete-time nature of this model can be limiting in practice, as one is unable to leverage flexible differential equation solvers at sampling time [Song et al., 2021, Jolicoeur-Martineau et al., 2021]. While work following Kerrigan et al. [2023] studies function-space SDEs [Lim et al., 2023c, Franzese et al., 2024, Pidstrigach et al., 2023], there has been growing interest in developing deterministic ODE-based alternatives to diffusion models [Lipman et al., 2023].

In this chapter,[1] we continue to add to the growing literature on function space generative models. In particular, we propose Functional Flow Matching (FFM), a continuous-time normalizing flow model for functional data. Given that Euclidean normalizing flow methods [Papamakarios et al., 2021, Kobyzev et al., 2020] are posed in terms of densities, which generally do not exist in infinite-dimensional spaces, a key challenge of performing this generalization is to pose a purely measure-theoretic model. In particular, our model is a generalization of the recently proposed Flow Matching model of Lipman et al. [2023].

[1]The content of this chapter was previously published as *Functional Flow Matching* (AISTATS 2024) [Kerrigan et al., 2024a], with minor changes.

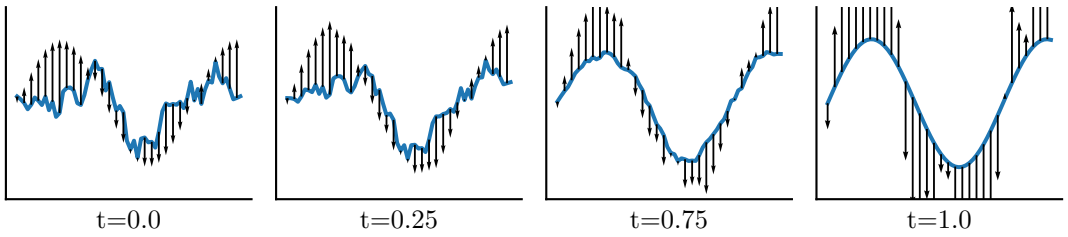


Figure 4.1: An illustration of our FFM method. The vector field $v_t(f) \in H$ (in black) transforms a noise sample $g \sim \mu_0 = \mathcal{N}(0, C_0)$ drawn from a Gaussian process with a Matérn kernel (at $t = 0$) to the function $f(x) = \sin(x)$ (at $t = 1$) via solving a function space ODE. By sampling many such $g \sim \mu_0$, we define a conditional path of measures μ_t^f approximately interpolating between $\mathcal{N}(0, C_0)$ and the function f, which we marginalize over samples $f \sim \nu$ from the data distribution in order to obtain a path of measures approximately interpolating between μ_0 and ν.

Our proposed FFM model first constructs a path of conditional Gaussian measures, approximately interpolating between a fixed reference Gaussian measure and a given function. A path of measures interpolating between said reference measure and the data distribution is then obtained by marginalizing these conditional paths over the data distribution. We learn a vector field on our space of functions which approximately generates this path of measures, allowing us to generate samples from our data distribution by solving a differential equation. Figure 4.1 illustrates our approach.

Our approach allows for simulation-free training, in the sense that no samples are drawn from the model. Moreover, our training objective is regression based, allowing us to avoid pathologies with regards to maximum likelihood training in a functional setting. We empirically verify our framework on several time series datasets and a fluid dynamics dataset, demonstrating that the FFM model outperforms several competitive function space models across a variety of domains.

4.1 Related Work

Here, we review work which is closely related to our proposed method.

Flow Matching and Normalizing Flows We generalize the Flow Matching model of Lipman et al. [2023], which is a novel approach to simulation-free continuous-time normalizing flows [Chen et al., 2018, Papamakarios et al., 2021, Kobyzev et al., 2020]. This approach has demonstrated impressive capabilities on several image generation tasks. However, Flow Matching and other recently proposed simulation-free continuous normalizing flows have only been explored for data distributions supported on finite-dimensional spaces, such as Euclidean spaces [Lipman et al., 2023, Albergo and Vanden-Eijnden, 2023, Liu et al., 2023, Neklyudov et al., 2023] and Riemannian manifolds [Chen and Lipman, 2024, Ben-Hamu et al., 2022]. In contrast, we propose Functional Flow Matching, a continuous-time normalizing flow for infinite-dimensional data. To the best of our knowledge, this is the first normalizing flow model posed in infinite-dimensional spaces.

Function Space Generative Models Recently, a number of authors have proposed function space generalizations of various deep generative models. Close in spirit to our work are those generalizing diffusion models [Ho et al., 2020, Song et al., 2021, Song and Ermon, 2019] to the infinite-dimensional setting. In particular, Kerrigan et al. [2023] and Lim et al. [2023a] propose function space generalizations of discrete-time diffusion models, whereas Pidstrigach et al. [2023], Franzese et al. [2024] and Hagemann et al. [2023] propose function space generalizations of continuous-time diffusion models. Beyond diffusion models, function space GANs [Rahman et al., 2022] and energy-based models [Lim et al., 2023b] have also been proposed. Our work adds to this growing literature on function space generative models by proposing the first function space normalizing flow model.

Discrete Functional Generative Models While there has been growing interest in developing generative models directly in infinite-dimensional spaces, there has also been work proposing generative models for functional data that operate directly on a discretization of the underlying space, for example, diffusion models for time series [Tashiro et al., 2021, Rasul et al., 2021a, Yan et al., 2021] (see Lin et al. [2023] for a recent survey on these methods). Other models, such as normalizing flows [Rasul et al., 2021b], latent variable models [Zhou et al., 2022, Rubanova et al., 2019, Yildiz et al., 2019], and GANs [Yoon et al., 2019, Kidger et al., 2021] have also been explored. However, these methods all operate directly on the discrete observations of a given time series. This has several drawbacks: for instance, it is difficult to transfer a model trained on one discretization to another, and often these models are ill-posed in the functional limit (i.e. as the discretization size goes to zero). In contrast, our work begins from a function space point of view, where we only discretize in order to perform computations.

4.2 Notation and Background

We begin by introducing some notation and background which we will later use to construct our model. Section 4.2.1 introduces notions related to flows on function spaces, and Section 4.2.2 introduces the weak continuity PDE [Stepanov and Trevisan, 2017] which plays a key role in our constructions.

4.2.1 Preliminaries

This chapter follows the notation outlined in Chapter 2, which we briefly recall here. Let $X \subset \mathbb{R}^d$ and consider a real separable Hilbert space H of functions $f : X \to \mathbb{R}$ equipped with the Borel σ-algebra $\mathcal{B}(H)$. We consider the setting where there is a probability measure ν on

H from which we have samples, i.e. random functions drawn from the data distribution ν. Our goal is to build a generative model which allows us to sample from ν. Importantly, any such generative model should be discretization-invariant, in the sense that the model should be able to generate functions which may be observed on any finite but arbitrary discretization of X.

In this work, we consider paths of probability measures $(\mu_t)_{t\in[0,1]}$, where $\mu_t \in \mathbb{P}(H)$ is a probability measure on H at every time $t \in [0, 1]$. In particular, we will construct a path of measures which approximately interpolates between a fixed reference measure μ_0 at time $t = 0$ and the data distribution at time $t = 1$, so that $\mu_1 \approx \nu$.[2] This interpolation is approximate in the sense that μ_1 will be a smoothed version of the data distribution, obtained from ν via convolution with a Gaussian measure having small variance [Bogachev, 1998, Appendix A].

We consider paths of probability measures which are generated locally, in the sense that there is some underlying time-dependent vector field on H such that the path of measures $(\mu_t)_{t\in[0,1]}$ is obtained by flowing samples $g \sim \mu_0$ along said vector field. More formally, a *(time-dependent) vector field* on H is a mapping $v : [0, 1] \times H \to H$.

The *flow* associated to a vector field $(v_t)_{t\in[0,1]}$ is the mapping $\phi : [0, 1] \times H \to H$ specified by the initial value problem

$$\partial_t \phi_t(g) = v_t(\phi_t(g)) \qquad \phi_0(g) = g. \tag{4.1}$$

As written, Equation (4.1) represents an ordinary differential equation (ODE) on the abstract and potentially infinite-dimensional space H. Such ODEs are often dubbed *abstract differential equations* [O'Regan, 1997, Zaidman, 1999]. We assume that all vector fields in this work are sufficiently regular such that a solution to Equation (4.1) is guaranteed to exist for all

[2] In Chapter 3, we used μ_0 to represent the data distribution. In this chapter, we will use the convention that μ_1 should represent (an approximation of) the data distribution. This notation is chosen to align with the existing conventions in the flow and diffusion literature.

$t \in [0,1]$ and ν-a.e. initial condition g.

Given any initial probability measure $\mu_0 \in \mathbb{P}(H)$, we may consider the path of probability measures generated by the flow ϕ. That is, for any $t \in [0,1]$ we define the measure μ_t via the pushforward of μ_0 along ϕ_t, i.e. $\mu_t = [\phi_t]_{\#}\mu_0$, so that $\mu_t(A) = \mu_0\left(\phi_t^{-1}(A)\right)$ for any measurable $A \subset H$. Here, ϕ_t is assumed to be measurable for all $t \in [0,1]$.

4.2.2 Weak Continuity PDE

Previously, we noted how one may obtain a path of probability measures from an initial probability measure $\mu_0 \in \mathbb{P}(H)$ by considering the pushforward of μ_0 along the flow of a given vector field $(v_t)_{t\in[0,1]}$.

Conversely, we say that the vector field $(v_t)_{t\in[0,1]}$ *generates* the path of measures $(\mu_t)_{t\in[0,1]}$ if the path $(\mu_t)_{t\in[0,1]}$ is obtained via the pushforward of μ_0 along the flow associated with $(v_t)_{t\in[0,1]}$. Directly verifying whether a vector field generates a given path of measures (by verifying the pushforward relationship) is typically infeasible. Instead, we can check if the two satisfy the *continuity equation*

$$\partial_t \mu_t + \operatorname{div}(v_t \mu_t) = 0 \qquad \text{on } \; H \times [0,1]. \tag{4.2}$$

We interpret this partial differential equation (PDE) in the weak sense [Ambrosio et al., 2005, Ch. 8], by which we mean that the pair $(v_t)_{t\in[0,1]}$ and $(\mu_t)_{t\in[0,1]}$ satisfy Equation (4.2) if

$$\int_0^1 \int_H \left(\partial_t \varphi(g,t) + \langle v_t(g), \nabla_g \varphi(g,t)\rangle\right) \mathrm{d}\mu_t(g)\,\mathrm{d}t = 0 \tag{4.3}$$

for all $\varphi : H \times [0,1] \to \mathbb{R}$ in an appropriate space of test functions. Typically, φ is assumed to be *cylindrical*, i.e. of the form $\varphi(f,t) = \psi(\pi(f),t)$ where $\pi : g \mapsto (\langle g, e_1\rangle, \ldots, \langle g, e_d\rangle)$ is a

d-dimensional projection of g via an orthonormal system $(e_i)_{i=1}^d$, and $\psi \in C_c^\infty(\mathbb{R}^d \times [0,1])$ is smooth and compactly supported.

We refer to Stepanov and Trevisan [2017, Theorem 3.4] for a rigorous discussion of this result in general metric spaces. Such results are often referred to as *superposition principles*. In the Euclidean setting, if one assumes that all measures admit a density with respect to some common dominating measure, it suffices to check the continuity equation directly, in which μ_t is replaced by a density p_t [Ambrosio et al., 2005, Villani, 2009].

Throughout this work, we assume all paths of measures and vector fields are sufficiently regular such that the superposition principle applies, i.e. it suffices to check the continuity equation to conclude whether a given path of measures is generated by a given vector field. In Theorem 19, we use the weak form of the continuity equation in order to construct a marginal vector field from conditional vector fields, such that this marginal vector field is guaranteed to generate our desired interpolating path of measures.

4.3 Function Space Flow Matching

Building on the notions in Section 4.2, we now introduce our Functional Flow Matching model (FFM). The Flow Matching model is a recently proposed continuous-time normalizing flow method developed for finite-dimensional spaces [Lipman et al., 2023, Chen and Lipman, 2024]. Our FFM approach builds on this earlier line of work to develop an extension of these methods to infinite-dimensional spaces.

The main technical challenge of generalizing the existing techniques to infinite-dimensional spaces is that existing methods rely heavily on the notion of a probability density function, either with respect to the Lebesgue measure in the case of a Euclidean space or with respect to the canonical volume measure on a Riemannian manifold. In infinite-dimensional (Banach)

spaces, there does not exist an analogue of the Lebesgue measure – that is, any nonzero translation invariant Borel measure must assign infinite measure to any open set [Eldredge, 2016].

As such, our FFM model is necessarily posed in measure-theoretic terms. Our derivations shed light on strict requirements needed to obtain a well-posed model. For instance, we require an absolute continuity assumption between the conditional and marginal measures defined in Section 4.3.1. In Euclidean spaces, such assumptions are easy to satisfy, but are non-trivial in infinite-dimensional spaces, even for the simple setting of Gaussian measures (see Section 4.3.3). Moreover, our derivations demonstrate that naively applying a white-noise Gaussian measure (as is done in the Euclidean setting) leads to an ill-posed model in function space.

4.3.1 Constructing a Path of Measures

Suppose we associate to every $f \in H$ a path of measures $(\mu_t^f)_{t\in[0,1]}$ such that $\mu_0^f = \mu_0$ is some fixed reference measure and μ_1^f is concentrated around f. For instance, μ_1^f could be a Gaussian measure with mean f and a covariance having small operator norm. We then marginalize over all such measures, where we mix over the data distribution ν. That is, we define a new probability measure $\mu_t \in \mathbb{P}(H)$ for $t \in [0,1]$ via

$$\mu_t(A) = \int \mu_t^f(A)\,\mathrm{d}\nu(f) \qquad \forall A \in \mathcal{B}(H). \tag{4.4}$$

Due to our conditions on μ_t^f, we then have that $\mu_0 = \mu_0$ and $\mu_1 \approx \nu$ is approximately the data distribution. Suppose further that each conditional path of measures μ_t^f is generated by some known vector field v_t^f. In the following theorem, we claim that we may construct a vector field v_t which generates the *marginal* path of measures μ_t from the conditional vector

fields v_t^f.

Theorem 19.

Assume that $\int_0^1 \int_{H\times H} ||v_t^f(g)||\,\mathrm{d}\mu_t^f(g)\,\mathrm{d}\nu(f)\,\mathrm{d}t < \infty$. *If* $\mu_t^f \ll \mu_t$ *for* ν*-almost every* f *and almost every* $t \in [0,1]$, *then the vector field*

$$v_t(g) = \int_H v_t^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\,\mathrm{d}\nu(f) \tag{4.5}$$

generates the marginal path of measures $(\mu_t)_{t\in[0,1]}$ *specified by Equation* (4.4). *That is,* $(v_t)_{t\in[0,1]}$ *and* $(\mu_t)_{t\in[0,1]}$ *solve the continuity equation* (4.2). *Here,* $\mathrm{d}\mu_t^f/\,\mathrm{d}\mu_t$ *is the Radon-Nikodym derivative of the conditional measure with respect to the marginal.*

Proof. In this proof, we denote the variable of integration for integrals over H via a subscript on the integral. We show that for an arbitrary but fixed test function φ,

$$\int_0^1 \int_g \partial_t\varphi(g,t)\,\mathrm{d}\mu_t(g)\,\mathrm{d}t = -\int_0^1 \int_g \langle v_t(g), \nabla_g\varphi(g,t)\,\mathrm{d}\mu_t(g)\,\mathrm{d}t. \tag{4.6}$$

To that end, we begin by analyzing the left-hand side, replacing the integration of the marginal measure μ_t with a double integral over its components:

$$\int_0^1 \int_g \partial_t\varphi(g,t)\,\mathrm{d}\mu_t(g)\,\mathrm{d}t = \int_0^1 \int_f \int_g \partial_t\varphi(g,t)\,\mathrm{d}\mu_t^f(g)\,\mathrm{d}\nu(f)\,\mathrm{d}t \tag{4.7}$$

By Fubini-Tonelli and using the assumption that v_t^f generates μ_t^f, we obtain via the continuity equation for (v_t^f, μ_t^f):

$$= -\int_f \int_0^1 \int_g \langle v_t^f(g), \nabla_g\varphi(g,t)\rangle\,\mathrm{d}\mu_t^f(g)\,\mathrm{d}t\,\mathrm{d}\nu(f) \tag{4.8}$$

Using our absolute continuity assumption and Fubini-Tonelli once again, we perform a change

of measure to obtain

$$= -\int_0^1 \int_f \int_g \langle v_t^f(g), \nabla_g \varphi(g,t) \rangle \left(\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g) \right) \mathrm{d}\mu_t(g)\, \mathrm{d}\nu(f)\, \mathrm{d}t \tag{4.9}$$

$$= -\int_0^1 \int_f \int_g \langle v_t^f(g) \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g), \nabla_g \varphi(g,t) \rangle\, \mathrm{d}\mu_t(g)\, \mathrm{d}\nu(f)\, \mathrm{d}t \tag{4.10}$$

Using the fact that Bochner integrals commute with inner products, an application of Fubini-Tonelli yields

$$= -\int_0^1 \int_g \left\langle \int_f v_t^f(g) \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\, \mathrm{d}\nu(f), \nabla_g \varphi(g,t) \right\rangle \mathrm{d}\mu_t(g)\, \mathrm{d}t \tag{4.11}$$

$$= -\int_0^1 \int_g \langle v_t(g), \nabla_g \varphi(g,t) \rangle\ \mathrm{d}\mu_t(g)\, \mathrm{d}t \tag{4.12}$$

Hence, we have shown that the vector field generating μ_t is given by

$$v_t(g) = \int_f v_t^f(g) \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\, \mathrm{d}\nu(f). \tag{4.13}$$

□

If this vector field v_t were known, we could generate samples by solving the corresponding flow ODE (Equation (4.1)) with initial condition $f \sim \mu_0$ drawn from our fixed reference measure. However, the vector field specified by Equation (4.5) is intractable. Thus, we will learn a model to approximate this unknown vector field. Note that our model will necessarily be a mapping between infinite dimensional spaces. We discuss how to parametrize such a model in Section 4.4.

The main technical assumption in Theorem 19 is that the conditional distributions μ_t^f are ν-almost surely absolutely continuous with respect to the marginal distribution μ_t. Although this assumption is not generally true even in the Euclidean setting, we prove in Theorem

(20) that this assumption holds under an additional equivalency condition on the conditional measures. In Section 4.3.3, we discuss how this equivalency assumption may be satisfied under a Gaussian parametrization.

Theorem 20.

Consider a probability measure ν on H and a collection of measures μ_t^f parametrized by $f \in H$. Suppose that the collection of parametrized measures are ν-a.e. mutually absolutely continuous. Define the marginal measure μ_t via Equation (4.4)*. Then, $\mu_t^f \ll \mu_t$ for ν-a.e. f.*

Proof. By assumption, there exists $\mathcal{G} \subseteq H$ with $\nu(G) = 1$ and for any $f, g \in \mathcal{G}$, we have $\mu_t^f \ll \mu_t^g$ and $\mu_t^g \ll \mu_t^f$. Fix $A \in \mathcal{B}(H)$ with $\mu_t(A) = 0$. We claim that $\mu^f(A) = 0$ for every $f \in \mathcal{G}$. Note that as $\mathcal{G} \subseteq H$ has full measure, the integral defining μ_T may be taken over $\mathcal{G}$ rather than H. Suppose for the sake of contradiction that $\mu_t^f(A) > 0$ for some $f \in \mathcal{G}$. From the mutual equivalencies of the measures parametrized by $\mathcal{G}$, it follows that $\mu_t^g(A) > 0$ for every $g \in \mathcal{G}$. Given the form of the mixture measure μ_t, it would then follow that $\mu_t(A) > 0$, which is a contradiction. Thus, $\mu_t^f \ll \mu_t$ for ν-a.e. f as claimed. □

4.3.2 Special Case: Gaussian Measures

In this section, we specialize to the setting where the reference measure μ_0 and conditional measures μ_t^f are chosen to be Gaussian measures [Bogachev, 1998]. We make this ansatz for several reasons. Foremost, our marginal vector field (Equation (4.5)) requires an absolute continuity assumption. In infinite-dimensional (separable) Banach spaces, the absolute continuity of Gaussian measures is well-understood, e.g. via the Cameron-Martin theorem and the Feldman-Hájek theorem [Da Prato and Zabczyk, 2014, Bogachev, 1998]. Moreover, we are able to parametrize our Gaussian measures via Gaussian processes [Rasmussen and Williams, 2006, Wild et al., 2022] for which a number of flexible choices of kernels have been explored in the machine learning literature.

More formally, for any $f \in H$ we define a conditional path of probability measures $(\mu_t)_{t\in[0,1]}$ to be a Gaussian measure $\mu_t^f = \mathcal{N}(m_t^f, C_t^f)$ with mean $m_t^f \in H$ and covariance operator $C_t^f : H \to H$. Note that the C_t^f are necessarily symmetric, non-negative and trace-class [Da Prato and Zabczyk, 2014, Ch. 2]. In particular, this rules out multiples of the identity operator (corresponding to white noise) as a valid choice for C_t^f, as these operators are not compact and hence not trace-class.

In practice, we parametrize $t \mapsto m_t^f$ by a Fréchet differentiable mapping and specify C_t^f by a covariance operator C_0 and variance schedule $t \mapsto \sigma_t^f \in \mathbb{R}_{>0}$ such that $C_t^f = (\sigma_t^f)^2 C_0$. At time $t = 0$, we choose to parametrize $\mu_0^f = \mu_0 = \mathcal{N}(0, C_0)$ as a centered Gaussian measure independent of the function $f \in H$. The measure μ_0 will serve as the reference measure in our generative model. In order to satisfy the desiderata of Section 4.3.1, at time $t = 1$ we will choose $m_1^f = f$ and C_t^f to have small operator norm so that μ_1^f is a Gaussian measure concentrated around f.

In this case, we note that the conditional flow $\phi^f : [0, 1] \times H \to H$ defined via $\phi_t^f(g) = \sigma_t^f g + m_t^f$ will push $g \sim \mathcal{N}(0, C_0)$ to the desired conditional measure μ_t^f, i.e. $\mu_t^f = [\phi_t^f]_{\#}\mathcal{N}(0, C_0)$. Using the flow ODE (Equation (4.1)), we see that a vector field generating this conditional path of measures is

$$v_t^f(g) = \frac{(\sigma_t^f)'}{\sigma_t^f}(g - m_t^f) + \frac{\mathrm{d}}{\mathrm{d}t} m_t^f \tag{4.14}$$

where $(\sigma_t^f)'$ is the ordinary time derivative of the variance schedule and $\mathrm{d}/\,\mathrm{d}t(m_t^f)$ is the Fréchet derivative of the mapping $t \mapsto m_t^f$. The proof of this fact is a straightforward generalization of Lipman et al. [2023, Theorem 3], which demonstrates the analogous relationship in finite-dimensional Euclidean spaces.

In this work, we consider two concrete parameterizations. In the first parametrization ("OT"),

the mean and variance are given as affine functions of t and f:

$$m_t^f = tf \qquad \sigma_t^f = 1 - (1 - \sigma_{\min})t. \tag{4.15}$$

The "OT" path is named as such as it corresponds to an optimal transport map between Gaussians in the Euclidean setting [Lipman et al., 2023, McCann, 1997].

In the second parametrization ("VP"), we set

$$m_t^f = \alpha_{1-t} f \qquad \sigma_t^f = \sqrt{1 - \alpha_{1-t}^2}. \tag{4.16}$$

This path is inspired inspired by probability paths defined via variance preserving diffusion models [Lipman et al., 2023, Song et al., 2021]. We additionally experimented with the "variance exploding" parametrization [Lipman et al., 2023, Song et al., 2021], but found empirically that this was not suitable for our setting. See Appendix B.1 for details. Here, $\sigma_{\min} \in \mathbb{R}_{>0}$ and $\alpha_t \in \mathbb{R}_{>0}$ are hyperparameters of the model controlling the variance of the conditional measures.

4.3.3 Absolute Continuity for Gaussians

In general, the absolute continuity assumption of Theorem 19 is difficult to satisfy in function spaces. In the Gaussian setting, we may reduce this assumption to assumptions regarding the parametrization of our Gaussian measures. By the Feldman-Hájek theorem [Da Prato and Zabczyk, 2014, Theorem 2.25], our conditional Gaussian measures μ_t^f will be mutually absolutely continuous if the difference in means lies in the Cameron-Martin space of C_t, i.e. $m_t^f - m_t^g \in C_t^{1/2}(H)$.

Thus, under suitable assumptions on the data distribution ν and an appropriate parametriza-

tion of the conditional means, our marginal vector fields (Equation (4.5)) will be well-defined as a consequence of Theorem 20. Suppose $C_t = \sigma_t^2 C_0$ is a scaled version of some fixed covariance operator C_0 with the assumption that $0 < \sigma_t^2 \leq M$ is positive and bounded above. By Lemma 6.15 of Stuart [2010], this choice guarantees us that the Cameron-Martin space is constant in time, i.e. $C_0^{1/2}(H) = C_t^{1/2}(H)$ for all $t \in [0, 1]$.

Assume further that the data distribution is supported on the Cameron-Martin space of C_0, i.e. $\nu(C_0^{1/2}(H)) = 1$. In this case, given our covariance parametrization, our Gaussian measures will be mutually absolutely continuous if e.g. m_t^f is an affine function of f. We note that the parametrizations suggested in Section 4.3.2 are all affine, and so under the assumption that the data is supported on the Cameron-Martin space $C_0^{1/2}(H)$ our setup is well-defined.

In practice, verifying whether the data distribution is supported on $C_0^{1/2}(H)$ is difficult. One option to guarantee this assumption is satisfied is to pre-process the data via some mapping $T : H \to C_0^{1/2}(H) \subseteq H$ whose image is contained in $C_0^{1/2}(H)$. We refer to Appendix C of Lim et al. [2023a] for a further discussion of such mappings and related results. We note that in practice, we do not find it necessary to perform this pre-processing.

4.3.4 Training the FFM Model

Ideally, we would like to perform functional regression on the marginal vector field defined via Equation (4.5), where we approximate $v_t(g)$ by a model $u_t(g \mid \theta)$ with parameters $\theta \in \mathbb{R}^p$. This could be achieved, for instance, by minimizing the loss

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], g \sim \mu_t} \left[\| v_t(g) - u_t(g \mid \theta) \|^2 \right] \tag{4.17}$$

where $\mathcal{U}[0, 1]$ denotes a uniform distribution over the interval $[0, 1]$. Note here that our model

is a mapping $u : \mathbb{R}^p \times [0,1] \times H \to H$, i.e. our model is a parametrized, time-dependent operator on the function space H. However, such a loss is intractable to compute – in fact, if we had access to $(v_t)_{t\in[0,1]}$, there would be no need to learn a model. Consider instead the conditional loss, defined via

$$\mathcal{J}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1], f\sim\nu, g\sim\mu_t^f}\left[\left\|v_t^f(g) - u_t(f \mid \theta)\right\|^2\right] \tag{4.18}$$

where, rather than regressing on the intractable v_t, we regress on the *known* conditional vector fields v_t^f. In the following theorem, we claim that minimizing $\mathcal{J}(\theta)$ is equivalent to minimizing $\mathcal{L}(\theta)$.

Theorem 21.

Assume that the true and model vector fields are square-integrable, i.e. $\int_0^1 \int_H \|v_t(g)\|^2 \,\mathrm{d}\mu_t(g)\,\mathrm{d}t < \infty$ *and* $\int_0^1 \int_H \|u_t(g \mid \theta)\|^2 \,\mathrm{d}\mu_t(g)\,\mathrm{d}t < \infty$. *Then,* $\mathcal{L}(\theta) = \mathcal{J}(\theta) + C$ *where* $C \in \mathbb{R}$ *is a constant independent of* θ.

Proof. First, note that since we are working in a real Hilbert space, for fixed $f, g \in H$ we have

$$\begin{aligned}\|v_t(g) - u_t(g \mid \theta)\|^2 &= \langle v_t(g) - u_t(g \mid \theta), v_t(g) - u_t(g \mid \theta)\rangle &(4.19)\\ &= \|v_t(g)\|^2 + \|u_t(g \mid \theta)\|^2 - 2\langle v_t(g), u_t(g \mid \theta)\rangle &(4.20)\end{aligned}$$

and similarly,

$$\left\|v_t^f(g) - u_t(g \mid \theta)\right\|^2 = \left\|v_t^f(g)\right\|^2 + \|u_t(g \mid \theta)\|^2 - 2\langle v_t(g), u_t(g \mid \theta)\rangle. \tag{4.21}$$

The first term in both is independent of the model parameters θ. We analyze the remaining two terms. Below, we use subscripts on integrals over $\mathcal{F}$ to denote the variable of integration.

First, using the fact that μ_t is a mixture measure,

$$\mathbb{E}_{t,\mu_t}\left[\|u_t(g \mid \theta)\|^2\right] = \tag{4.22}$$

$$= \int_0^1 \int_g \|u_t(g \mid \theta)\|^2 \, \mathrm{d}\mu_t(g) \, \mathrm{d}t \tag{4.23}$$

$$= \int_0^1 \int_f \int_g \|u_t(g \mid \theta)\|^2 \, \mathrm{d}\mu_t^f(g) \, \mathrm{d}\nu(f) \, \mathrm{d}t \tag{4.24}$$

$$= \mathbb{E}_{t,g\sim\mu_t^f,f\sim\nu}\left[\|u_t(g \mid \theta)\|^2\right]. \tag{4.25}$$

Next, using the exchangeability between Bochner integrals and inner products and Fubini-Tonelli,

$$\mathbb{E}_{t,g\sim\mu_t}\left[\langle v_t(g), u_t(g \mid \theta)\right] = \tag{4.26}$$

$$= \int_0^1 \int_g \langle v_t(g), u_t(g \mid \theta)\rangle \, \mathrm{d}\mu_t(g) \, \mathrm{d}t \tag{4.27}$$

$$= \int_0^1 \int_g \left\langle \int_f v_t^f(g) \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g) \, \mathrm{d}\nu(f), u_t(g \mid \theta) \right\rangle \mathrm{d}\mu_t(g) \, \mathrm{d}t \tag{4.28}$$

$$= \int_0^1 \int_f \int_g \langle v_t^f(g), u_t(g \mid \theta)\rangle \left(\frac{d\mu_t^f}{d\mu_t}(g)\right) \mathrm{d}\mu_t(g) \, \mathrm{d}\nu(f) \, \mathrm{d}t \tag{4.29}$$

$$= \int_0^1 \int_f \int_g \langle v_t^f(g), u_t(g \mid \theta) \, \mathrm{d}\mu_t^f(g) \, \mathrm{d}\nu(f) \, \mathrm{d}t \tag{4.30}$$

$$= \mathbb{E}_{t,f\sim\nu,g\sim\mu_t^f}\left[\langle v_t^f(g), u_t(g \mid \theta)\rangle\right]. \tag{4.31}$$

This shows the equivalency of the two losses. □

4.4 Experiments

We now investigate the empirical performance of FFM on several real-world datasets. In all settings, we assume we are working in the space $H = L^2([0,1])$ and we parametrize $u_t(- \mid \theta)$

via a Fourier Neural Operator (FNO) [Li et al., 2021]. Sampling is achieved by drawing a sample from the reference measure $g \sim \mu_0$ and numerically solving the flow ODE (Equation (4.1)) with initial condition g. In our implementation, we use the DOPRI solver [Dormand and Prince, 1980]. Details can be found in Appendix B.1.[3]

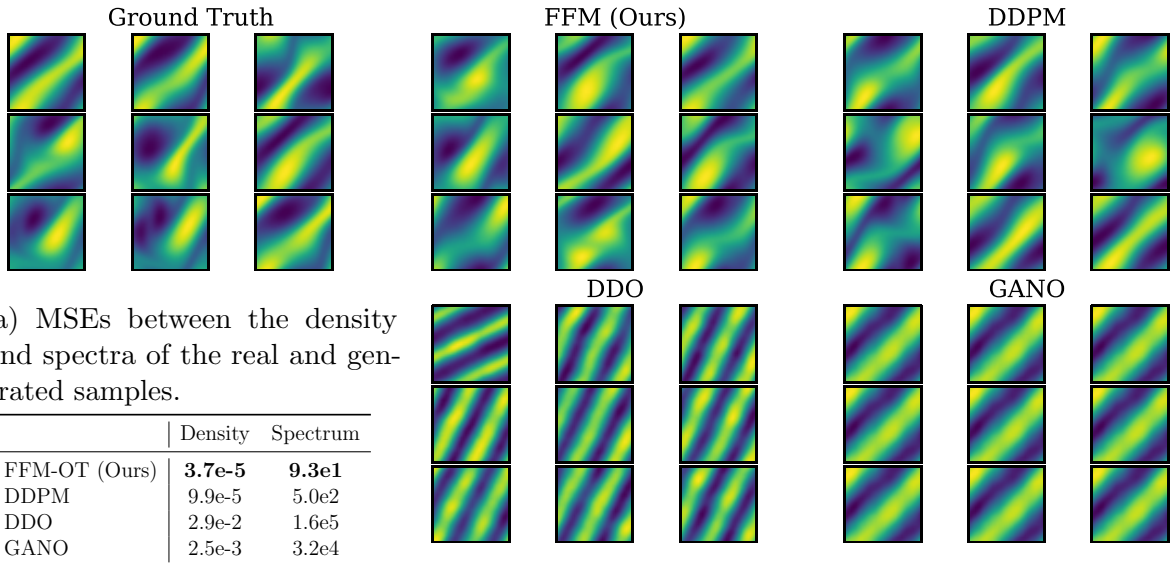
Datasets Our experiments in 1D include five datasets selected for their diverse correlation structures, exhibiting distinctive patterns that enable visual evaluation of generated samples. Plots of original and generated samples, as well as a detailed description of each dataset, can be found in Appendix B.1.2. The first dataset (AEMET) consists of a set of 73 curves describing the mean daily temperature at various locations [Febrero-Bande and de la Fuente, 2012]. The second is a gene expression time series dataset [Orlando et al., 2008], and the remaining three consist of global economic time series on population, GDP per capita, and labor force size [Bolt and Van Zanden, 2020, Inklaar et al., 2018, International Financial Statistics, 2022], We also experiment with a dataset of solutions to the Navier-Stokes equation on a 2D torus [Li et al., 2022].

Baselines We compare against several functional generative models: the Denoising Diffusion Operator (DDO) [Lim et al., 2023a] with NCSN noise scale, GANO [Rahman et al., 2022], and functional DDPM [Kerrigan et al., 2023]. We do not compare to non-functional methods, as we are primarily interested in developing discretization-invariant generative models.

All noise was specified via a Gaussian process with a Matrn kernel where the lenthscale and variance is tuned for each dataset and method. Generally, tuning the parameters of the kernel is key to obtaining high-quality results across all models considered.

For the sake of a fair comparison, we used the same neural architecture for all models, with the exception of GANO which requires a generator and discriminator pair. We used the

[3]Code for all of our experiments is available at `github.com/GavinKerrigan/functional_flow_matching`



(a) MSEs between the density and spectra of the real and generated samples.

	Density	Spectrum
FFM-OT (Ours)	**3.7e-5**	**9.3e1**
DDPM	9.9e-5	5.0e2
DDO	2.9e-2	1.6e5
GANO	2.5e-3	3.2e4

Figure 4.2: Samples from the Navier-Stokes dataset ("ground truth") and samples from the various models considered in this work. Our FFM-OT model and DDPM qualitatively match the ground truth samples, whereas DDO and GANO suffer from mode collapse. Table 4.2a compares the density and spectra between 1000 real and generated samples, showing that our proposed method outperforms the others by a large margin on pointwise metrics. Note that we do not study the FFM-VP parametrization on this dataset due to computational costs.

code provided by the authors of DDPM and GANO but re-implemented the DDO model. For all models, we performed extensive hyperparameter tuning and report the best results. Generally, we find the FFM methods are less sensitive to hyperparameter choices than the baseline methods.

Results Figure 4.2 shows samples from the Navier-Stokes dataset and samples generated from the various models we consider. Qualitatively, our FFM model and the DDPM model match the ground-truth samples, whereas DDO and GANO suffer from mode collapse.

Figure 4.3 shows samples from the AEMET dataset and generated samples from the models we consider. Our FFM model is able to qualitatively match the samples from the ground truth distribution. The DDPM samples are similar in quality, but do not respect the range of values seen in the data. For DDO, we observe smoothness issues, and for GANO, we again see mode collapse issues.

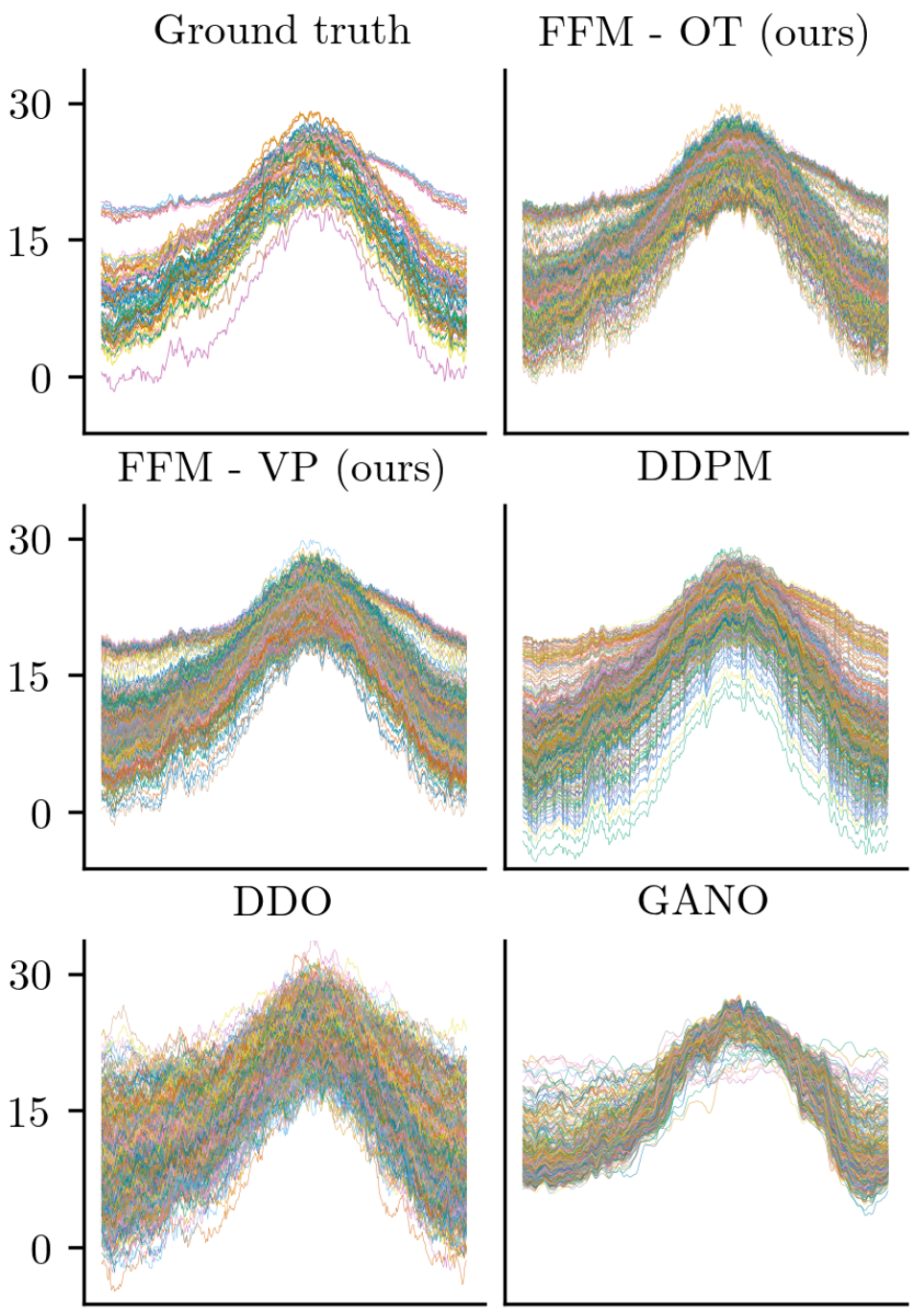


Figure 4.3: Unconditional generation of 500 samples on the AEMET dataset. Samples from our FFM model and DDPM appear visually to better match the characteristics of the real data relative to DDO and GANO.

Table 4.1: Average MSEs between true and generated samples for pointwise statistics on five one-dimensional datasets, along with the standard deviation across ten random seeds. The average number of function evaluations (NFEs) for each sampling procedure in our implementation is also reported. Our FFM models obtain the best average performance across nearly all metrics, while simultaneously requiring fewer NFEs than the diffusion baselines.

		Mean	Variance	Skewness	Kurtosis	Autocorrelation	NFEs
AEMET	FFM-OT (ours)	**8.4e-2** (9.9e-2)	1.7e+0 (1.1e+0)	7.7e-2 (6.6e-2)	3.3e-2 (3.7e-2)	**3.0e-6** (4.0e-6)	668
	FFM-VP (ours)	1.3e-1 (1.4e-1)	**1.5e+0** (1.2e+0)	**5.e-2** (4.3e-2)	**1.7e-2** (1.6e-2)	6.0e-6 (7.0e-6)	488
	DDPM	3.0e-1 (3.0e-1)	3.5e+0 (4.6e+0)	2.2e-1 (2.2e-1)	4.8e-2 (3.7e-2)	1.2e-5 (9.e-6)	1000
	DDO	2.4e-1 (1.4e-1)	6.6e+0 (5.1e+0)	2.1e-1 (4.1e-2)	3.8e-2 (1.3e-2)	6.7e-4 (1.3e-4)	2000
	GANO	6.5e+1 (1.9e+2)	3.7e+1 (4.0e+1)	2.9e+0 (4.8e+0)	3.3e-1 (4.0e-1)	1.2e-3 (3.1e-3)	1
Genes	FFM-OT (ours)	6.7e-4 (4.5e-4)	3.9e-3 (2.6e-4)	2.4e-1 (4.7e-2)	7.7e-2 (9.0e-3)	2.5e-4 (1.7e-4)	386
	FFM-VP (ours)	**4.2e-4** (3.8e-4)	**7.3e-4** (3.5e-4)	**1.9e-1** (6.1e-2)	**4.3e-2** (1.1e-2)	**1.3e-4** (1.0e-4)	290
	DDPM	8.8e-4 (4.5e-4)	1.9e-3 (4.2e-4)	3.6e-1 (1.9e-1)	6.3e-2 (1.1e-2)	4.3e-4 (9.3e-5)	1000
	DDO	4.2e-3 (1.5e-3)	1.2e-3 (3.6e-4)	3.0e-1 (5.7e-2)	1.1e-1 (1.1e-2)	1.0e-3 (1.7e-4)	2000
	GANO	4.6e-3 (2.0e-3)	7.4e-3 (1.5e-3)	1.7e+0 (1.3e+0)	3.3e-1 (8.4e-2)	2.e-3 (1.0e-3)	1
Pop.	FFM-OT (ours)	**3.9e-5** (3.8e-5)	**7.0e-6** (9.e-6)	4.1e+0 (5.3e+0)	9.0e-2 (1.0e-1)	**2.7e-5** (4.6e-5)	662
	FFM-VP (ours)	6.3e-5 (4.5e-5)	**7.0e-6** (7.e-6)	**1.3e+0** (6.1e-1)	7.8e-2 (4.5e-2)	2.5e-3 (5.2e-4)	494
	DDPM	5.7e-5 (5.2e-5)	6.0e-6 (7.0e-6)	1.9e+0 (1.2e+0)	**5.9e-2** (4.4e-2)	5.6e-5 (3.5e-5)	1000
	DDO	1.9e-4 (8.7e-5)	2.7e-4 (1.9e-5)	4.2e+0 (4.1e-1)	2.7e-1 (3.7e-2)	3.2e-2 (1.9e-3)	2000
	GANO	1.1e-3 (9.8e-4)	4.3e-5 (7.1e-5)	8.e+0 (2.4e+0)	8.6e-1 (5.3e-1)	1.6e-3 (3.6e-3)	1
GDP	FFM-OT (ours)	**2.0e-5** (1.2e-5)	9.e-6 (6.e-6)	6.3e-1 (3.5e-1)	**3.9e-2** (1.9e-2)	**2.8e-5** (1.4e-5)	536
	FFM-VP (ours)	4.1e-5 (2.1e-5)	**8.0e-6** (7.0e-6)	**6.2e-1** (4.1e-1)	5.0e-2 (2.5e-2)	1.9e-4 (2.3e-5)	494
	DDPM	1.6e-4 (1.5e-4)	2.5e-5 (2.9e-5)	8.6e-1 (5.9e-1)	5.1e-2 (2.1e-2)	1.4e-4 (1.0e-4)	1000
	DDO	2.1e-4 (1.1e-4)	2.9e-4 (9.4e-5)	1.7e+0 (1.1e-1)	2.7e-1 (2.4e-2)	9.6e-3 (1.5e-3)	2000
	GANO	8.4e-4 (7.8e-4)	5.0e-5 (3.7e-5)	2.6e+0 (1.3e+0)	2.1e-1 (1.4e-1)	1.6e-4 (1.6e-4)	1
Labor	FFM-OT (ours)	**6.9e-5** (6.1e-5)	2.6e-5 (1.1e-5)	5.4e+0 (3.3e+0)	1.5e-1 (1.8e-1)	**1.3e-4** (7.5e-5)	308
	FFM-VP (ours)	7.1e-5 (5.5e-5)	**2.1e-5** (9.0e-6)	**2.0e+0** (1.5e+0)	**8.6e-2** (7.3e-2)	5.8e-4 (1.4e-4)	302
	DDPM	4.2e-4 (3.3e-4)	3.5e-4 (5.6e-4)	1.8e+3 (3.5e+3)	1.0e+1 (1.5e+1)	2.9e-4 (1.6e-4)	1000
	DDO	3.1e-4 (1.9e-4)	4.0e-4 (1.2e-4)	4.8e+0 (5.3e-1)	4.3e-1 (3.9e-2)	7.8e-3 (1.2e-3)	2000
	GANO	3.2e-3 (6.3e-3)	6.5e-4 (4.6e-4)	7.8e+0 (7.6e+0)	1.2e+0 (3.7e-1)	1.8e-3 (9.4e-4)	1

Quantitatively, Table 4.1 evaluates model performance on the one-dimensional datasets by computing pointwise statistics of the generated functions and computing the MSE between these pointwise statistics and those of the real data. Table 4.2a reports the MSE between the density and spectra [Lim et al., 2023b] of the real and generated samples on the Navier-Stokes dataset. See Appendix B.2 for visualizations.

Variants of FFM perform the best, on average, in almost all metrics considered across the wide range of domains on which we performed evaluation. While pointwise statistics have limitations, for functional models there are no clear alternatives for evaluation, and pointwise metrics are broadly used in the literature [Rahman et al., 2022, Lim et al., 2023a]. Together with the qualitative results, these metrics further validate the performance of our method.

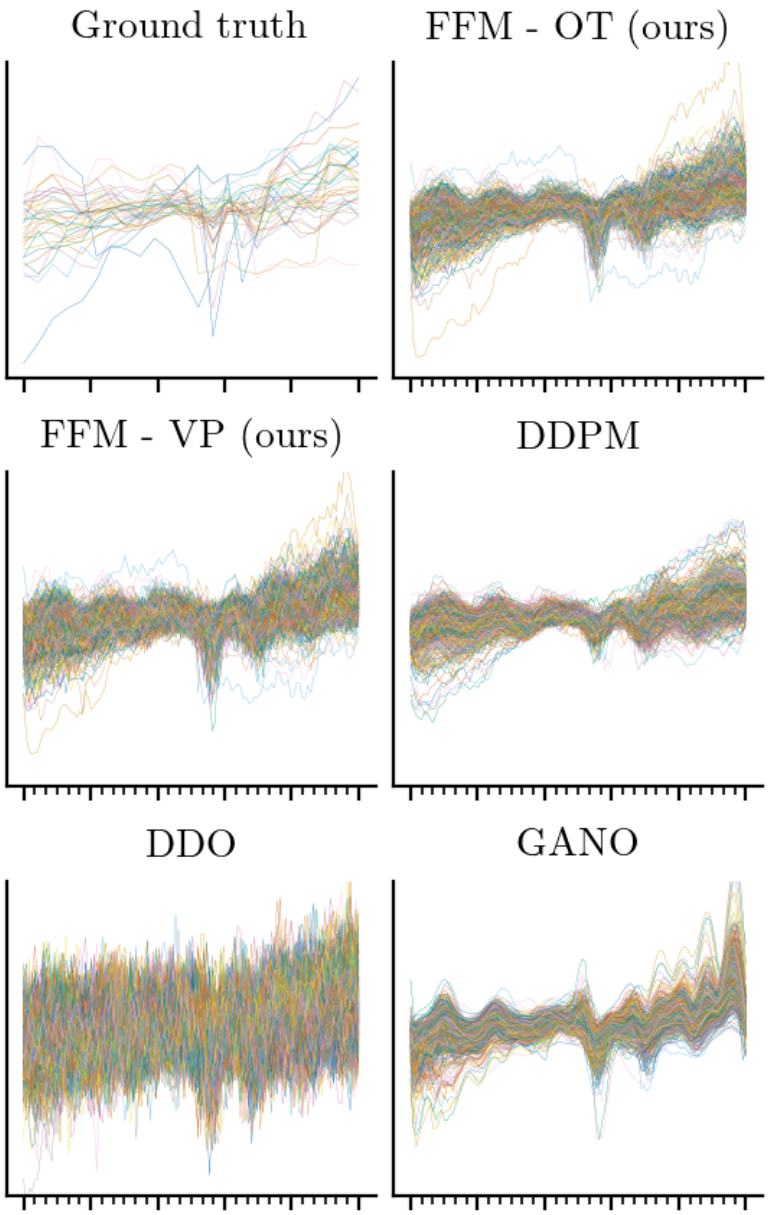


Figure 4.4: Samples from the Labor dataset and samples from the various models at 5x super-resolution.

A key benefit of our model is the ability to perform generation at arbitrary resolutions, a necessary component in any functional task. We demonstrate this in Figure 4.4. All models are trained on the original data resolution, but samples are drawn at a 5x resolution. Samples from FFM and DDPM qualitatively match the characteristics of the ground truth distribution, whereas samples from DDO and GANO do not match the smoothness of the original data.

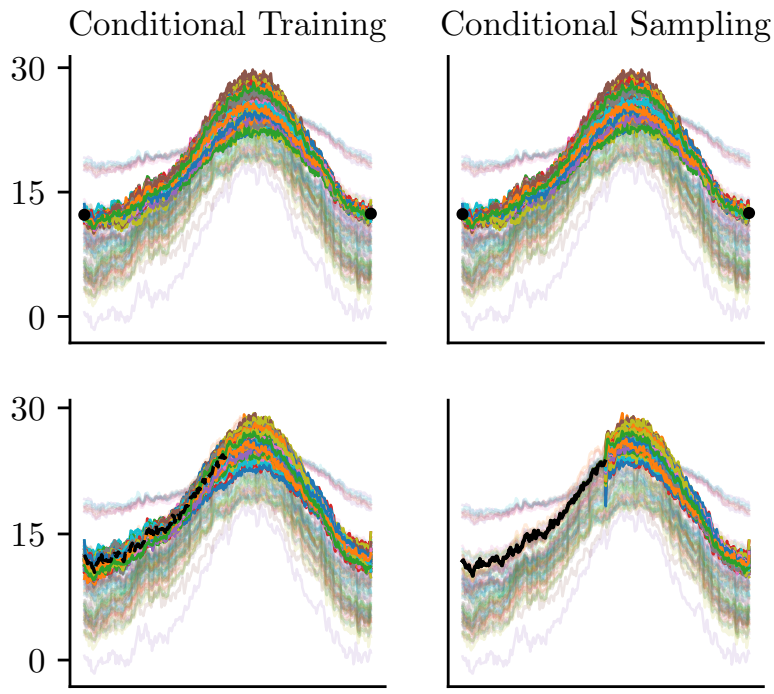


Figure 4.5: Conditional samples from the FFM-OT model. Darker curves indicate samples and lighter curves depict real data. Conditioning information is shown in black. The first column corresponds to a conditionally trained model and the second column corresponds to a conditionally trained model in addition to conditional sampling. We see that, while the conditionally trained model takes into account the conditioning information, the conditional sampling method allows us to enforce equality of the generated samples to the conditioning information at the observation locations.

Conditional Generation We also demonstrate an extension of our method for conditional tasks, such as interpolating (or extrapolating) a finite set of given observations. We explore two approaches: conditional training and a modified sampling process inspired by ILVR [Choi et al., 2021]. We note alternative conditional methods [Mathieu et al., 2023] are readily applicable as well. In Figure 4.5, we demonstrate these two approaches. See Appendix B.3 for details.

4.5 Conclusion

We introduce Functional Flow Matching (FFM), a continuous-time normalizing flow model which allows us to model infinite-dimensional distributions. We demonstrate that FFM is able to outperform several recently proposed function space generative models in terms of qualitative samples and pointwise metrics on a diverse set of benchmarks. Our work builds the foundations for function space normalizing flows, and our hope is that future work may build on these foundations. In terms of limitations, FFM is implemented via the FNO [Li et al., 2021], which can only handle data observed on uniform grids. Exploring architectures which alleviate this assumption may increase the applicability of our methods. Additionally, there are no established benchmarks for functional generation, unlike FID [Heusel et al., 2017] for images. Developing benchmarks for these tasks is critical for future work.

Chapter 5

Dynamic Conditional Optimal Transport through Simulation-Free Flows

In this chapter[1], we focus our study of transport-based generative models to the conditional setting. Many fundamental tasks in machine learning and statistics may be posed as modeling a conditional distribution $\nu(u \mid y)$, but where obtaining an analytical representation of $\nu(u \mid y)$ is often impractical.

While sampling-based approaches such as Markov Chain Monte Carlo (MCMC) methods are useful, they suffer from several limitations. First, MCMC requires numerous likelihood evaluations, rendering it prohibitively expensive in scientific and engineering applications where the likelihood is determined by an expensive numerical simulator. Second, MCMC must be run anew for every observation y, which is impractical in applications such as Bayesian inverse problems [Dashti and Stuart, 2013] and generative modeling [Mirza and

[1]The content of this chapter was previously published as *Dynamic Conditional Optimal Transport through Simulation-Free Flows* (NeurIPS 2024) [Kerrigan et al., 2024b], with minor modifications.

Osindero, 2014]. These limitations motivate the need for a *likelihood-free* [Cranmer et al., 2020] and *amortized* [Amos, 2023] approach. While methods like ABC [Beaumont, 2010] and variational inference [Blei et al., 2017] address these challenges, they are difficult to scale to high dimensions or have limited flexibility.

Recently, generative models such as normalizing flows [Papamakarios et al., 2019, 2021], GANs [Ramesh et al., 2022b], and diffusion models [Sharrock et al., 2024] have shown promise in amortized and likelihood-free inference. These models may be viewed in the framework of *measure transport* [Baptista et al., 2020], where samples $u \sim \eta(u)$ from a tractable source distribution are transformed by a mapping $T(y, u)$ such that the transformed samples are approximately distributed as $\nu(u \mid y)$. One way to achieve this is through *triangular* mappings [Baptista et al., 2020, Spantini et al., 2022], where a joint source distribution $\eta(y, u)$ is transformed by a mapping of the form $T : (y, u) \mapsto (T_Y(y), T_U(y, u))$. Under suitable assumptions, if T transforms the source $\eta(y, u)$ into the target $\nu(y, u)$, then $T_U(y, -)$ couples the conditionals $\eta(u \mid y)$ and $\nu(u \mid y)$.

Typically, such a map T is not unique [Wang et al., 2023], and a natural idea is thus to regularize the transport and search for an admissible mapping that is in some sense optimal. In other words, learning a conditional sampler may be phrased as finding a conditional optimal transport (COT) map. While there exists some work on learning COT maps, these approaches often rely on a difficult adversarial optimization problem [Baptista et al., 2020, Hosseini et al., 2023, Bunne et al., 2022, Ray et al., 2024] or simulating from the model during training [Baptista et al., 2023, Wang et al., 2023].

In this chapter, we propose a conditional generative model for likelihood-free inference based on a dynamic formulation of conditional optimal transport. In particular, we develop a general theoretical framework for dynamic conditional optimal transport in separable Hilbert spaces. Our framework is applicable in infinite-dimensional spaces, enabling applications in function space Bayesian inference. In Section 5.3, we study the *conditional Wasserstein*

space $\mathbb{P}_p^\mu(Y \times U)$ and show that this space admits constant speed geodesics between any two measures. In Section 5.4, we characterize the absolutely continuous curves of measures in $\mathbb{P}_p^\mu(Y \times U)$ via the continuity equation and *triangular* vector fields. As a consequence, we obtain conditional generalizations of the McCann interpolants [McCann, 1997] and the Benamou-Brenier Theorem [Benamou and Brenier, 2000].

In Section 5.5, we propose COT flow matching (COT-FM), a simulation-free flow-based model for conditional generation. This model directly leverages our theoretical framework, where we learn to model a path of measures interpolating between an arbitrary source and target distribution via a geodesic in the conditional Wasserstein space. Lastly, we demonstrate our method on several challenging conditional generation tasks. We apply our method to two Bayesian inverse problems – one arising from the Lotka-Volterra dynamical system, and an infinite-dimensional problem arising from the Darcy Flow PDE. Our method shows competitive performance against recent COT methods.

5.1 Related Work

Conditional Optimal Transport. Conditional Optimal Transport (COT) remains relatively under-explored in both machine learning and related fields. Recent approaches learn static COT maps via input convex networks [Bunne et al., 2022, Wang et al., 2023] or normalizing flows [Wang et al., 2023]. In addition, there have been a number of heuristic approaches to conditional simulation through W-GANs [Sajjadi et al., 2017, Adler and Öktem, 2018, Kim et al., 2022, 2023], for which Chemseddine et al. [2023] provide a rigorous basis. Closely related to our method are those which employ triangular plans [Carlier et al., 2016, Trigila and Tabak, 2016], which have been modeled through GANs in Euclidean spaces [Baptista et al., 2020] and function spaces [Hosseini et al., 2023]. In contrast, our work uses a novel *dynamic* formulation of COT, which we model through a generalization of flow matching

[Lipman et al., 2023, Albergo et al., 2024]. This allows us to use flexible architectures while avoiding the difficulties of training GANs [Arora et al., 2018].

Simulation-Free Continuous Normalizing Flows. Flow matching [Lipman et al., 2023] and stochastic interpolants [Albergo et al., 2023] are a class of methods for building continuous-time normalizing flows in a simulation-free manner. Notably, these works do not approximate an optimal transport between the source and target measures. Pooladian et al. [2023] and Tong et al. [2024] propose instead to couple the source and target distributions via optimal transport, leading to marginally optimal paths. In this work, we study an extension of these techniques for conditional generation.

While some works [Davtyan et al., 2023, Gebhard et al., 2023, Isobe et al., 2024, Wildberger et al., 2024] have applied flow matching for conditional generation, these approaches do not employ COT. Notably, the aforementioned approaches are limited to the finite-dimensional setting, whereas our method adds to the growing literature on function-space generative models [Hosseini et al., 2023, Kerrigan et al., 2023, 2024a, Lim et al., 2023a, Franzese et al., 2024]. Barboni et al. [2024] and Chemseddine et al. [2024] appeared concurrently to our work with similar results, but only study COT in the finite-dimensional setting.

5.2 Background and Notation

The notation in this chapter largely follows that outlined in Chapter 2, which we recall here with some additional material which is specific to this chapter. Let H, H' represent arbitrary separable Hilbert spaces, equipped with the Borel σ-algebra. We use $\mathbb{P}(H)$ to represent the space of Borel probability measures on H, and $\mathbb{P}_p(H) \subseteq \mathbb{P}(H)$ to represent the subspace of measures having finite pth moment. If $\eta \in \mathbb{P}(H)$ is a probability measure on H and $T : H \to H'$ is measurable, then the pushforward measure $T_{\#}\eta(-) = \eta(T^{-1}(-))$ is a

probability measure on H'. Maps of the form e.g. $\pi^H : H \times H' \to H$ represent the canonical projection.

In this chapter, we will focus on two separable Hilbert spaces of interest. The first, Y, is a space of observations, and the second, U, is a space of unknowns. These spaces may be of infinite dimensions, but a case of practical interest is when Y and U are finite dimensional Euclidean spaces. We will consider the product space $Y \times U$, equipped with the canonical inner product obtained via the sum of the inner products on Y and U, under which the space $Y \times U$ is also a separable Hilbert space. Let $\eta \in \mathbb{P}(Y \times U)$ be a joint probability measure. The measures $\pi^Y_\# \eta \in \mathbb{P}(Y)$ and $\pi^U_\# \eta \in \mathbb{P}(U)$ obtained via projection are the *marginals* of η. We use $\eta^y \in \mathbb{P}(U)$ to represent the measure obtained by conditioning η on the value $y \in Y$. By the disintegration theorem [Bogachev and Ruas, 2007, Chapter 10], such conditional measures exist and are essentially unique, in the sense that there exists a Borel set $E \subseteq Y$ with $\pi^Y_\# \eta(E) = 0$, and the η^y are unique for $y \notin E$.

5.2.1 Static Conditional Optimal Transport

In conditional optimal transport, we are given a target measure $\nu \in \mathbb{P}(Y \times U)$ and some source measure $\eta \in \mathbb{P}(U)$, and we seek a transport map $T : Y \times U \to U$ such that $T_\#(y, -)_\# \eta = \nu^y$ for all $y \in Y$. If such a map were available, by drawing samples $u_0 \sim \eta$ and transforming them, one would obtain samples $T(y, u) \sim \nu^y$. Solving this transport problem for each fixed y is expensive at best, or impossible when only has a single (or no) samples $(y, u) \sim \nu$ for any given y. Thus, one must leverage information across different observations y. To that end, recent work has focused on the notion of *triangular mappings* $T : Y \times U \to Y \times U$ [Hosseini et al., 2023, Baptista et al., 2020] of the form $T(y, u) = (T_Y(y), T_U(T_Y(y), u))$ for some $T_Y : Y \to Y$ and $T_U : Y \times U \to U$. Triangular mappings are of interest as they allow us to obtain conditional couplings from joint couplings.

Proposition 22 (Theorem 2.4 [Baptista et al., 2020], Prop. 2.3 [Hosseini et al., 2023])**.** *Suppose* $\eta, \nu \in \mathbb{P}(Y \times U)$ *and* $T : Y \times U \to Y \times U$ *is triangular. If* $T_{\#}\eta = \nu$*, then* $T_U(T_Y(y), -)_{\#}\eta^y = \nu^{T_Y(y)}$ *for* $\pi^Y_{\#}\eta$*-almost every* y*.*

In many scenarios of practical interest, the source measure η and the target measure ν have the same Y-marginals. We will henceforth make this assumption, and use $\mu = \pi^Y_{\#}\eta = \pi^Y_{\#}\nu$ to represent this marginal. In this case, we may take T_Y to be the identity mapping, so that the conclusion of Proposition 22 simplifies to $T_U(y, -)_{\#}\eta^y = \nu^y$ for μ-almost every y. We note that in situations where such an assumption does not hold, one may simply preprocess the source measure η via an invertible mapping T_Y satisfying $[T_Y]_{\#}[\pi^Y_{\#}\eta] = \pi^Y_{\#}\nu$ [Hosseini et al., 2023, Prop 3.2].

Given a source and target measures $\eta, \nu \in \mathbb{P}^{\mu}(Y \times U)$ and a cost function $c : (Y \times U)^2 \to \mathbb{R}$, the *conditional Monge problem* seeks to find a triangular mapping solving

$$\inf_T \left\{ \int_{Y \times U} c(y, u, T(y, u)) \, \mathrm{d}\eta(y, u) \mid T_{\#}\eta = \nu, T : (y, u) \mapsto (y, T_U(y, u)) \right\}. \tag{5.1}$$

The conditional Monge problem also admits a relaxation under which one only considers couplings whose Y-components are almost surely equal. To that end, for $\eta, \nu \in \mathbb{P}^{\mu}_p(Y \times U)$ we define the set of *triangular couplings* $\Pi_Y(\eta, \nu)$ to be the couplings of η and ν that almost surely fix the Y-components,

$$\Pi_Y(\eta, \nu) = \left\{ \gamma \in \mathbb{P}\left((Y \times U)^2\right) \mid \pi^{1,2}_{\#}\gamma = \eta, \pi^{3,4}_{\#}\gamma = \nu, \pi^{1,3}_{\#} = (I, I)_{\#}\mu \right\}. \tag{5.2}$$

In other words, a triangular coupling $\gamma \in \Pi_Y(\eta, \nu)$ has samples $(y_0, u_0, y_1, u_1) \sim \gamma$ such that $y_0 = y_1$ almost surely. The *conditional Kantorovich problem* seeks a triangular coupling

solving

$$\inf_{\gamma} \left\{ \int_{(Y\times U)^2} c(y_0, u_0, y_1, u_1)\, \mathrm{d}\gamma(y_0, u_0, y_1, u_1) \mid \gamma \in \Pi_Y(\eta, \nu) \right\}. \tag{5.3}$$

Hosseini et al. [2023] prove the existence of minimizers to the conditional Kantorovich and Monge problems under very general assumptions. Moreover, optimal couplings to the conditional Kantorovich problem induce optimal couplings for μ-almost every conditional measure. Assuming sufficient regularity assumptions on the conditional measures, unique solutions to the conditional Monge problem exist. We restate these results here for the sake of completeness.

Proposition 23 (Prop 3.3 [Hosseini et al., 2023])**.**
Fix $\eta, \nu \in \mathbb{P}^\mu(Y \times U)$. Suppose the cost function c is continuous, $\inf c > -\infty$, and there exists a finite cost coupling $\gamma \in \Pi_Y(\eta, \nu)$. Then, the conditional Kantorovich problem admits a minimizer $\gamma^\star$. Moreover, $\gamma^{\star,y_0}(y_1, u_0, u_1) = \hat{\gamma}^{\star,y_0}(u_0, u_1)\delta(y_1 - y_0)$ where for μ-almost every y the measure $\gamma^{\star,y}$ is an optimal coupling for η^y, ν^y under the cost $c^y(u_0, u_1) = c(y, u_0, y, u_1)$

Proposition 24 (Prop 3.8 [Hosseini et al., 2023])**.**
Fix $1 < p < \infty$ and $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$. Suppose $c(y_0, u_0, y_1, u_1) = |u_0 - u_1|^p$. If η^y assign zero measure to Gaussian null sets for μ-almost every y, then there is a unique solution $T^\star$ to the conditional Monge problem, and $\gamma^\star = (I, T^\star)_\# \eta$ is the unique solution to the conditional Kantorovich problem. If ν^y also assign zero measure to Gaussian null sets for μ-almost every y, then $T^\star$ is injective η-almost everywhere.

5.3 Conditional Wasserstein Space

Motivated by our discussion on triangular transport maps, we introduce the conditional Wasserstein spaces, consisting of joint measures with finite pth moments and having fixed

Y-marginals μ. Interestingly, Gigli [2008, Chapter 4] studies the same space for the purposes of constructing geometric tangent spaces in the usual Wasserstein space.

Definition 25 (Conditional Wasserstein Space)**.**
Suppose $\mu \in \mathbb{P}(Y)$ is given and $1 \leq p < \infty$. The conditional p-Wasserstein space is

$$\mathbb{P}_p^\mu(Y \times U) = \left\{\gamma \in \mathbb{P}_p(Y \times U) \mid \pi^Y_\# \gamma = \mu\right\}. \tag{5.4}$$

We now equip $\mathbb{P}_p^\mu(Y \times U)$ with a metric W_p^μ, the conditional Wasserstein distance. Intuitively, the conditional Wasserstein distance measures the usual Wasserstein distance between all of the conditional distributions in expectation under the fixed Y-marginal μ.

Definition 26 (Conditional p-Wasserstein Distance)**.**
Suppose $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$ and $1 \leq p < \infty$. The function $W_p^\mu : \mathbb{P}_p^\mu(Y \times U) \times \mathbb{P}_p^\mu(Y \times U) \to \mathbb{R}$,

$$W_p^\mu(\eta, \nu) = \left(\mathbb{E}_{y\sim\mu}\left[W_p^p(\eta^y, \nu^y)\right]\right)^{1/p} = \left(\int_Y W_p^p(\eta^y, \nu^y)\, \mathrm{d}\mu(y)\right)^{1/p} \tag{5.5}$$

is the conditional p-Wasserstein distance. W_p is the usual Wasserstein distance for measures on U.

By Jensen's inequality we have $W_p^\mu(\eta, \nu) \geq \mathbb{E}_{y\sim\mu}\left[W_p(\eta^y, \nu^y)\right]$. For $p > 1$, this inequality is strict unless $W_p(\eta^y, \nu^y)$ is μ-almost surely constant.

We note that $W_p^\mu(\eta, \nu)$ may be viewed as the minimal value of the constrained Kantorovich problem in Equation (5.3) when one takes the cost to be the metric on the space $Y \times U$. Similar results, relating the conditional Wasserstein distance to triangular couplings, have appeared previously, but our proof is independent of these prior works [Chemseddine et al., 2023, Gigli, 2008].

Proposition 27 (Equivalent Formulation of the Conditional Wasserstein Distance)**.**

Fix $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$ and $1 \leq p < \infty$. Then, $W_p^\mu(\eta, \nu)$ is well-defined, finite, and

$$W_p^{\mu,p}(\eta, \nu) = \min_\gamma \left\{ \int_{(Y\times U)^2} d^p(y_0, u_0, y_1, u_1) \, \mathrm{d}\gamma \mid \gamma \in \Pi_Y(\eta, \nu) \right\} \tag{5.6}$$

where $W_p^{\mu,p}(\eta, \nu)$ represents the p-th power of the conditional p-Wasserstein distance.

Proof. The cost function d^p is clearly continuous and non-negative, and hence by Proposition 23 it suffices to exhibit a finite-cost coupling $\gamma \in \Pi_Y(\eta, \nu)$ between η and ν. Indeed, take the conditionally independent coupling

$$\gamma(y_0, u_0, y_1, u_1) = \eta(u_0 \mid y_1)\nu(u_1 \mid y_1)\delta(y_1 - y_0)\mu(y_1) \tag{5.7}$$

which is clearly in $\Pi_Y(\eta, \nu)$. We then have that

$$\begin{aligned}
&\int_{(Y\times U)^2} d^p(y_0, u_0, y_1, u_1) \, \mathrm{d}\gamma(y_0, u_0, y_1, u_1) = \int_{(Y\times U)^2} \|(y_0, u_0) - (y_1, u_1)\|_{Y\times U}^p \, \mathrm{d}\gamma(y_0, u_0, y_1, u_1) \\
&\leq 2^p \int_{(Y\times U)^2} \left(\|(y_0, u_0)\|_{Y\times U}^p + \|(y_1, u_1)\|_{Y\times U}^p \right) \mathrm{d}\gamma(y_0, u_0, y_1, u_1) \\
&= 2^p \left(\int_{Y\times U} \|(y_0, u_0)\|_{Y\times U}^p \, \mathrm{d}\eta(y_0, u_0) + \int_{Y\times U} \|(y_1, u_1)\|_{Y\times U}^p \, \mathrm{d}\nu(y_1, u_1) \right) < +\infty.
\end{aligned}$$

Hence, Equation (5.6) admits a minimizer $\gamma^\star \in \Pi_Y(\eta, \nu)$. By Proposition 23, this minimizer may be taken to have the form $\gamma^\star = \gamma^{\star,y_1}(u_0, u_1)\delta(y_1 - y_0)\mu(y_1)$ where $\gamma^{\star,y_1}(u_0, u_1)$ is $\mu(y_1)$-almost surely an optimal coupling between η^{y_1}, ν^{y_1} for the cost $|u_1 - u_0|^p$. Thus,

$$\begin{aligned}
\int_{(Y\times U)^2} d^p \, \mathrm{d}\gamma^\star &= \int_Y \int_{U^2} |u_1 - u_0|^p \, \mathrm{d}\gamma^{\star,y}(u_0, u_1) \, \mathrm{d}\mu(y) &(5.8)\\
&= \int_Y W_p^p(\eta^y, \nu^y) \, \mathrm{d}\mu(y) = W_p^{p,\mu}(\eta, \nu). &(5.9)
\end{aligned}$$

Here, we emphasize that the μ-almost sure uniqueness of the disintegrations of η, ν along Y

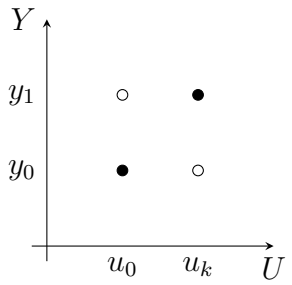


Figure 5.1: The counterexample in Proposition 28. The measure η_k is shown in black and the measure ν_k is shown in white.

result in a well-defined expression.

Moreover, if $\eta \in \mathbb{P}_p^\mu(Y \times U)$ it follows that $\eta^y \in \mathbb{P}_p(U)$ for μ-a.e. y, because

$$\int_Y \int_U |u|^p \,\mathrm{d}\eta^y(u) \,\mathrm{d}\mu(y) \leq \int_Y \int_U |(y,u)|^p \,\mathrm{d}\eta^y(u) \,\mathrm{d}\mu(y) \tag{5.10}$$

$$= \int_{Y\times U} |(y,u)|^p \,\mathrm{d}\eta(y,u) < +\infty. \tag{5.11}$$

Thus all considered p-Wasserstein distances on U are finite. □

In the following, we show that the conditional Wasserstein distance is a well-defined metric as well as a few other metric properties.

Proposition 28 (Some Properties of W_p^μ)**.**

Let $1 \leq p < \infty$.

1. *W_p^μ is well-defined, finite, and equals the minimal conditional Kantorovich cost.*
2. *W_p^μ is a metric on the space $\mathbb{P}_p^\mu(Y \times U)$.*
3. *There does not exist $C > 0$ such that $W_p^\mu(\eta,\nu) \leq C W_p(\eta,\nu)$ for all $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$.*
4. *For all $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$, $W_p\left(\pi^U_\# \eta, \pi^U_\# \nu\right) \leq W_p^\mu(\eta,\nu)$ and $W_p(\eta,\nu) \leq W_p^\mu(\eta,\nu)$.*

Proof. **Part (a).** This is simply a restatement of Proposition 27.

Part (b). Fix $\eta, \nu, \rho \in \mathbb{P}_p^\mu(Y \times U)$. Since W_p is a metric on $\mathbb{P}_p(U)$, we immediately obtain the symmetry of W_p^μ. Moreover, we have that $W_p^\mu(\eta, \nu) = 0$ if and only if $\eta^y = \nu^y$ for μ-almost every y. Thus, if $W_p^\mu(\eta, \nu) = 0$ and $E \subseteq Y \times U$ is Borel measurable,

$$\eta(E) = \int_Y \eta^y(E^y)\,\mathrm{d}\mu(y) = \int_Y \nu^y(E^y)\,\mathrm{d}\mu(y) = \nu(E). \tag{5.12}$$

which shows that $\eta = \nu$. Here, $E^y = \{u \mid (y, u) \in E\}$ is the y-slice of E. Conversely, if $\eta = \nu$, then $\eta^y = \nu^y$ up to a μ-null set by the essential uniqueness of disintegrations. Thus, $W_p^\mu(\eta, \nu) = 0$ if and only if $\eta = \nu$.

By Minkowski's inequality and the triangle inequality for W_p on $\mathbb{P}_p(U)$, we see

$$W_p^\mu(\eta, \nu) \leq \left(\mathbb{E}_{y\sim\mu}\left[(W_p(\eta^y, \rho^y) + W_p(\rho^y, \nu^y))^p\right]\right)^{1/p} \tag{5.13}$$

$$\leq \mathbb{E}_{y\sim\mu}[W_p^p(\eta^y, \rho^y)]^{1/p} + \mathbb{E}_{y\sim\mu}[W_p^p(\rho^y, \nu^y)]^{1/p} \tag{5.14}$$

$$= W_p^\mu(\eta, \rho) + W_p^\mu(\rho, \nu). \tag{5.15}$$

Part (c). We provide a counterexample. Fix any $u_0 \neq 0 \in U$ and $y_0, y_1 \in Y$ such that $y_0 \neq y_1$. Define $\mu = \frac{1}{2}(\delta_{y_0} + \delta_{y_1})$. Set $u_k = (k+1)u_0$ for $k = 1, 2, \ldots$ and for each k, define two measures on $Y \times U$ by

$$\eta_k = \frac{1}{2}(\delta_{y_0 u_0} + \delta_{y_1 u_k}) \qquad \nu_k = \frac{1}{2}(\delta_{y_1 u_0} + \delta_{u_k y_0}). \tag{5.16}$$

It is clear that

$$W_p^{\mu,p}(\eta_k, \nu_k) = k^p |u_0|^p \qquad W_p^p(\eta_k, \nu_k) = \min\{k^p |u_0|^p, |y_1 - y_0|^p\}. \tag{5.17}$$

Moreover, as $k \to \infty$ we have $W_p^\mu(\mu_k, \nu_k) \to \infty$ but $W_p^p(\nu_k, \eta_k)$ remains bounded. See Figure 5.1.

Part (d). First, the unconditional distance $W_p(\eta, \nu)$ may be obtained via an unrestricted coupling in $\Pi(\eta, \nu)$, i.e. the set of all joint measures on $Y \times U$ having marginals η, ν. Since $\Pi(\eta, \nu) \supseteq \Pi_Y(\eta, \nu)$, by part (a) we see that $W_p(\eta, \nu) \leq W_p^\mu(\eta, \nu)$.

Let $\gamma^\star(y_0, u_0, y_1, u_1) = \gamma^{\star, y_1}(u_0, u_1)\delta(y_1 - y_0)\mu(y_1)$ be an optimal $\gamma^\star \in \Pi_Y(\eta, \nu)$. We claim that $\gamma(u_0, u_1) := \int_Y \gamma^{\star,y}(u_0, u_1)\,\mathrm{d}\mu(y)$ couples $\pi^U_\# \eta$ and $\pi^U_\# \nu$. Let $\pi^0 : (u_0, u_1) \mapsto u_0$ be the projection onto the first coordinate of $U \times U$. Observe that for μ-almost every y, we have that $\gamma^{\star,y} \in \Pi(\eta^y, \nu^y)$ is optimal, and, in particular, $\pi^0_\# \gamma^{\star,y} = \eta^y$. Fix an arbitrary $\varphi \in C_b(U)$. We then have

$$\int_U \varphi(u_0)\,\mathrm{d}\pi^0_\#\gamma(u_0) = \int_{U^2} (\varphi \circ \pi^0)\,\mathrm{d}\gamma(u_0, u_1) \tag{5.18}$$

$$= \int_Y \int_{U^2} (\varphi \circ \pi^0)\,\mathrm{d}\gamma^{\star,y}(u_0, u_1)\,\mathrm{d}\mu(y) = \int_Y \int_U \varphi(u_0)\,\mathrm{d}\pi^0_\#\gamma^{\star,y}(u_0)\,\mathrm{d}\mu(y) \tag{5.19}$$

$$= \int_Y \int_U \varphi(u_0)\,\mathrm{d}\eta^y(u_0)\,\mathrm{d}\mu(y) = \int_{Y\times U} \varphi(u_0)\,\mathrm{d}\eta(u_0, y) \tag{5.20}$$

$$= \int_{Y\times U} (\varphi \circ \pi^U)\,\mathrm{d}\eta(u_0, y) = \int_U \varphi\,\mathrm{d}\pi^U_\#\eta(u_0). \tag{5.21}$$

Thus $\pi^U_\#\gamma = \pi^U_\#\eta$. A similar argument shows that for the map $\pi^1 : (u_0, u_1) \mapsto u_1$ we have $\pi^1_\#\gamma = \pi^U_\#\nu$, so that $\gamma \in \Pi(\pi^U_\#\eta, \pi^U_\#\nu)$.

Now, as $\gamma^{\star,y_1}(u_0, u_1) \in \Pi(\eta^{y_1}, \nu^{y_1})$ is μ-almost surely optimal in the usual Wasserstein sense,

$$W_p^{p,\mu}(\eta, \nu) = \int_Y \int_{U^2} |u_0 - u_1|^p\,\mathrm{d}\gamma^{\star,y}(u_0, u_1)\,\mathrm{d}\mu(y) \tag{5.22}$$

$$= \int_{U^2} |u_0 - u_1|^p\,\mathrm{d}\gamma(u_0, u_1) \tag{5.23}$$

$$\geq W_p^p(\pi^U_\#\eta, \pi^U_\#\nu) \tag{5.24}$$

since $\gamma \in \Pi(\pi^U_\# \eta, \pi^U_\# \nu)$ is a coupling but potentially sub-optimal.

□

Proposition 28(c, d) together shows that one should expect the topology generated by W_p^μ to be stronger than the unconditional distance W_p. Here, we note that Gigli [2008] and Chemseddine et al. [2023] previously showed that W_p^μ is a metric through an equivalence with restricted couplings. Our approach builds on the results of Hosseini et al. [2023] and is somewhat more direct, and hence our proofs may be of independent interest. We include here an example where the conditional 2-Wasserstein distance may be explicitly computed.

Example: Gaussian Measures. Suppose Y and U are Euclidean spaces (of possibly different dimensions), and that $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$ are Gaussians of the form

$$\eta = \mathcal{N}\left(\begin{bmatrix} m \\ m_u^\eta \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_{12}^\eta \\ \Sigma_{21}^\eta & \Sigma_{22}^\eta \end{bmatrix}\right) \qquad \nu = \mathcal{N}\left(\begin{bmatrix} m \\ m_u^\nu \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_{12}^\nu \\ \Sigma_{21}^\nu & \Sigma_{22}^\nu \end{bmatrix}\right). \tag{5.25}$$

This form is chosen to ensure that η and ν have equal Y-marginals. It follows that $\mu = \pi^Y_\# \eta = \pi^Y_\# \nu = \mathcal{N}(m, \Sigma)$. Let

$$Q^\eta = \Sigma_{22}^\eta - \Sigma_{21}^\eta \Sigma^{-1} \Sigma_{12}^\eta \qquad Q^\nu = \Sigma_{22}^\nu - \Sigma_{21}^\nu \Sigma^{-1} \Sigma_{12}^\nu \qquad R = (\Sigma_{21}^\eta - \Sigma_{21}^\nu)\Sigma^{-1}. \tag{5.26}$$

We have that the conditionals η^y, ν^y are available in closed-form:

$$\eta^y = \mathcal{N}\left(m_u^\eta + \Sigma_{21}^\eta \Sigma^{-1}(y - m), Q^\eta\right) \qquad \nu^y = \mathcal{N}\left(m_u^\nu + \Sigma_{21}^\nu \Sigma^{-1}(y - m), Q^\nu\right). \tag{5.27}$$

Thus, for any fixed y, we use the known closed-form unconditional Wasserstein distance to obtain

$$W_2^2(\eta^y, \nu^y) = \left|m_u^\eta - m_u^\nu + R(y-m)\right|^2 + \mathrm{Tr}\left(Q^\eta + Q^\nu - 2\left((Q^\eta)^{1/2} Q^\nu (Q^\eta)^{1/2}\right)^{1/2}\right). \quad (5.28)$$

We now take an expectation over $y \sim \mu = \mathcal{N}(m, \Sigma)$ to compute $W_2^{\mu,2}$. Observe that $R(y-m) \sim \mathcal{N}(0, R\Sigma R^\mathsf{T})$ and that $\mathbb{E}_{y\sim\mu}[|R(y-m)|^2] = \mathrm{Tr}(R\Sigma R^\mathsf{T})$. Thus,

$$\begin{aligned} W_2^{\mu,2}(\eta,\nu) &= \mathbb{E}_{y\sim\mu}\left[W_2^2(\eta^y,\nu^y)\right] && (5.29)\\ &= \mathbb{E}_{y\sim\mu}\left[|m_u^\eta - m_u^\nu|^2 + 2\langle m_u^\eta - m_u^\nu, R(y-m)\rangle + |R(y-m)|^2\right] && (5.30)\\ &\quad + \mathrm{Tr}\left(Q^\eta + Q^\nu - 2\left((Q^\eta)^{1/2} Q^\nu (Q^\eta)^{1/2}\right)^{1/2}\right) \\ &= |m_u^\eta - m_u^\nu|^2 + \mathrm{Tr}\left(Q^\eta + Q^\nu - 2\left((Q^\eta)^{1/2} Q^\nu (Q^\eta)^{1/2}\right)^{1/2} + R\Sigma R^T\right). && (5.31) \end{aligned}$$

This form, perhaps unsurprisingly, closely resembles the unconditional Wasserstein distance between two Gaussians, except for the presence of an additional $\mathrm{Tr}(R\Sigma R^\mathsf{T})$ term. Note that when η, ν have uncorrelated Y, U components, we precisely recover $W_2^2(\pi^U_\# \eta, \pi^U_\# \nu)$ as one may expect.

As a special case of interest, if $Y = U = \mathbb{R}$ and

$$\eta = \mathcal{N}(0, I) \qquad \nu = \mathcal{N}\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right) \quad |\rho| < 1 \quad (5.32)$$

then we obtain as a special case of Equation (5.29) that $W_2^{\mu,2}(\eta,\nu) = 2(1 - \sqrt{1-\rho^2})$. This is zero if and only if $\rho = 0$, i.e. $\eta = \nu$. However, $\pi^U_\# \eta = \pi^U_\# \nu = \mathcal{N}(0,1)$ and $W_2(\pi^U_\# \eta, \pi^U_\# \nu) = 0$ regardless of ρ. Moreover, the unconditional distance is $W_2^2(\eta,\nu) = 2\left(2 - \sqrt{1-\rho} - \sqrt{1+\rho}\right)$, from which it is easy to verify that $W_2(\eta,\nu) \le W_2^\mu(\eta,\nu)$.

Conditional Wasserstein Space as a Geodesic Space. We now turn our attention to the geodesics in $\mathbb{P}_p^\mu(Y \times U)$. In particular, we show that there exists a constant speed geodesic between any two measures in $\mathbb{P}_p^\mu(Y \times U)$, generalizing a similar result in the unconditional setting [Santambrogio, 2015, Theorem 5.27]. Moreover, we show that under suitable regularity assumptions, solutions to the conditional Monge problem (5.1) induce constant speed geodesics. Our motivation for studying geodesics in $\mathbb{P}_p^\mu(Y \times U)$ is practical – in Section 5.5, we show how one can model geodesics in $\mathbb{P}_p^\mu(Y \times U)$ in order to obtain a conditional flow-based model whose paths are easy to integrate.

A *curve* is a continuous function $\gamma_\bullet : I \to \mathbb{P}_p^\mu(Y \times U)$ where $I = (a, b) \subseteq \mathbb{R}$ is any open interval of finite length. If (γ_t) is an absolutely continuous curve, then its metric derivative $|\gamma'|(t)$ [Ambrosio et al., 2005, Chapter 1] exists for almost every $t \in (a, b)$. A curve (γ_t) is called a *constant speed geodesic* if for all $a < s \leq t < b$, we have $W_p^\mu(\gamma_s, \gamma_t) = |t - s| W_p^\mu(\gamma_a, \gamma_b)$. It is straightforward to show that every constant speed geodesic is absolutely continuous.

Theorem 29 ($\mathbb{P}_p^\mu(Y \times U)$ is a Geodesic Space)**.**
For any $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$, there exists a constant speed geodesic between η and ν.

Proof. Write $\lambda_t : (Y \times U)^2 \to Y \times U$ for the linear interpolant

$$\lambda_t(y_0, u_0, y_1, u_1) = (ty_0 + (1-t)y_1, tu_0 + (1-t)u_1) \qquad 0 \leq t \leq 1. \tag{5.33}$$

Let $\gamma^\star \in \Pi_Y(\eta, \nu)$ be an optimal restricted coupling, and consider the path of measures in $\mathbb{P}_p(Y \times U)$ given by

$$\gamma_t = [\lambda_t]_\# \gamma^\star \qquad 0 \leq t \leq 1. \tag{5.34}$$

Step one: We check that for each $0 \leq t \leq 1$, we have $\gamma_t \in \mathbb{P}_p^\mu(Y \times U)$. That is, we need to check that for all Borel $A \subseteq Y$, we have $\gamma_t(A \times U) = \mu(A)$. Indeed, recall that restricted

measures are concentrated on the set

$$\mathscr{C} := \left\{(y_0, u_0, y_1, u_1) \in (Y \times U)^2 \mid y_0 = y_1\right\}. \tag{5.35}$$

Thus, we may evaluate the measure γ_t on $A \times U$ to see

$$\begin{aligned}\gamma_t(A \times U) &= \gamma^\star\left\{\lambda_t^{-1}(A \times U)\right\} = \gamma^\star\left\{(y, u_0, y, u_1) \mid y \in A\right\} \\ &= \pi^1_\# \gamma^\star(A) = (\pi^1 \circ \pi^{1,2})_\# \gamma^\star(A) \\ &= \pi^1_\# \eta(A) = \mu(A)\end{aligned}$$

i.e. $\gamma_t(A \times Y) = \mu(A)$ as claimed.

Step two: We show that $W_p^\mu(\gamma_t, \gamma_s) = |t-s| W_p^\mu(\eta, \nu)$. Set $\gamma_t^s := (\lambda_t, \lambda_s)_\# \gamma^\star$ for $0 \leq s < t \leq 1$. We claim $\gamma_t^s \in \Pi_Y(\gamma_t, \gamma_s)$. Indeed, we have $\pi^{1,2}_\# \gamma_t^s = \gamma_t$ because for all Borel $A \subseteq Y \times U$,

$$(\lambda_t, \lambda_s)_\# \gamma^\star\left(A \times Y \times U\right) = \gamma^\star\left(\lambda_t^{-1}(A)\right) = (\lambda_t)_\# \gamma^*(A). \tag{5.36}$$

An analogous calculation shows that $\pi^{3,4}_\# \gamma_t^s = \gamma_s$, so that $\gamma_t^s \in \Pi(\gamma_t, \gamma_s)$. We now check that $\gamma_t^s \in \mathcal{R}_Y(Y \times U)$. Indeed, suppose $E \subseteq Y \times U$ is a Borel set such that $E \cap \mathscr{C} = \varnothing$. In other words, for every $(y_0, u_0, y_1, u_1) \in E$ we have $y_0 \neq y_1$. Set $D := (\lambda_t, \lambda_s)^{-1}(E)$. We claim $D \cap \mathscr{C} = \varnothing$, so that

$$\gamma_t^s(E) = (\lambda_t, \lambda_s)_\# \gamma^\star(E) = \gamma^\star((\lambda_t, \lambda_s)^{-1}(E)) \tag{5.37}$$

$$= \gamma^\star(D \cap \mathscr{C}) = 0. \tag{5.38}$$

Indeed, if $c = (y, u_0, y, u_1) \in \mathscr{C}$, then

$$(\lambda_t, \lambda_s)(c) = (y, tu_0 + (1-t)u_1, y, su_0 + (1-s)u_1) \notin E \tag{5.39}$$

$$\implies c \notin (\pi_t, \pi_s)^{-1}(E). \tag{5.40}$$

Thus $\gamma_t^s \in \Pi_Y(\eta, \nu)$ as claimed. Now, we have

$$\begin{aligned}
W_p^{\mu,p}(\gamma_t, \gamma_s) &\leq \int_{(Y\times U)^2} d^p\left(y_0, u_0, y_1, u_1\right) \, \mathrm{d}\lambda_t^s(y_0, u_0, y_1, u_1) \\
&= \int_{(Y\times U)^2} d^p\left(\lambda_t(y_0, u_0, y_1, u_1), \lambda_s(y_0, u_0, y_1, u_1)\right) \, \mathrm{d}\gamma^\star(y_0, u_0, y_1, u_1) \\
&= \int_{(Y\times U)^2} \left(|(t-s)(y_0 - y_1)|^2 + |(t-s)(u_0 - u_1)|^2\right)^{p/2} \, \mathrm{d}\gamma^\star(y_0, u_0, y_1, u_1) \\
&= |t-s|^p \int_{(Y\times U)^2} d^p(y_0, u_0, y_1, u_1) \, \mathrm{d}\gamma^\star(y_0, u_0, y_1, u_1) \\
&= |t-s|^p W_p^{\mu,p}(\eta, \nu).
\end{aligned}$$

Conversely, an application of the previous inequality and the triangle inequality show that for $0 \leq s \leq t \leq 1$,

$$W_p^\mu(\eta, \nu) \leq W_p^\mu(\eta, \gamma_s) + W_p^\mu(\gamma_s, \gamma_t) + W_p^\mu(\gamma_t, \nu) \tag{5.41}$$

$$\leq s W_p^\mu(\eta, \nu) + W_p^\mu(\gamma_s, \gamma_t) + (1-t) W_p^\mu(\eta, \nu). \tag{5.42}$$

Rearranging the previous inequality implies $|t-s| W_p^\mu(\eta, \nu) \leq W_p^\mu(\gamma_s, \gamma_t)$ for all $s, t \in [0,1]$, and hence $W_p^\mu(\gamma_t, \gamma_s) = |t-s|^p W_p^\mu(\eta, \nu)$. □

When an optimal triangular coupling $\gamma^\star \in \Pi_Y(\eta, \nu)$ is induced by an injective triangular map $T^\star$, we may recover a constant speed geodesic in $\mathbb{P}_p^\mu(Y \times U)$, generalizing the McCann interpolant [McCann, 1997] to the conditional setting. We refer to Proposition 24 for sufficient conditions on η, ν under which such a $T^\star$ exists. Informally, samples from $(y_0, u_0) \sim \eta$ flow

in a straight path at a constant speed to their destination $T^\star(y_0, u_0)$.

Theorem 30 (Conditional McCann Interpolants)**.**
Fix $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$. Suppose $T^\star(y,u) = (y, T_{\mathcal{U}}^\star(y,u))$ is an injective triangular map solving the conditional Monge problem (5.1)*. Define the maps $T_t : Y \times U \to Y \times U$ for $0 \le t \le 1$ via $T_t = (1-t)I + tT^\star$, and define the curve of measures $\gamma_t = [T_t]_\# \eta \in \mathbb{P}_p^\gamma(Y \times U)$. Then,*

1. *(γ_t) is absolutely continuous and a constant speed geodesic between η, ν*
2. *The vector field $v_t(T_t^\star(y,u)) = (0, T_U^\star(y,u) - u)$ generates the path γ_t, in the sense that (γ_t, v_t) solve the continuity equation* (5.53)*.*

Proof. Consider the function $w_t : Y \times U \to U$ given by

$$w_t(y,u) = (0, T_U^\star(y,u) - u) = (0, w_{t,U}(y,u)) \tag{5.43}$$

and note this is precisely $w_t(y,u) = \partial_t T_t^\star(y,u)$. Define the vector field

$$v_t(y,u) = \left(w_t \circ T_t^{\star,-1}\right)(y,u) = \left(0, (w_{t,\mathcal{U}} \circ T_{t,\mathcal{U}}^{\star,-1})(y,u)\right). \tag{5.44}$$

For any $\varphi \in \mathrm{Cyl}(Y \times U)$, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \int_{Y\times U} \varphi(y,u)\, \mathrm{d}\gamma_t(y,u) &= \frac{\mathrm{d}}{\mathrm{d}t} \int_{Y\times U} \varphi(y,u)\, \mathrm{d}[T_t]_\# \eta(y,u) && (5.45)\\
&= \frac{\mathrm{d}}{\mathrm{d}t} \int_{Y\times U} \varphi(y, T_{t,U}^\star(y,u))\, \mathrm{d}\eta(y,u) && (5.46)\\
&= \int_{Y\times U} \langle \nabla\varphi(y, T_{t,U}^\star(y,u), w_t(y,u))\rangle\, \mathrm{d}\eta(y,u) && (5.47)\\
&= \int_{Y\times U} \langle \nabla\varphi(y,u), v_t(y,u)\rangle\, \mathrm{d}\gamma_t(y,u) && (5.48)
\end{aligned}$$

which shows that (γ_t, v_t) solve the continuity equation.

Now, note that for $0 \leq a \leq b \leq 1$, we have

$$\int_a^b \|v_t\|_{L^p(\gamma_t, Y\times U)} \,\mathrm{d}t = \int_a^b \left(\int_{Y\times U} \left| w_t \circ T_t^{\star,-1} \right|^p (y,u) \,\mathrm{d}\gamma_t(y,u) \right)^{1/p} \mathrm{d}t \tag{5.49}$$

$$= \int_a^b \left(\int_{Y\times U} |w_t|^p(y,u) \,\mathrm{d}\eta(y,u) \right)^{1/p} \mathrm{d}t \tag{5.50}$$

$$= \int_a^b \left(\int_{Y\times U} |u - T_U^\star(y,u)|^p(y,u) \,\mathrm{d}\eta(y,u) \right)^{1/p} \mathrm{d}t \tag{5.51}$$

$$= (b-a) W_p^\mu(\eta,\nu). \tag{5.52}$$

In particular, $\int_0^1 \|v_t\|_{L^p(\gamma_t, Y\times U)} \,\mathrm{d}t < \infty$ and so by Theorem 32 (γ_t) is absolutely continuous. A similar calculation shows that $(b-a)W_p^\mu(\eta,\nu) = W_p^\mu(\gamma_b,\gamma_a) = \int_a^b |\gamma'(t)|$, where the last line follows from the absolute continuity of γ_t. Thus, $\|v_t\|_{L^p(\gamma_t, Y\times U)} = |\gamma'|(t)$ for almost every $t \in [0,1]$ by Lebesgue differentiation.

□

5.4 Conditional Benamou-Brenier Theorem

In this section, we prove a characterization of the absolutely continuous curves in $\mathbb{P}_p^\mu(Y \times U)$. As a corollary, we obtain a conditional generalization of the Benamou-Brenier Theorem [Benamou and Brenier, 2000], giving us a dynamical characterization of the conditional Wasserstein distance. Roughly speaking, all such curves are generated by a vector field on $Y \times U$ which has zero velocity in the Y component. This is natural, as all measures in $\mathbb{P}_p^\mu(Y \times U)$ have a fixed Y-marginal μ. Such a vector field can be informally seen as tangent to a curve of measures, and is the dynamic analogue of the triangular maps discussed in Section 5.2. More formally, given an open interval $I \subseteq \mathbb{R}$, a time-dependent Borel vector field $v : I \times Y \times U \to Y \times U$ is said to be *triangular* if there exists a Borel vector field

$v^U : I \times Y \times U \to U$ such that $v_t(y,u) = \big(0, v_t^U(y,u)\big)$.

Continuity Equation. We introduce some necessary background which allows us to link vector fields to curves of measures. The *continuity equation* $\partial_t \gamma_t + \operatorname{div}(v_t \gamma_t) = 0$ describes the evolution of a measure γ_t which flows along a given vector field v_t [Ambrosio et al., 2005, Chapter 8]. This equation must be understood distributionally, i.e. for every φ in an appropriate space of test functions,

$$\int_I \int_{Y \times U} \left(\partial_t \varphi(y,u,t) + \langle v_t(y,u), \nabla_{y,u} \varphi(y,u,t) \rangle\right) \, \mathrm{d}\gamma_t(y,u) \, \mathrm{d}t = 0. \tag{5.53}$$

We consider cylindrical test functions $\varphi \in \operatorname{Cyl}(Y \times U \times I)$, i.e. of the form $\varphi(y,u,t) = \psi(\pi^d(y,u),t)$ where $\pi^d : Y \times U \to \mathbb{R}^d$ maps $(y,u) \to (\langle (y,u), e_1 \rangle, \ldots, \langle (y,u), e_d \rangle)$ where $\{e_1, e_2, \ldots, e_d\}$ is any orthonormal family in $Y \times U$. In the finite dimensional setting, one may take $\varphi \in C_c^\infty(Y \times U)$ to be smooth and compactly supported [Ambrosio et al., 2005, Remark 8.1.1].

We now prove Lemma 3, which is key in proving Theorem 32 below. Informally, Lemma 3 states that if (5.53) is satisfied for a joint distribution and triangular vector field, then the continuity equation is also satisfied for the corresponding conditional distributions and U components of the vector field.

Lemma 3 (Triangular Vector Fields Preserve Conditionals)**.**
Suppose $v_t(y,u) = (0, v_t^U(y,u))$ is triangular and that $(\gamma_t) \subset \mathbb{P}_p^\mu(Y \times U)$ is a path of measures such that (v_t, γ_t) satisfy the continuity equation in the distributional sense. Then, it follows that for μ-almost every $y \in Y$, we have $\partial_t \gamma_t^y + \nabla \cdot (v_t^U(y,-)\gamma_t^y) = 0$.

Proof. Fix any $\varphi \in \operatorname{Cyl}(U \times I)$. Suppose $\psi \in \operatorname{Cyl}(Y)$ is given, and note that $\psi(y)\varphi(u,t) \in \operatorname{Cyl}(Y \times U \times I)$. As (v_t, γ_t) solve the continuity equation, it follows from the triangular

structure of v_t that upon testing against $\psi\varphi$ we have

$$\int_I \int_Y \psi(y) \int_U \left(\partial_t \varphi(u,t) + \langle v_t^U(y,u), \nabla_u \varphi(u,t)\rangle\right) \mathrm{d}\gamma_t^y(u)\,\mathrm{d}\mu(y)\,\mathrm{d}t = 0. \tag{5.54}$$

Because $\psi(y) \in \mathrm{Cyl}(Y)$, it is of the form $\rho(\pi(y))$ where $\pi : Y \to \mathbb{R}^k$ for some $k \geq 1$ and $\rho \in C_c^\infty(\mathbb{R}^k)$. Taking ρ to be a sequence of smooth approximations to the indicator function of an arbitrary rectangle $E = E_1 \times E_2 \times \cdots \times E_k \subseteq \mathbb{R}^k$, we see

$$\int_{\pi^{-1}(E)} \int_I \int_U \left(\partial_t \varphi(u,t) + \langle v_t^U(y,u), \nabla_u \varphi(u,t)\rangle\right) \mathrm{d}\gamma_t^y(u)\,\mathrm{d}t\,\mathrm{d}\mu(y) = 0. \tag{5.55}$$

As Y is separable, the Borel σ-algebra on Y is generated by the cylinder sets, i.e. those which are precisely of the form $\pi^{-1}(E)$ for some finite-dimensional rectangle E. We have thus shown that for an arbitrary Borel measurable set $E \subseteq Y$,

$$\int_E \int_I \int_U \left(\partial_t \varphi(u,t) + \langle v_t^U(y,u), \nabla_u \varphi(u,t)\rangle\right) \mathrm{d}\gamma_t^y(u)\,\mathrm{d}t\,\mathrm{d}\mu(y) = 0. \tag{5.56}$$

From this, it follows that

$$\int_I \int_U \left(\partial_t \varphi(u,t) + \langle v_t^U(y,u), \nabla_u \varphi(u,t)\rangle\right) \mathrm{d}\gamma_t^y(u)\,\mathrm{d}\mu(y)\,\mathrm{d}t = 0 \qquad \mu\text{-almost every } y. \tag{5.57}$$

□

Absolutely Continuous Curves. In this section, we state our characterization of absolutely continuous curves in $\mathbb{P}_p^\mu(Y \times U)$. Informally, given such a curve, Theorem 31 provides us with a triangular vector field which generates the curve, in the sense that the pair solve the continuity equation.

We now proceed to prove the main results of this section. First, we introduce some preliminary

notions. We define the map $j_q : L^q(\gamma, Y \times U) \to L^p(\gamma, Y \times U)$ for $1/p + 1/q = 1$ via

$$j_q(w) = \begin{cases} |w|^{q-2} w & w \neq 0 \\ 0 & w = 0 \end{cases} \tag{5.58}$$

which is the Fréchet differential of the convex functional $\frac{1}{q} \|w\|^q_{L^q(\gamma, Y \times U)}$. A straightforward calculation shows that this map satisfies

$$\|j_q(w)\|^p_{L^p(\gamma, Y \times U)} = \|w\|^q_{L^q(\gamma, Y \times U)} = \int_{Y \times U} \langle j_q(w), w \rangle \, \mathrm{d}\gamma(y, u). \tag{5.59}$$

See also Ambrosio et al. [2005, Chapter 8].

Theorem 31 (Absolutely Continuous Curves in $\mathbb{P}^\mu_p(Y \times U)$)**.**
Let $I \subset \mathbb{R}$ be an open interval, and suppose $\gamma_t : I \to \mathbb{P}^\mu_p(Y \times U)$ is an absolutely continuous in the W^μ_p metric with $|\gamma'|(t) \in L^1(I)$. Then, there exists a Borel vector field $v_t(y, u)$ such that

1. *v_t is triangular*
2. *$v_t \in L^p(\gamma_t, Y \times U)$ and $\|v_t\|_{L^p(\gamma_t, Y \times U)} \leq |\gamma'|(t)$ for a.e. t*
3. *(v_t, γ_t) solve the continuity equation distributionally.*

Proof. Assume without loss of generality that $|\gamma'|(t) \in L^\infty(I)$ and that $I = (0, 1)$ [Ambrosio et al., 2005, Lemma 1.1.4, Lemma 8.1.3]. Fix any $\varphi \in \mathrm{Cyl}(Y \times U)$. For $s, t \in I$ there exists an optimal triangular coupling $\gamma_{st} \in \Pi_Y(\gamma_s, \gamma_t)$. By Hölder's inequality,

$$|\gamma_t(\varphi) - \gamma_s(\varphi)| \leq \mathrm{Lip}(\varphi) W^\mu_p(\gamma_s, \gamma_t). \tag{5.60}$$

It follows that $t \mapsto \gamma_t(\varphi)$ is absolutely continuous. We can introduce the upper semicontinuous

and bounded map

$$H(y_0,u_0,y_1,u_1) = \begin{cases} |\nabla\varphi(y_0,u_0)| & (y_0,u_0) = (y_1,u_1) \\ \frac{|\varphi(y_0,u_0)-\varphi(y_1,u_1)|}{|(y_0,u_0)-(y_1,u_1)|} & (y_0,u_0) \neq (y_1,u_1) \end{cases}. \tag{5.61}$$

For $|h|$ sufficiently small, choose any optimal coupling $\gamma_{(s+h)h} \in \Pi_Y(\gamma_{s+h},\gamma_s)$ and note that

$$\frac{|\gamma_{s+h}(\varphi)-\gamma_s(\varphi)|}{|h|} \leq \frac{1}{|h|}\int_{(Y\times U)^2} |(y_0,u_0)-(y_1,u_1)| H(y_0,u_0,y_1,u_1)\,\mathrm{d}\gamma_{(s+h)s} \tag{5.62}$$

$$\leq \frac{W_p^\mu(\gamma_{s+h},\gamma_s)}{|h|}\left(\int_{(Y\times U)^2} H^q(y_0,u_0,y_1,u_1)\,\mathrm{d}\gamma_{(s+h)h,s}\right)^{1/q}. \tag{5.63}$$

If t is a point of metric differentiability for $t\mapsto\gamma_t$, note that $\gamma_{(t+h)t} \to (I,I)_\#\gamma_t$ narrowly, where I is the identity map on $Y\times U$. Moreover, since $\gamma_t \in \mathbb{P}_p^\mu(Y\times U)$, it follows that on the diagonal we have that almost surely $H(y_0,u_0,y_0,u_1) = \iota(|\nabla_u\varphi(y_0,u_0)|$. Thus,

$$\limsup_{h\to 0}\frac{|\gamma_{t+h}(\varphi)-\gamma_t(\varphi)|}{|h|} \leq |\gamma'|(t)\left(\int_{Y\times U}|H|^q(y_0,u_0,y_0,u_0)\,\mathrm{d}\gamma_t(y_0,u_0)\right)^{1/q} \tag{5.64}$$

$$= |\gamma'|(t)\,\|\iota(\nabla_u\varphi)\|_{L^q(\gamma_t,Y\times U)} = |\gamma'|(t)\,\|\nabla_u\varphi\|_{L^q(\gamma_t,U)}. \tag{5.65}$$

Taking $Q = Y\times U\times I$ and $\gamma = \int\gamma_t\,\mathrm{d}t$, fix any $\varphi\in\mathrm{Cyl}(Q)$. We have that

$$\begin{aligned}&\int_Q \partial_s\varphi(y,u,s)\,\mathrm{d}\gamma(y,u,s)\\ &= \lim_{h\downarrow 0}\int_I \frac{1}{h}\left(\int_{Y\times U}\varphi(y,u,s)\,\mathrm{d}\gamma_s(y,u) - \int_{(Y\times U)}\varphi(y,u,s)\,\mathrm{d}\gamma_{s+h}(y,u)\right)\mathrm{d}s.\end{aligned} \tag{5.66}$$

An application of Fatou's Lemma, Equation (5.64), and Hölder's inequality gives us

$$\left|\int_Q \partial_s\varphi(y,u,s)\,\mathrm{d}\gamma(y,u,s)\right| \leq \left(\int_J |\gamma'|(s)\,\mathrm{d}s\right)^{1/p}\left(\int_Q |\nabla_u\varphi(y,u,s)|^q\,\mathrm{d}\mu(y,u,s)\right)^{1/q} \tag{5.67}$$

for any interval $J \subset I$ with $\operatorname{supp} \varphi \subset J \times Y \times U$.

Fix the subspace

$$V = \{\iota(\nabla_u \varphi(y,u,s)) : \varphi \in \operatorname{Cyl}(Q)\} \subseteq Y \times U \tag{5.68}$$

and denote by $\overline{V}$ its $L^q(\gamma, Y \times U \times I)$ closure. Define the linear functional $L : V \to \mathbb{R}$ via

$$L(\nabla_u \varphi) = -\int_Q \partial_s \varphi(y,u,s)\, \mathrm{d}\gamma(y,u,s) \tag{5.69}$$

and note that Equation (5.67) implies that L is a bounded linear functional on V. Thus (by Hahn-Banach and the fact that $V \subseteq \overline{V}$ is dense) we may uniquely extend L to $\overline{V}$. We thus have a convex minimization problem

$$\min_{w \in \overline{V}} \frac{1}{q} \int_Q |w(y,u,s)|^q \, \mathrm{d}\gamma(y,u,s) - L(w) \tag{5.70}$$

which admits the unique solution w such that $j_q(w) - L = 0$. In particular, the estimate (5.67) shows that the above functional is coercive and hence admits a minimizer which we may obtain via its differential as a consequence of convexity. Thus, we obtain a triangular vector field $v = j_q(w)$ such that for all $\varphi \in \operatorname{Cyl}(Q)$,

$$\langle v, \nabla \varphi \rangle = \int_Q \langle v(y,u,s), \nabla \varphi(y,u,s) \rangle \, \mathrm{d}\gamma(y,u,s) = \langle L, \nabla \varphi \rangle = -\int_Q \partial_s \varphi(y,u,s)\, \mathrm{d}\gamma(y,u,s). \tag{5.71}$$

This precisely shows that (v_t, γ_t) is a triangular distributional solution to the continuity equation.

Now, choose any interval $J \subset I$ and choose a sequence $\eta^k \in C_c^\infty(J)$, with $0 \leq \eta^k \leq 1$ and

$\eta_k \to \mathbb{1}_J$ as $k \to \infty$. Moreover choose a sequence $(\nabla_u \varphi_n) \subset V$ converging to $w = j_p(v)$ in $L^q(\gamma, Q)$. Our previous calculations give

$$\int_Q \eta^k(s)|v(y,u,s)|^p \,\mathrm{d}\gamma(y,u,s) = \int_Q \eta^k(s)\langle v, w\rangle \,\mathrm{d}\gamma = \lim_{n\to\infty} \int_Q \eta^k \langle v, \nabla_u \varphi_n\rangle \,\mathrm{d}\gamma \tag{5.72}$$

$$= \lim_{n\to\infty} \langle L, \nabla_u(\eta^k \varphi_n)\rangle \leq \left(\int_J |\gamma'|^p(s)\,\mathrm{d}s\right)^{1/p} \left(\int_{J\times Y\times U} |v|^p \,\mathrm{d}\gamma\right)^{1/p}. \tag{5.73}$$

Taking $k \to \infty$ we see that

$$\int_J \int_{Y\times U} |v_t(y,u)|^p \,\mathrm{d}\gamma_t(y,u)\,\mathrm{d}t \leq \int_J |\gamma'|^p(s)\,\mathrm{d}s \tag{5.74}$$

and since $J \subset I$ was arbitrary, we conclude

$$\|v_t\|_{L^p(\gamma_t, Y\times U)} \leq |\gamma'|(t) \qquad \text{a.e.-}t. \tag{5.75}$$

□

Conversely, we show in Theorem 32 that if the pair (γ_t, v_t) solve the continuity equation and v_t is triangular, then the curve (γ_t) is absolutely continuous and $|\gamma'(t)| \leq \|v_t\|_{L^p(\gamma_t, Y\times U)}$. The main technique of this result is to study the collection of *conditional* continuity equations (which is feasible by Lemma 3) and to apply the converse of Ambrosio et al. [2005, Theorem 8.3.1]. In this setting, the infinite-dimensional result is obtained via a finite-dimensional approximation argument.

Theorem 32 (Continuous Curves Generated by Triangular Vector Fields)**.**
Suppose that $\gamma_t : I \to \mathbb{P}_p^\mu(Y \times U)$ is narrowly continuous and (v_t) is a triangular vector field such that (γ_t, v_t) solve the continuity equation with $\|v_t\|_{L^p(\gamma_t, Y\times U)} \in L^1(I)$. Then, $\gamma_t : I \to \mathbb{P}_p^\mu(Y \times U)$ is absolutely continuous in the W_p^μ metric and $|\gamma'|(t) \leq \|v_t\|_{L^p(\mu, Y\times U)}$ for almost every t.

Proof. We first assume that U is finite dimensional. Our strategy is to check the hypotheses necessary for Ambrosio et al. [2005, Theorem 8.3.1] to hold for μ-almost every y, followed by an application of this theorem. By Lemma 3, for μ-almost every y we have that $(\gamma_t^y, v_t^U(y,-))$ solve the continuity equation distributionally on $I \times U$.

By Jensen's inequality (and the assumption $p \geq 1$) we see

$$\int_I \|v_t\|_{L^p(\gamma_t, Y\times U)} \,\mathrm{d}t = \int_I \mathbb{E}_{y\sim\mu}\left[\left\|v_t^U(y,-)\right\|^p_{L^p(\gamma_t^y,U)}\right]^{1/p} \,\mathrm{d}t \tag{5.76}$$

$$\geq \int_I \mathbb{E}_{y\sim\mu}\left[\left\|v_t^U(y,-)\right\|_{L^p(\gamma_t^y,U)}\right] \,\mathrm{d}t \tag{5.77}$$

$$= \mathbb{E}_{y\sim\mu}\left[\int_I \left\|v_t^U(y,-)\right\|_{L^p(\gamma_t^y,U)} \,\mathrm{d}t\right]. \tag{5.78}$$

Since the first term is finite, it follows that

$$\left\|v_t^U(y,-)\right\|_{L^p(\gamma_t^y,U)} \in L^1(I) \qquad \mu\text{-almost every } y. \tag{5.79}$$

Now Ambrosio et al. [2005, Lemma 8.1.2] shows that for μ-almost every y we have that (γ_t^y) admits a narrowly continuous representative $(\tilde{\gamma}_t^y)$ with $\tilde{\gamma}_t^y = \gamma_t^y$ for almost every t. It follows from Ambrosio et al. [2005, Theorem 8.3.1] that for any $t_1 \leq t_2$ in I, we have

$$W_p^p(\tilde{\gamma}_{t_1}^y, \tilde{\gamma}_{t_2}^y) \leq (t_2 - t_1)^{p-1} \int_{t_1}^{t_2} |v_t^U(y,u)|^p \,\mathrm{d}\tilde{\gamma}_t^y(u)\,\mathrm{d}t \tag{5.80}$$

$$= (t_2 - t_1)^{p-1} \int_{t_1}^{t_2} |v_t^U(y,u)|^p \,\mathrm{d}\gamma_t^y(u)\,\mathrm{d}t \tag{5.81}$$

where the second line follows as $\tilde{\gamma}_t^y = \gamma_t^y$ for almost every t.

Let $\tilde{\gamma}_t = \int_Y \tilde{\gamma}_t^y \,\mathrm{d}\mu(y)$ be the measure obtained via marginalizing over the Y-variables. Taking

an expectation over $y \sim \mu$, the previous inequality shows us that

$$\frac{W_p^{\mu,p}(\tilde{\gamma}_{t_1}, \tilde{\gamma}_{t_2})}{(t_2 - t_1)^p} \leq \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \|v_t\|^p_{L^p(\gamma_t, Y \times U)} \, \mathrm{d}t. \tag{5.82}$$

Now, note that t_1 is almost surely a Lebesgue point of the right-hand side and $\tilde{\gamma}_{t_1} = \gamma_{t_1}$. Taking $t_2 \to t_1$ along a sequence where $\tilde{\gamma}_{t_2} = \gamma_{t_2}$ shows us that

$$|\gamma'|(t) \leq \|v_t\|_{L^p(\gamma_t), Y \times U} \tag{5.83}$$

for almost every $t \in I$.

In the case that U is infinite dimensional, fix any $y \in Y$ such that Lemma 3 holds (which is of full measure) and fix a countable orthonormal basis (e_k) for U. Set $\pi^d : U \to \mathbb{R}^d$ to be the projection operator for this basis, i.e. $u \mapsto (\langle u, e_1 \rangle, \ldots, \langle u, e_d \rangle)$. We consider the collection of finite dimensional conditional measures $\gamma_t^{d,y} = \pi^d_\# \gamma_t^y$. By the same argument in Ambrosio et al. [2005, Theorem 8.3.1], there exists a vector field $v_t^{d,y}$ on $\mathbb{R}^d$ such that $(\gamma_t^{d,y}, v_t^{d,y})$ solve the continuity equation and

$$\left\| v_t^{d,y} \right\|_{L^p(\gamma_t^{d,y}, \mathbb{R}^d)} \leq \left\| v_t^U(y, -) \right\|_{L^p(\gamma_t^y, U)}. \tag{5.84}$$

It follows from the finite-dimensional case above that for almost every $t_1 \leq t_2$, we have

$$W_p^p(\gamma_{t_1}^{d,y}, \gamma_{t_2}^{d,y}) \leq (t_2 - t_1)^{p-1} \int_{t_1}^{t_2} \left\| v_t^U(y, -) \right\|^p_{L^p(\gamma_t^y, U)} \, \mathrm{d}t. \tag{5.85}$$

Let $\hat{\gamma}_t^{y,d} = (\pi^d)^\star_\# \gamma_t^{y,d}$ where $(\pi^d)^\star : \mathbb{R}^d \to U$ maps $z \mapsto \sum_{k=1}^d z_k e_k$. As $d \to \infty$ we have $\hat{\gamma}_t^{d,y} \to \gamma_t^y$ narrowly for all $t \in I$. Since $(\pi^d)^\star$ is an isometry, Ambrosio et al. [2005, Lemma 7.1.4]

shows that

$$W_p^p(\gamma_{t_1}^y, \gamma_{t_2}^y) \leq \liminf_{d\to\infty} W_p^p(\gamma_{t_1}^{d,y}, \gamma_{t_2}^{d,y}) \leq (t_2 - t_1)^{p-1} \int_{t_1}^{t_2} \left\| v_t^U(y, -) \right\|_{L^p(\gamma_t^y, U)}^p \, \mathrm{d}t. \tag{5.86}$$

Now, integration with respect to $\mathrm{d}\mu(y)$ yields

$$W_p^{p,\mu}(\gamma_{t_1}, \gamma_{t_2}) \leq (t_2 - t_1)^{p-1} \int_{t_1}^{t_2} \|v_t\|_{L^p(\gamma_t, Y\times U)}^p \, \mathrm{d}t. \tag{5.87}$$

Taking $t_2 \to t_1$ shows that for almost every t we have

$$|\gamma'|(t) \leq \|v_t\|_{L^p(\gamma_t, Y\times U)}. \tag{5.88}$$

□

As a corollary of Theorem 31 and Theorem 32, we obtain a conditional version of the Benamou-Brenier theorem [Benamou and Brenier, 2000]. The proof of Theorem 33 largely follows the unconditional case (see e.g. Ambrosio et al. [2005, Chapter 8]), but we include it for the sake of completeness.

Theorem 33 (Conditional Benamou-Brenier)**.**
Let $1 < p < \infty$. For any $\eta, \nu \in \mathbb{P}_p^\mu(Y \times U)$, we have

$$W_p^{p,\mu}(\eta, \nu) = \min_{(\gamma_t, v_t)} \left\{ \int_0^1 \|v_t\|_{L^p(\mu_t)}^p \mid (v_t, \gamma_t) \text{ solve } (5.53),\ \gamma_0 = \eta, \gamma_1 = \nu, \text{ and } v_t \text{ is triangular} \right\}.$$

Proof. Write M for the infimum on the right-hand side.

First, suppose that (v_t, μ_t) are admissible and $\int_0^1 \|v_t\|_{L^p(\mu_t)} < \infty$. It follows from Theorem 32

that (γ_t) is an absolutely continuous curve in $\mathbb{P}_p^\mu(Y \times U)$ and $\|v_t\|_{L^p(\mu_t, Y\times U)} \geq |\gamma'|(t)$. Thus,

$$W_p^{\mu,p}(\eta,\nu) \leq \left(\int_0^1 |\gamma'|(t)\,\mathrm{d}t\right)^p \leq \int_0^1 \|v_t\|^p_{L^p(\mu_t, Y\times U)}\,\mathrm{d}t \leq M. \tag{5.89}$$

Conversely, by Theorem 29 there exists a constant speed geodesic $(\gamma_t) \subset \mathbb{P}_p^\mu(Y\times U)$ connecting η and ν. Recall that constant speed geodesics are absolutely continuous. By Theorem 31, there exists a Borel triangular vector field v_t such that (v_t, γ_t) solve the continuity equation, and moreover $\|v_t\|_{L^p(\mu_t, Y\times U)} \leq |\gamma'|(t)$. In fact, because (v_t, γ_t) solve the continuity equation, Theorem 32 yields that $\|v_t\|_{L^p(\mu_t, Y\times U)} = |\gamma'|(t)$.

Since γ_t is a constant speed geodesic in $\mathbb{P}_p^\mu(Y \times U)$, it follows that $|\mu'|(t) = W_p^\mu(\eta,\nu)$ for almost every $t \in (0,1)$. Hence,

$$W_p^{\mu,p}(\eta,\nu) = \int_0^1 |\gamma'|(t)^p\,\mathrm{d}t = \int_0^1 \|v_t\|^p_{L^p(\gamma_t, Y\times U)} \geq M. \tag{5.90}$$

Thus, $W_p^{\mu,p}(\eta,\nu) = M$ as desired. □

5.5 COT Flow Matching

We have thus far seen that the COT problem (5.3) admits a dynamical formulation by Theorem 33, where one may take the underlying vector fields to be triangular. We use these results to design a principled model for conditional generation based on flow matching [Lipman et al., 2023, Albergo et al., 2024, Tong et al., 2024]. We hereafter use the squared-distance cost (i.e. $p = 2$).

Flow Matching. We assume that we have access to samples $z_0 = (y_0, u_0) \sim \eta(y_0, u_0) \in \mathbb{P}_p^\mu(Y \times U)$ from a source measure, and samples $z_1 = (y_1, u_1) \sim \nu(y_1, u_1) \in \mathbb{P}_p^\mu(Y \times U)$ from a target measure. Let $z = (z_0, z_1) \sim \rho(z_0, z_1) \in \Pi(\eta, \nu)$ be any coupling of the source and target measure. We specify a collection of measures and vector fields on $Y \times U$ via

$$\gamma_t(y, u \mid z) = \mathcal{N}\left(y, u \mid t z_1 + (1-t)\, z_0, C\right) \qquad v_t(y, u \mid z) = z_1 - z_0 \tag{5.91}$$

where C is any trace-class covariance operator [Da Prato and Zabczyk, 2014]. As is standard in flow matching [Lipman et al., 2023, Kerrigan et al., 2024a], we obtain from Equations (5.91) a marginal measure $\gamma_t(y, u)$ and vector field $v_t(y, u)$ satisfying the continuity equation via

$$\gamma_t(y, u) = \int_{(Y \times U)^2} \gamma_t(y, u \mid z)\, \mathrm{d}\rho(z) \qquad v_t(y, u) = \int_{(Y \times U)^2} v_t(y, u \mid z) \frac{\mathrm{d}\gamma_t(y, u \mid z)}{\mathrm{d}\gamma_t(y, u)}\, \mathrm{d}\rho(z). \tag{5.92}$$

This marginal path $(\gamma_t)_{t=0}^1$ interpolates between the source measure ($t = 0$) and a smoothed version of the target measure ($t = 1$). To transform source samples from η into target samples from ν, we seek to learn the intractable vector field $v_t(y, u)$ with a model $v^\theta(t, y, u)$ by minimizing the loss[2]

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \rho(z), \gamma_t(y,u|z)} \left\| v^\theta(t, y, u) - v_t(y, u \mid z) \right\|^2 \tag{5.93}$$

which has the same θ-gradient as the MSE loss to the true vector field $u_t(y, u)$ [Tong et al., 2024].

[2]Previous work has referred to this as the *conditional flow matching loss* [Tong et al., 2024], which is not to be confused with the notion of conditioning that we focus on in this work.

COT Flow Matching. In the preceding section, $\rho(z)$ may be an arbitrary coupling between η and ν. Motivated by Proposition 22, we will choose ρ to be a COT coupling. Under sufficient regularity conditions (see Section 5.2.1), this COT plan will be induced by a triangular map. In turn, Theorem 30 gives us that this triangular map is generated by a triangular vector field of the form (5.91). Thus, we parametrize our model u^θ to also be triangular. Moreover, we recover the optimal dynamic transport given in Theorem 33 as $\mathrm{Tr}(C) \to 0$ by a pointwise application of [Tong et al., 2024, Proposition 3.4].

Given a collection of samples $\{z_0^i, z_1^i\}_{i=1}^n$ drawn from η and ν, we approximate a conditional optimal coupling ρ using standard numerical techniques with the cost function $c_\epsilon(y_0, u_0, y_1, u_1) = |y_1 - y_0|^2 + \epsilon|u_1 - u_0|^2$ for some $0 < \epsilon \ll 1$. Intuitively, such a cost penalizes mass transfer along the Y dimension, which is precisely the constraint sought in the COT problem (5.3). Hosseini et al. [2023, Prop. 3.11] show that as $\epsilon \downarrow 0$, we recover the true optimal triangular map. The COT coupling can either be precomputed for small datasets or computed on each minibatch drawn during training. After training, we obtain a learned triangular vector field $v^\theta(t, y, u)$. Given an arbitrary fixed $y \in Y$, we may approximately sample from the target $\nu(u \mid y)$ by sampling $u_0 \sim \eta(u_0 \mid y)$ and numerically solving the corresponding flow equation $\partial_t(y, u_t) = v^\theta(t, y, u_t)$ with initial condition (y, u_0).

Source Measure. Our framework is agnostic to the choice of source measure η, allowing for flexibility in the modeling process. The main requirement is that the Y-marginals of the source η and target η must match. In some scenarios, this is trivially satisfied. If one is interested in using a source distribution which is simply random noise, one may take $\eta(y_0, u_0) = \pi^Y_\# \nu(y_0) \otimes \eta_U(u_0)$ to be the product of two independent distributions where η_U is arbitrary, e.g. Gaussian noise.

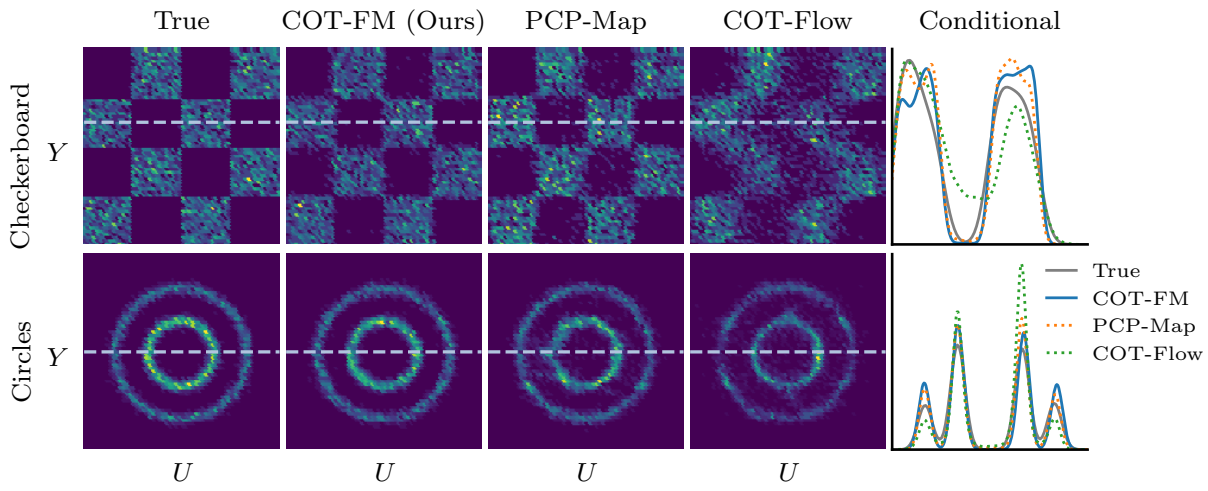


Figure 5.2: Samples from the ground-truth joint target distribution and the various models. Samples from COT-FM more closely match the ground-truth distribution than the baselines. In the final column, we plot conditional KDEs for samples drawn conditioned on the y value indicated by the dashed horizontal line. See Appendix C.1 for a larger figure and additional results.

5.6 Experiments

We now illustrate our methodology (COT-FM) on a variety of conditional simulation tasks. We compare our method against several competitive baselines, namely PCP-Map [Wang et al., 2023], COT-Flow [Wang et al., 2023], and WaMGAN [Hosseini et al., 2023]. These baselines are chosen as they reflect current state-of-the-art approaches to learning COT maps. We additionally compare against flow matching [Lipman et al., 2023, Wildberger et al., 2024] without COT, i.e. where the coupling between the source and target measures is the independent coupling $\rho(z_0, z_1) = \eta \otimes \nu$. Overall, our method (COT-FM) typically outperforms these baselines across the diverse and challenging set of tasks we consider. We find that PCP-Map [Wang et al., 2023] is a strong baseline, but we emphasize that this model relies on the use of an input-convex neural network [Amos et al., 2017] and it is hence unclear how to adapt this method to e.g. images. Appendix C.1 contains further details and results for all of our experiments.

Table 5.1: Distances between the ground-truth and generated joint distributions for the 2D datasets. Our method (COT-FM) obtains lower distances than the considered baselines. Average results ± one standard deviation are reported across five test sets, with the lowest average distance in bold.

	Checkerboard		Moons		Circles		Swissroll	
	W_2 (1e-2)	MMD (1e-3)	W_2 (1e-2)	MMD (1e-3)	W_2 (1e-2)	MMD (1e-3)	W_2 (1e-2)	MMD (1e-3)
PCP-Map	6.27±0.81	0.21±0.13	8.44±1.09	0.22±0.10	6.19±0.43	**0.20**±0.17	5.35±0.93	0.16±0.13
COT-Flow	8.20±0.49	0.26±0.16	18.49±2.22	1.32±0.79	10.04±1.69	0.24±0.22	6.47±0.69	0.19±0.19
FM	8.81±0.58	0.24±0.20	15.55±0.77	1.85±0.22	7.03±0.17	0.45±0.11	8.18±0.34	0.58±0.09
COT-FM (Ours)	**4.69**±1.00	**0.17**±0.13	**6.50**±1.41	**0.13**±0.10	**5.56**±0.43	**0.20**±0.04	**4.64**±1.26	**0.15**±0.19

2D Synthetic Data. We first consider synthetic distributions where $Y = U = \mathbb{R}$. Our source measure is taken to be the independent product $\eta(y, u) = \pi^Y_\# \nu \otimes \mathcal{N}(0, 1)$. We plot ground-truth joint distributions and samples for two datasets in Figure 5.2. See Appendix C.1 for additional results. Samples from our method (COT-FM) closely match those from the ground-truth distribution, whereas samples from PCP-Map and COT-Flow [Wang et al., 2023] can produce samples in regions of zero support under the ground-truth distribution. In Table 5.1, we provide a quantitative analysis, where we measure the W_2 and MMD distances between the generated and ground-truth joint distributions. This is motivated by Proposition 22, as triangular maps which couple the joint distributions necessarily couple the conditional distributions. Our method outperforms the baselines across all metrics.

Lotka-Volterra (LV) Dynamical System. Here we estimate parameters of the LV model given only noisy observations of its solution. The LV model has parameters $u = (\alpha, \beta, \gamma, \delta) \in \mathbb{R}^4_{\geq 0}$ and a pair of coupled nonlinear ODEs of the form

$$\frac{\mathrm{d}p_1(t)}{\mathrm{d}t} = \alpha p_1 - \beta p_1 p_2 \qquad \frac{\mathrm{d}p_2(t)}{\mathrm{d}t} = -\gamma p_2 + \delta p_1 p_2 \tag{5.94}$$

whose solution $p(t) = (p_1(t), p_2(t)) \in \mathbb{R}^2_{\geq 0}$ represents the number of prey and predator species at time $t \in [0, T]$. Following Alfonso et al. [2023], we assume $p(0) = (30, 1)$ and that $\log(u) \sim \mathcal{N}(m, 0.5I)$ with $m = (-0.125, -3, -0.125, -3)$. Given parameters $u \in \mathbb{R}^4_{\geq 0}$, we

Table 5.2: Statistical distances between MCMC and posterior samples $u \sim \nu(u \mid y)$ for each method on the LV dataset. Average results $\pm$ one standard deviation reported across five test sets.

	W_2 (1e-2)	MMD (1e-3)
PCP-Map	5.04$\pm$0.05	2.67$\pm$2.1
COT-Flow	4.86$\pm$1.1	**0.83**$\pm$0.50
FM	11.41$\pm$0.26	2.65$\pm$0.14
COT-FM (Ours)	**4.02**$\pm$0.06	0.95$\pm$0.03

simulate Equation (5.94) for $t \in \{0, 2, \ldots, 20\}$ to obtain a solution $z(u) \in \mathbb{R}^{22}_{\geq 0}$. An observation $y \in \mathbb{R}^{22}_{\geq 0}$ is obtained by the addition of log-normal noise, i.e. $\log(y) \sim \mathcal{N}(\log(z(u), 0.1I)$. We thus may simulate many (y, u) pairs from the target measure for training.

As a benchmark, we follow the settings of Alfonso et al. [2023] and choose parameters $u = (0.83, 0.041, 1.08, 0.04)$ to generate a single observation y as described above. In Figure 5.3, we plot a histogram of $10,000$ samples from the posterior $\nu(u \mid y)$ of COT-FM. Since the ground-truth posterior is intractable, we compare against differential evolution Metropolis MCMC [Braak, 2006]. Samples from our method qualitatively resemble those from MCMC, and the posterior mode is typically close to the true unknown u (shown in red). Our method is quantitatively closest to the MCMC samples in the W_2 metric, and competitive in the MMD metric (Table 5.2). Importantly, we note that the flow matching ablation (FM), which does not include the use of COT couplings, performs markedly worse than our proposed COT-FM method. Appendix C.1 contains additional results.

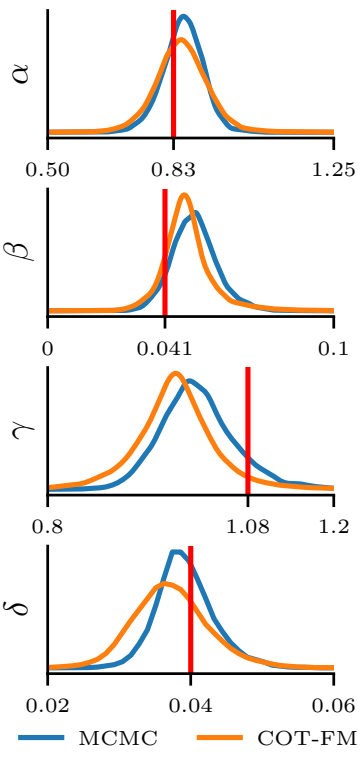


Figure 5.3: Sample KDEs on the Lotka-Volterra inverse problem. The red lines denote the true parameter values.

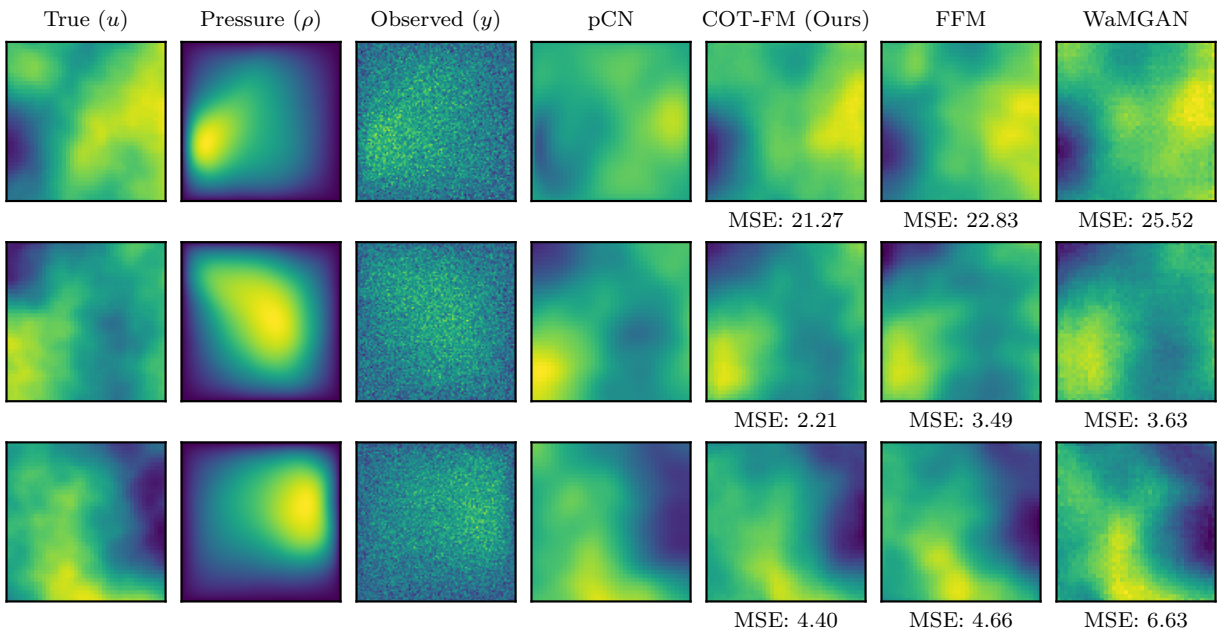


Figure 5.4: Darcy flow illustration. A true permeability u is shown, as well as the pressure field ρ and its observed, noisy version y. We compare an ensemble average of posterior samples from the various methods against MCMC (pCN) [Cotter et al., 2013]. COT-FM achieves the lowest MSE to pCN.

Darcy Flow Inverse Problem. Here we consider an infinite-dimensional Bayesian inverse problem from the 2D Darcy flow PDE. The setting is adapted from Hosseini et al. [2023]. We opt to compare against WaMGAN [Hosseini et al., 2023], as this is currently the only other extant amortized function-space COT method, and FFM [Kerrigan et al., 2023] as a function-space flow matching ablation.

Table 5.3: Predictive performance of the generated samples on the Darcy flow inverse problem. Average result ± one standard deviation obtained on 5 test sets of 5,000 samples each.

	MSE (1e-2)	CRPS (1e-2)
WaMGAN	6.55$\pm$0.07	18.75$\pm$0.10
FFM	7.30$\pm$0.07	**15.47**$\pm$0.06
COT-FFM (Ours)	**5.40**$\pm$0.08	15.56$\pm$0.08

The Darcy flow PDE is an elliptic equation on a smooth domain $\Omega \subseteq \mathbb{R}^d$ which relates a permeability field $\exp(u)$, a pressure field ρ, and a source term f via $-\operatorname{div} \exp(u)\nabla\rho = f$ on Ω subject to $\rho = 0$ on $\partial\Omega$. Our goal is to recover the permeability u from noisy measurements y of the pressure ρ. Both the unknown u and observations y are functions and thus infinite-dimensional. To

define our target measure, we specify a prior $\nu(u) = \mathcal{N}(0, C)$ with a Matérn kernel C of lengthscale $\ell = 1/2$ and $\nu = 3/2$. Given $u \sim \eta(u)$, the Darcy flow PDE is solved numerically [Alnæs et al., 2015] to obtain a solution $\rho(u)$ observed at some finite but arbitrary number of points $\{x_1, \ldots, x_n\} \subset \mathbb{R}^2$. An observation $y(u)$ is obtained by adding Gaussian noise to each observation, i.e. $y(u) \sim \mathcal{N}(\rho(u), \sigma^2 I)$ where $\sigma = 2.5 \times 10^{-2}$. We implement all models via a Fourier Neural Operator [Li et al., 2021], allowing us to work with arbitrary discretizations, as required by the functional nature of this problem.

We provide an illustration in Figure 5.4. As the true posterior is intractable, we compare against preconditioned Crank-Nicolson (pCN) [Cotter et al., 2013], a function-space MCMC method. In Figure 5.4, we plot the mean posteriors obtained from the various methods. Qualitatively, both COT-FM and FFM are good approximations to pCN, while WaMGAN has visual artifacts. However, the MSE between our method and the pCN mean is lower than that of FFM. Table 5.3 provides a quantitative comparison between the methods on a test set of 5,000 samples, where we measure MSE and CRPS [Hersbach, 2000]. We compare the ensemble mean of 10 samples against the true u value as running pCN for each observation is prohibitively expensive. COT-FM outperforms FFM and WaMGAN in terms of MSE and is on-par with FFM in terms of CRPS. See Appendix C.1 for further results.

5.7 Conclusion

We analyze conditional optimal transport from a dynamic point of view. Our analysis culminates in the characterization of absolutely continuous curves of measures in a conditional Wasserstein space, resulting in a conditional analog of the Benamou-Brenier Theorem. We use these result to build a framework of triangular flow matching to develop simulation-free methods for conditional generative models. Our methods are applicable across a wide class of problems, and we demonstrate our methodology on several challenging inverse problems.

Chapter 6

Forecasting Continuous-Time Event Data with Flow Matching

This chapter is somewhat of a departure from the previous chapters, where we made the fundamental assumption that our data measure is supported on a Hilbert space. While Hilbert spaces are quite general, there are several data modalities which are not captured in this setting. In this chapter, we focus on continuous-time event sequences, in which events occur at irregular intervals. Such data can be viewed as a draw from a probability measure supported on a *configuration space*, whose elements are counting measures. While configuration spaces are generally non-Hilbertian, these spaces are infinite-dimensional and carry a rich structure which we leverage to develop transport-based generative models.

Many stochastic processes, ranging from consumer behavior [Hernandez et al., 2017] to the occurrence of earthquakes [Ogata, 1998], are best understood as a sequence of discrete events which occur at random times. Any observed event sequence, consisting of one or more event times, may be viewed as a draw from a temporal point process (TPP) [Daley and Vere-Jones, 2003] which characterizes the distribution over such sequences. Given a collection of observed

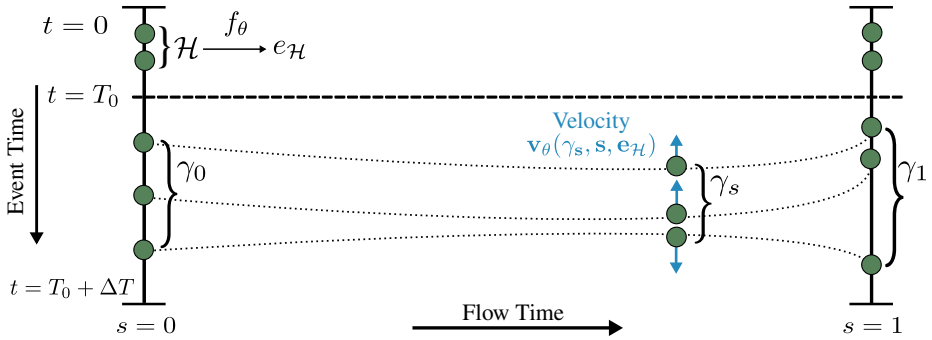


Figure 6.1: All illustration of forecasting with our `EventFlow` method. The horizontal axis indicates the flow time s, and the vertical axis indicates the support of the TPP $\mathcal{T} = [0, T]$. We first encode the observed history $\mathcal{H}$ into an embedding $e_\mathcal{H} = f_\theta(\mathcal{H})$. At $s = 0$, we independently draw n events in the forecasting window $[T_0, T_0 + \Delta T]$ from a fixed reference distribution, constituting a sample γ_0 from a mixed-binomial TPP. Each event can be thought of as a particle, which is assigned a velocity by a neural network $v_\theta(\gamma_s, s, e_\mathcal{H})$. Each particle flows along its corresponding velocity field until reaching its terminal point at $s = 1$, whereby we obtain a forecasted sequence γ_1.

event sequences, faithfully modeling the underlying TPP is critical in both understanding and forecasting the phenomenon of interest.

While multiple different parametric TPP models have been proposed [Hawkes, 1971, Isham and Westcott, 1979], their limited flexibility limits their application when modeling complex real-world sequences. This has motivated the use of neural networks [Du et al., 2016, Mei and Eisner, 2017] in modeling TPPs. To date, most neural network based TPP models are autoregressive in nature [Shchur et al., 2020a, Zhang et al., 2020], where a model is trained to predict the next event time given an observed history of events. However, in many tasks, we are interested not only in the next event, but in the entire sequence of events which is to follow. While these models can achieve high likelihoods, their performance in many-step forecasting tasks can be unsatisfactory due to compounding errors arising from the autoregressive sampling procedure [Xue et al., 2022, Lüdke et al., 2023].

Moreover, existing models are typically trained via a maximum likelihood procedure (see Section 6.2) which involves computing the CDF implied by the learned model. When using a neural model, computing this CDF necessitates techniques such as Monte Carlo estimation to

properly compute the loss [Mei and Eisner, 2017]. In addition, sampling from intensity-based models [Du et al., 2016, Mei and Eisner, 2017, Yang et al., 2022] is nontrivial, requiring an expensive and difficult to implement approach based on the thinning algorithm [Lewis and Shedler, 1979, Ogata, 1981, Xue et al., 2024].

Motivated by these limitations, we propose `EventFlow`, a generative model which directly learns the joint event time distributions, thus allowing us to avoid autoregressive sampling altogether. Our proposed model extends the flow matching framework [Lipman et al., 2023, Albergo and Vanden-Eijnden, 2023, Liu et al., 2023] to the setting of TPPs, where we learn a continuous flow from a reference TPP to our data TPP. At an intuitive level, samples from our model are generated by drawing a collection of event times from a reference distribution and flowing these events along a learned vector field. The number of events is fixed throughout this process, decoupling the event counts and their times, so that the distribution over event counts can be learned or otherwise specified. See Figure 6.1 for an illustration. As our primary contribution regards the modeling of the event times themselves, we focus on unmarked point processes in this chapter.

More specifically, in this chapter we develop `EventFlow`, a novel generative model for temporal point processes. Our model is suitable for both unconditional generation tasks (i.e., generating draws from the underlying data TPP) and conditional generation tasks (e.g., forecasting future events given a history), and is able to forecast multiple events simultaneously. Our model provides a new perspective on modeling TPPs and sidesteps common pitfalls in existing approaches. In particular, `EventFlow` is likelihood-free, non-autoregressive, easy to sample from, and straightforward to implement. On standard benchmark datasets, `EventFlow` obtains uniformly strong performance on a multi-step forecasting task, and matches or exceeds the performance of state-of-the-art models for unconditional generation.

6.1 Related Work

Temporal Point Processes The statistical modeling of temporal point processes (TPPs) is a classical subject with a long history [Daley and Vere-Jones, 2003, Hawkes, 1971, Isham and Westcott, 1979]. The contemporary modeling paradigm, based on neural networks [Du et al., 2016], typically operates by learning a *history encoder* and an *event decoder*. The history encoder seeks to learn a fixed-dimensional vector representation of the history of a sequence up to some given time, and the decoder seeks to model a distribution over the subsequent event(s).

Numerous models have been proposed for both components. Popular choices for the history encoder include RNN-based models [Du et al., 2016, Shchur et al., 2020a, Mei et al., 2019] or attention-based models [Zhang et al., 2020, Zuo et al., 2020, Yang et al., 2022]. While attention-based encoders can provide longer-range contexts, this benefit typically comes at the cost of additional memory overhead. Similarly, a wide range of forms for the event decoder have also been proposed. The most common approach is to parametrize a conditional intensity function via a neural network. For instance, several authors [Mei and Eisner, 2017, Zuo et al., 2020, Zhang et al., 2020] model the conditional intensity using a parametric form inspired by the Hawkes process [Hawkes, 1971], and Du et al. [2016] model the (log-)conditional intensity through an affine function of the history embedding. Similarly, Okawa et al. [2019] model the conditional intensity using a mixture of Gaussian kernels.

Most closely related to our work are approaches which use generative models as decoders. These models often do not assume a parametric form for the decoder, enhancing their flexibility. For instance, Xiao et al. [2017b] propose the use of W-GANs to generate new events. Similarly, Shchur et al. [2020a] learn the distribution over the next inter-arrival time via a normalizing flow. Lin et al. [2022] benchmark several choices of generative models, including diffusion, GANs, and VAEs. Despite the flexibility of these models, these

approaches are all autoregressive in nature, making them ill-suited for multi-step forecasting tasks. In contrast, Lüdke et al. [2023] propose a diffusion-style model which is able to avoid autoregressive sampling via an iterative refinement procedure.

Our approach can be viewed as a novel approach for building flexible decoders for TPPs, extending flow matching to the setting of continuous-time event sequences. In contrast to prior work using generative models, our model is likelihood-free and non-autoregressive, achieving strong performance on long-term forecasting tasks. The work of Lüdke et al. [2023] is perhaps most closely related to ours, but we emphasize that the method of Lüdke et al. [2023] requires an involved training and sampling procedure. In contrast, our method is straightforward to both implement and sample from, while simultaneously outperforming existing approaches.

Flow Matching The recently introduced flow matching framework (or stochastic interpolants) [Lipman et al., 2023, Albergo and Vanden-Eijnden, 2023, Liu et al., 2023] describes a class of generative models which are closely related to both normalizing flows [Papamakarios et al., 2021] and diffusion models [Ho et al., 2020, Song et al., 2021]. Intuitively, these models learn a path of probability distributions which interpolates between a fixed reference distribution and the data distribution. These models are a popular alternative to diffusion, providing greater flexibility in model design, with recent applications in image generation [Ma et al., 2024, Dao et al., 2023], DNA and protein design [Stark et al., 2024, Campbell et al., 2024], and point cloud generation [Buhmann et al., 2023, Wu et al., 2023]. To the best of our knowledge, our approach is the first to explore flow matching for TPPs.

6.2 Autoregressive TPP Models

We first provide a brief review of autoregressive point process models and discuss their shortcomings. Informally, one may think of an event sequence as a set $\{t_k\}_{k=1}^n$ of increasing event times. We will use $\mathcal{H}_t$ to represent the history of a sample up to (and including) time t, i.e., $\mathcal{H}_t = \{t_k : t_k \leq t\}$. Similarly, we use $\mathcal{H}_{t^-} = \{t_k : t_k < t\}$ to represent the history of a sample prior to time t. In the autoregressive setting, the time of a single future event t is modeled conditioned on the observed history of a sequence. This is typically achieved by either directly modeling a distribution over t [Shchur et al., 2020a], or equivalently by modeling a conditional intensity function [Du et al., 2016].

In the first approach, a conditional probability density of the form $p(t \mid \mathcal{H}_{t_n})$ is learned, allowing us to specify a joint distribution over event times $p(t_1, \ldots, t_n)$ autoregressively via $p(t_1, \ldots, t_n) = p(t_1) \prod_{k=2}^n p(t_k \mid \mathcal{H}_{t_{k-1}})$. Alternatively, we may define the *conditional intensity* $\lambda^*(t) := \lambda(t \mid \mathcal{H}_{t^-}) = p(t \mid \mathcal{H}_{t_n})/(1 - F(t \mid \mathcal{H}_{t_n}))$, where $F(t \mid \mathcal{H}_{t_n}) = \int_{t_n}^t p(s \mid \mathcal{H}_{t_n}) \, \mathrm{d}s$ is the CDF associated with $p(t \mid \mathcal{H}_{t_n})$. Informally, the conditional intensity $\lambda^*(t)$ can be thought of [Rasmussen, 2011] as the instantaneous rate of occurrence of events at time t given the previous n events and given that no events have occurred since t_n. By integrating $\lambda^*(t)$, one can show that

$$F(t \mid \mathcal{H}_{t_n}) = 1 - \exp\left(-\int_{t_n}^t \lambda^*(s) \, \mathrm{d}s\right) \qquad p(t \mid \mathcal{H}_{t_n}) = \lambda^*(t) \exp\left(-\int_{t_n}^t \lambda^*(s) \, \mathrm{d}s\right) \tag{6.1}$$

and thus one may recover the conditional distribution from the conditional intensity under mild additional assumptions [Rasmussen, 2011, Prop 2.2].

The Likelihood Function Suppose we observe an event sequence $\{t_k\}_{k=1}^n$ on the interval $[0, T]$. The *likelihood* of this sequence can be seen loosely as the probability of seeing precisely

n events at these times. The likelihood may be expressed in terms of either the density or intensity via

$$L\left(\{t_k\}\right) = p(t_1, \ldots, t_n)\left(1 - F(T \mid \mathcal{H}_{t_n})\right) = \left(\prod_{k=1}^{n} \lambda^*(t_k)\right) \exp\left(-\Lambda^*(T)\right) \tag{6.2}$$

where the CDF term is included to indicate that no events beyond t_n have occurred and $\Lambda^*(T) = \int_0^T \lambda^*(s)\,\mathrm{d}s$ is the total intensity. Autoregressive models are typically trained by maximizing this likelihood [Du et al., 2016, Mei and Eisner, 2017, Shchur et al., 2020a]. We emphasize that this likelihood is not simply the joint event-time density $p_n(t_1, \ldots, t_n)$, as the likelihood measures the fact that no events occur after t_n.

It is worth noting that evaluating $L(\{t_k\})$ can be non-trivial in practice. For models which parametrize $\lambda^*(t)$ via a neural network [Du et al., 2016, Mei and Eisner, 2017], computing the total intensity $\Lambda^*(T)$ is often done via a Monte Carlo integral, requiring several forward passes of the model to evaluate $\lambda^*(t)$ at different values of t. Models which directly parametrize the density $p(t \mid \mathcal{H}_t)$ suffer from the same drawback when computing the corresponding CDF in Equation (6.2). Moreover, some approaches, such as the diffusion-based approach of Lin et al. [2022], are only trained to maximize an ELBO of $p(t \mid t_1, \ldots, t_n)$, and are thus unable to compute the proper likelihood in Equation (6.2).

Sampling Autoregressive Models In many tasks, we are interested not only in an accurate model of the intensity (or distribution), but also sampling new event sequences from the corresponding distribution. For instance, when forecasting an event sequence, we may want to generate several forecasts in order to provide uncertainty quantification over these predictions. However, sampling from existing autoregressive models can be difficult.

For instance, the flow-based model of Shchur et al. [2020a] requires a numerical approximation to the inverse of the model to perform sampling. Similarly, the diffusion-based approach of

Lin et al. [2022] can require several hundred forward passes of the model to generate a single event time, rendering it costly when generating long sequences. Moreover, the predictive performance of autoregressive models is often unsatisfactory on multi-step generation tasks due to the accumulation of errors over many steps [Lin et al., 2021, Lüdke et al., 2023]. This difficulty is particularly pronounced for intensity-based models [Du et al., 2016, Mei and Eisner, 2017, Zhang et al., 2020], where naively computing the implied distribution in Equation (6.1) is prohibitively expensive. Instead, sampling from intensity-based models is typically achieved via the thinning algorithm [Ogata, 1981, Lewis and Shedler, 1979]. However, this algorithm has several hyperparameters to tune, is challenging to parallelize, and can be difficult for practitioners to implement [Xue et al., 2024].

6.3 EventFlow

Motivated by the limitations of autoregressive models, we propose `EventFlow`, which has a number of distinct advantages over existing approaches. First, `EventFlow` directly models the joint distribution over event times, thereby avoiding autoregression entirely. Second, our model is likelihood-free, avoiding the Monte Carlo estimates needed to estimate the likelihood in Equation (6.2) during training. Third, sampling from our model amounts to solving an ordinary differential equation. This is straightforward to implement and parallelize, allowing us to avoid the difficulties of thinning-based approaches used in existing models. We build upon the flow matching (or stochastic interpolant) framework [Lipman et al., 2023, Albergo and Vanden-Eijnden, 2023, Liu et al., 2023] to develop our model. We begin below by focusing on the unconditional setting, and later discuss how to extend our method for conditional generation as necessary.

6.3.1 Preliminaries

We first introduce some necessary background and notation. Let $\mathcal{T} = [0, T]$ be a finite-length interval. The set Γ denotes the *configuration space* of $\mathcal{T}$ [Albeverio et al., 1998], i.e., the set of all finite counting measures on the set $[0, T]$. A point $\gamma \in \Gamma$ corresponds to a measure of the form $\gamma = \sum_{k=1}^{n} \delta[t^k]$, i.e., a finite collection of Dirac deltas located at event times $t^k \in \mathcal{T}$. A *temporal point process (TPP)* on $\mathcal{T}$ is a probability distribution μ over the configuration space Γ. Informally, μ represents a distribution over sequences γ living in the configuration space Γ which constitutes the set of valid sequences. We use $N : \Gamma \to \mathbb{Z}_{\geq 0}$ to denote the counting functional, i.e., $N(\gamma)$ is the number of events in the TPP realization γ.[1] While it is common to represent TPPs as distributions over random sets of event times, in our approach it will be more convenient to represent TPPs as random measures [Kallenberg, 2017].

We assume all TPP distributions are atomless [Kallenberg, 2017, Ch. 1], i.e., the probability of observing an event at any singleton is zero. In addition, we assume all TPPs are simple [Kallenberg, 2017, Ch. 2], i.e., no more than one event can occur simultaneously. Under these assumptions, a TPP μ can be fully characterized [Daley and Vere-Jones, 2003, Prop. 5.3.II] by a probability distribution which specifies the number of events and a *collection* of joint densities corresponding to the event times themselves. In a slight abuse of notation, we will write $\mu(n)$ for the corresponding distribution over event counts, and $\{\mu^n(t^1, \ldots, t^n)\}_{n=1}^{\infty}$ for the collection of joint distributions. In other words, for any given $n \in \mathbb{Z}_{\geq 0}$, the probability of observing n events in the interval $\mathcal{T}$ is $\mu(n)$, and $\mu^n(t^1, \ldots, t^n)$ describes the corresponding joint distribution of event times. We further restrict each μ^n to be supported only on the ordered sets, so that we may assume $t^1 < t^2 < \cdots < t^n$.

Let μ_1 represent the data distribution and μ_0 represent a reference distribution. That is,

[1] This can be thoughts of in terms of the counting process, i.e. $N(\gamma)$ corresponds to the value of the associated counting process at the ending time T, or the total number of events in γ that have occurred in the interval $[0, T]$.

both $\mu_0, \mu_1 \in \mathbb{P}(\Gamma)$ are TPP distributions. To construct our model, we will define a path of TPPs $\eta_s \in \mathbb{P}(\Gamma)$ which approximately interpolates from our reference distribution to our data distribution. Throughout, we use $s \in [0, 1]$ to denote a flow time and $t \in [0, T]$ to denote an event time. These two time axes are in a sense orthogonal to one another (see Figure 6.1).

6.3.2 Balanced Couplings

Our first step is to define a useful notion of couplings [Villani, 2009], allowing us to pair event sequences drawn from μ_0 with those drawn from μ_1. A *coupling* between two TPPs $\mu, \nu \in \mathbb{P}(\Gamma)$ is a joint probability measure $\rho \in \mathbb{P}(\Gamma \times \Gamma)$ over pairs of event sequences (γ_0, γ_1) such that the marginal distributions of ρ are μ and ν. We say that the coupling ρ is *balanced* if draws $(\gamma_0, \gamma_1) \sim \rho$ are such that $N(\gamma_0) = N(\gamma_1)$ almost surely. In other words, balanced couplings only pair event sequences with equal numbers of events.

While we will later see how to interpolate between any two given event sequences, this coupling will allow us to decide *which* sequences to interpolate. In particular, a balanced coupling will allow us to pair sequences such that they always have the same number of events, allowing us to avoid the addition or deletion of events during both training and sampling and thus simplifying our model. We will use $\Pi_b(\mu, \nu)$ to denote the set of balanced couplings of μ, ν, and the following proposition shows $\Pi_b(\mu, \nu)$ is nonempty if and only if the event count distributions of μ and ν are equal, placing a structural constraint on the suitable choices of a reference measure.

Proposition 34 (Existence of Balanced Couplings)**.**
Let $\mu, \nu \in \mathbb{P}(\Gamma)$ be two TPPs. The set of balanced couplings $\Pi_b(\mu, \nu)$ is nonempty if and only if $\mu(n) = \nu(n)$ have the same distribution over event counts.

Proof. Let $A_1, A_2 \subseteq \Gamma$ be Borel measurable [Daley and Vere-Jones, 2003, Prop. 5.3] subsets

of the configuration space Γ, i.e. each of A_1, A_2 is a measurable collection of event sequences. Observe that for $i = 1, 2$, each A_i can be written as a disjoint union

$$A_i^n = \bigcup_{n=0}^{\infty} \mathcal{T}^n \cap A_i \tag{6.3}$$

i.e. $A_i^n \subseteq A_i$ is the subset of A_i containing only sequences with n events. Note each A_i^n is a Borel measurable subset of $\mathcal{T}^n$.

Now, suppose that $\mu(n) = \nu(n)$ have equal event count distributions. We define the coupling $\rho \in \mathbb{P}(\Gamma \times \Gamma)$ by

$$\rho(A_1 \times A_2) = \sum_{n=0}^{\infty} \mu(n)\mu^n(A_1^n)\nu^n(A_2^n). \tag{6.4}$$

Here, in a slight abuse of notation, we use μ^n, ν^n to denote the corresponding joint probability measures over n events, i.e., both are Borel probability measures on $\mathcal{T}^n$. Since the n-dimensional projection of Γ in Equation (6.3) is simply $\mathcal{T}^n$, it is immediate that $\rho(A_1 \times \Gamma) = \mu(A_1)$ and $\rho(\Gamma \times A_2) = \nu(A_2)$, so that ρ is indeed a coupling. Moreover, it is clear that the coupling is balanced.

Conversely, suppose $\rho \in \Pi_b(\mu_0, \mu_1)$ is a balanced coupling. Let $N : \Gamma \to \mathbb{Z}_{\geq 0}$ be the event counting functional and let $\pi^1, \pi^2 : \Gamma \times \Gamma \to \Gamma$ denote the canonical projections of $\Gamma \times \Gamma$ onto its components. That is, $\pi^1 : (\gamma_0, \gamma_1) \mapsto \gamma_0$ and $\pi^2 : (\gamma_0, \gamma_1) \mapsto \gamma_1$. Furthermore, let $(N, N) : \Gamma \times \Gamma \to \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ denote the product of the counting functional, i.e. $(N, N)(\gamma_0, \gamma_1) = (N(\gamma_0), N(\gamma_1))$. Note that the pushforward $N_{\#}\mu$ yields the event count distribution $\mu(n)$ of μ (and analogously for ν).

Now, observe that composing the projections and counting functionals yields

$$\pi^1 \circ (N, N) = N \circ \pi^1 \qquad \pi^2 \circ (N, N) = N \circ \pi^2. \tag{6.5}$$

As ρ is a coupling, we have that $\mu = \pi^1_\# \rho$ and $\nu = \pi^2_\# \rho$. From these observations, it follows that

$$\begin{aligned} N_\# \mu = N_\# \left(\pi^1_\# \rho\right) = (N \circ \pi^1)_\# \rho & \tag{6.6} \\ = (\pi^1 \circ (N, N))_\# \rho = \pi^1_\# \left((N, N)_\# \rho\right) & \tag{6.7} \\ = \pi^2_\# \left((N, N)_\# \rho\right) = N_\# \nu & \tag{6.8} \end{aligned}$$

where the equality in the penultimate line follows from the fact that ρ is balanced. Thus, we have shown that the existence of a balanced coupling implies that $N_\# \mu = N_\# \nu$, i.e. the event count distributions are equal. □

In practice, we follow a simple strategy for choosing both the reference TPP μ_0 and the coupling ρ. Suppose q is any given density on our state space $\mathcal{T}$, e.g., a uniform distribution. We take μ_0 to be a mixed binomial process [Kallenberg, 2017, Ch. 3] whose event count distribution is given by that of the data $\mu_1(n)$ and joint event distributions given by independent products of q (up to sorting). That is, to sample from μ_0, one can simply sample $n \sim \mu_1(n)$ from the empirical event count distribution followed by sampling and sorting n i.i.d. points $t^k \sim q$.

To draw a sample from our coupling ρ, we first sample a data sequence $\gamma_1 \sim \mu_1$, followed by sampling $N(\gamma_1)$ events independently from q and sorting to produce a draw $\gamma_0 \sim \mu_0$. We call this coupling the *independent balanced coupling* of μ and ν.

6.3.3 Interpolant Construction

We now proceed to construct our interpolant $\eta_s \in \mathbb{P}(\Gamma)$. We will construct this path of TPPs via a local procedure which we then marginalize over a given balanced coupling. To that end, let ρ be any balanced coupling of the reference measure μ_0 and the data measure μ_1, and suppose $z := (\gamma_0, \gamma_1) \sim \rho$ is a draw from this coupling. As ρ is balanced, we have that

$$\gamma_0 = \sum_{k=1}^{n} \delta[t_0^k] \qquad \gamma_1 = \sum_{k=1}^{n} \delta[t_1^k] \tag{6.9}$$

are both a collection of n events. As TPPs are fully characterized by their joint distributions over event times, we will henceforth describe our procedure for a fixed (but arbitrary) number of events n. First, we the define measure $\gamma_s^z \in \Gamma$ via

$$\gamma_s^z = \sum_{k=1}^{n} \delta[t_s^k] \qquad t_s^k = (1-s)t_0^k + st_1^k \qquad 0 \le s \le 1 \tag{6.10}$$

where we use the superscript z to denote the dependence on the pair $z = (\gamma_0, \gamma_1)$. In other words, γ_s^z linearly interpolates each corresponding event in γ_0 and γ_1. This defines a path $(\gamma_s^z)_{s=0}^1$ in the configuration space Γ which evolves the reference sample γ_0 into the data sample γ_1.

In order to perform the marginalization step, we now lift this deterministic path $(\gamma_s^z)_{s=0}^1 \subset \Gamma$ to a path of TPP distributions $(\eta_s^z)_{s=0}^1 \subset \mathbb{P}(\Gamma)$. We define the point process distribution $\eta_s^z \in \mathbb{P}(\Gamma)$ implicitly by adding independent Gaussian noise to each of the events in γ_s^z. That is, a draw $\hat{\gamma}_s^z \sim \eta_s^z$ may be simulated via

$$\hat{\gamma}_s^z = \sum_{k=1}^{n} \delta\left[t_s^k + \epsilon^k\right] \qquad \epsilon^k \sim \mathcal{N}(0, \sigma^2). \tag{6.11}$$

In principle this means that the support of η_s^z is larger than $\mathcal{T}$, but in practice we choose σ^2

sufficiently small such that this is not a concern. The addition of noise ϵ_k is instrumental in obtaining a well-specified model, but in practice the noise variance σ^2 is not a critical hyperparameter. We note that this noising step is typical in flow matching models [Lipman et al., 2023, Tong et al., 2024].

Finally, for any $s \in [0,1]$, we define the marginal TPP measure η_s via $\eta_s = \int \eta_s^z \, \mathrm{d}\rho(z)$. Observe that, by construction, the event count distribution $\eta_s(n)$ is given by $\mu_1(n)$ for all $s \in [0,1]$. This path of TPP distributions η_s approximately interpolates from the reference TPP μ_0 at $s = 0$ to the data TPP μ_1 at $s = 1$, in the sense that at the endpoints, the joint event time distributions $\eta_0^n(t^1, \ldots, t^n)$ and $\eta_1^n(t^1, \ldots, t^n)$ are given by a convolution of $\mu_0^n(t^1, \ldots, t^n)$ and $\mu_1^n(t^1, \ldots, t^n)$ with the Gaussian $\mathcal{N}(0, \sigma^2 I_n)$. As $\sigma^2 \downarrow 0$, it is clear that we recover a genuine interpolant [Tong et al., 2024].

We now shift our attention to the transport of a single event t_s^k for a fixed k. Through the addition of Gaussian noise, we have constructed a path of Gaussian distributions $\mathcal{N}(t_s^k, \sigma^2)$ whose mean is determined by the location of the kth event at the flow time s. This transport of a Gaussian can be achieved infinitesimally through the constant vector field $v_s^k : [0,T] \to \mathbb{R}$ given by $v_s^k(t) = t_1^k - t_0^k$ [Tong et al., 2024]. Thus, the evolution in (6.11) is generated by the vector field $v_s^z : \mathcal{T}^n \to \mathbb{R}^n$ given by

$$v_s^z(\gamma) = \begin{bmatrix} v_s^1, \ldots, v_s^n \end{bmatrix}^\mathsf{T} = \begin{bmatrix} t_1^1 - t_0^1, & \ldots, & t_1^n - t_0^n \end{bmatrix}^\mathsf{T} \qquad 0 \le s \le 1. \tag{6.12}$$

Informally, we view $v_s^z : \mathcal{T}^n \to \mathbb{R}^n$ as prescribing a constant (but different) velocity to each of the n events. For a fixed pair $z = (\gamma_0, \gamma_1)$ and a given sample $\gamma_0' \sim \eta_0^z$, solving the system of ordinary differential equations $\mathrm{d}\gamma_s' = v_s^z(\gamma_s') \, \mathrm{d}s$ with initial condition γ_0' will result in a collection of events which is concentrated around the true event times γ_1. Note that here, we

view this differential equation as an ODE in $\mathcal{T}^n$. If we draw many samples $\gamma_0 \sim \eta_0^z$ and solve the corresponding ODE, the distribution over events at any intermediate time s will be given by η_s^z.

In other words, the vector field v_s^z generates the path of distributions η_s^z. However, this path is conditioned on z, and we would like to find the vector field v_s which generates the *unconditional* path η_s. As is standard in flow matching [Lipman et al., 2023, Tong et al., 2024, Albergo and Vanden-Eijnden, 2023], the unconditional vector field $v_s : \mathcal{T}^n \to \mathbb{R}^n$ may be obtained through a weighted marginalization procedure via

$$v_s(\gamma) = \int v_s^z(\gamma) \frac{\mathrm{d}\eta_s^z}{\mathrm{d}\eta_s}(\gamma) \, \mathrm{d}\rho(z). \tag{6.13}$$

We have thus far described a procedure for interpolating between a given reference distribution μ_0^n and the data distribution μ_1^n for a given, fixed number of events n. As n was arbitrary, we have successfully constructed a family of interpolants which will enable us to sample from the joint event distribution for any n. However, to fully characterize the TPP distribution, we need to also specify the event count distribution. For unconditional generation tasks, this is straightforward – we simply follow the empirical event count distribution see in the training data. We describe our approach for modeling the event count distribution in conditional tasks in the following section.

6.3.4 Training, Parametrization, and Sampling

To train the model, we would like to perform regression on the vector fields v_s in Equation (6.13). If we knew this vector field v_s, we could draw samples from the data TPP by drawing a sample event sequence $\gamma_0 \sim \mu_0$ from the reference TPP, and flowing each event along the vector field v_s.

Algorithm 1: Training Step for `EventFlow`

Sample $\gamma_1 \sim \mu_1$, $s \sim \mathcal{U}[0,1]$, and $\epsilon \sim \mathcal{N}(0,1)$
$e_{\mathcal{H}} = \varnothing$ `/* Null history */`
if *forecast* **then**
    Sample split time $T_0 \in [\Delta T, T - \Delta T]$
    Construct history $\mathcal{H} \leftarrow \{t \in \gamma_1 : t \leq T_0\}$
    Embed history $e_{\mathcal{H}} \leftarrow f_\theta(\mathcal{H})$
    Construct future $\gamma_1 \leftarrow \{t \in \gamma_1 : T_0 < t \leq T_0 + \Delta T\}$
Set $n \leftarrow N(\gamma_1)$
Sample $t_0^1, \ldots, t_0^n \sim q$ and sort to construct γ_0
Compute γ_s^z via $t_s^k = (1-s)t_0^k + st_s^k$
Take a gradient step on $\left\|\gamma_1 - \gamma_0 - v_\theta\left(\gamma_s^z + \epsilon, s, e_{\mathcal{H}}\right)\right\|^2$

Training Foremost, although the marginal vector field in Equation (6.13) admits an analytical form, it is intractable to compute in practice as the marginal measure η_s is not available. To overcome this, we may instead perform regression on the *conditional* vector fields v_s^z. Here, $v_\theta(\gamma_s, s)$ will represent a neural network with parameters θ which takes in a sequence γ_s of $N(\gamma_s) = n$ event times, along with the flow time s. That is, we seek to minimize the loss

$$J(\theta) = \mathbb{E}_{s,(\gamma_0,\gamma_1),\hat{\gamma}_s^z}\left[\left\|\gamma_1 - \gamma_0 - v_\theta(\hat{\gamma}_s^z, s)\right\|^2\right] \tag{6.14}$$

which can be shown to be equal to MSE regression on the *unconditional* v_s up to an additive constant not depending on θ [Lipman et al., 2023, Tong et al., 2024]. We note here that, although the regression target v_s^z is linear, the unconditional vector field v_s will in general be nonlinear. In practice, this loss is estimated by uniformly sampling a flow time $s \in [0,1]$, a pair $z = (\gamma_0, \gamma_1) \sim \rho$ from our balanced independent coupling, and drawing a noisy interpolant $\hat{\gamma}_s^z \sim \eta_s^z$ according to Equation (6.11).

To train the model on a forecasting task, where the goal is to predict a future sequence of events conditioned on a history $\mathcal{H}$, we embed $\mathcal{H}$ into a fixed-dimensional vector representation $e_{\mathcal{H}} = f_\theta(\mathcal{H})$ via a learned encoder f_θ before providing this to the model $v_\theta(\gamma_s, s, e_{\mathcal{H}})$ and

minimizing Equation (6.14). Note that we jointly train the encoder f_θ and vector field v_θ. See Algorithm 1.

Parametrization The second challenge is that we must learn a vector field $v_\theta(\gamma, s)$ in n dimensions for arbitrary values of n. In other words, v_θ is a neural network which takes in a flow time $s \in [0, 1]$ and a sequence of events γ, and must produce $N(\gamma)$ scalar outputs. We achieve this through an attention-based architecture, which we detail in Appendix D.2. At a high level, the flow time s is transformed via a learnable embedding into a fixed-dimensional vector. Similarly, each event in γ is transformed into fixed-dimensional vector via a learned embedding (which is shared across the events, but not the flow time). The flow-time embedding is then added to each event embedding, and the resulting sequence is passed through a standard transformer architecture [Yang et al., 2022, Vaswani et al., 2017], resulting in a sequence of $N(\gamma)$ vectors. Finally, each of these vectors is projected into one dimension via a linear layer to produce the sequence of $N(\gamma)$ velocities.

For conditional tasks, we must also compute an encoding $e_{\mathcal{H}} = f_\theta(\mathcal{H})$ of the history $\mathcal{H}$. This is done by a separate transformer encoder, which operates in the same fashion as described in the previous paragraph, but without the use of the flow-time s as an input and without the final linear projection layer. This embedding is fed into the intermediate layers of our velocity network via cross-attention.

Lastly, for forecasting tasks we must also learn a model of the event count distribution $p_\phi(n \mid \mathcal{H})$. We treat this as a classification problem, where the goal is to predict the number of events n occurring in the forecast window given the history $\mathcal{H}$. While this does imply there is some maximal number of events, we in practice set this to the maximal sequence length seen in the training set, which is typically much larger than the typical sequence length. We again use an attention-based model, analogous to our velocity model, but where we aggregate the final sequence embedding by averaging and passing this through a small MLP. The model

Algorithm 2: Sampling Step for `EventFlow`

Choose a flow time discretization $0 = s_0 < s_1 < \cdots < s_K = 1$
$e_{\mathcal{H}} = \varnothing$ `/* Null history */`
if *forecast* **then**
  Embed history $e_{\mathcal{H}} \leftarrow f_\theta(\mathcal{H})$
  Sample $n \sim p_\phi(n \mid \mathcal{H})$
else
  Sample $n \sim \mu_1(n)$
Sample $t_0^1, \ldots, t_0^n \sim q$ and sort to construct γ_0
for $k = 1, 2, \ldots, K$ **do**
  $h_k \leftarrow s_k - s_{k-1}$
  $\gamma_{s_k} \leftarrow \gamma_{s_{k-1}} + h_k v_\theta(\gamma_{s_{k-1}}, s_{k-1}, e_{\mathcal{H}})$

$p_\phi(n \mid \mathcal{H})$ is trained by minimizing the cross-entropy loss.

Sampling Once the network v_θ is learned, we may draw samples from the model by first drawing a reference sequence $\gamma_0 \sim \mu_0$ and solving the corresponding system of ODEs parametrized by v_θ. More concretely, we first fix a number of events n to generate. When seeking to unconditionally generate new sequences from the underlying data TPP μ_1, we simply sample n from the empirical event count distribution $\mu_1(n)$. For conditional tasks, we draw $n \sim q(n \mid \mathcal{H})$ from the learned conditional distribution over event counts.

Next, we draw n initial events, corresponding to $s = 0$, by sampling and sorting $t_0^1, \ldots, t_0^n \sim q$. In practice, we use $q = \mathcal{N}(0, I_n)$ as we normalize our sequences into the range $[-1, 1]$ during training and sampling (followed by renormalization to the data scale). Since we have fixed n, we may view this initial draw as a vector $\gamma_0 = [t_0^1, \ldots, t_0^n] \in \mathcal{T}^n$. This event sequence γ_0 then serves as the initial condition for the system of ODEs $d\gamma_s = v_\theta(\gamma_s, s)\, \mathrm{d}s$ which can be solved numerically. In our experiments, we use the forward Euler scheme, i.e., we specify a discretization $\{0 = s_0 < s_1 < \cdots < s_K = 1\}$ of the flow time (in practice, uniform) and recursively compute

$$\gamma_{s_k} = \gamma_{s_{k-1}} + h_k v_\theta(\gamma_{s_{k-1}}, s_{k-1}) \qquad k = 1, 2, \ldots, K \tag{6.15}$$

where $h_k = s_k - s_{k-1}$ represents a scalar step size. While other choices of numerical solvers are certainly possible, we found that this simple scheme was sufficient as the model sample paths are typically close to linear. See Algorithm 2 for the full procedure.

6.4 Experiments

We study our proposed `EventFlow` model under two settings. The first is a conditional generative modeling task, where we seek to forecast both the number and times of future events given a history. The second is an unconditional task, where we aim to learn a representation of the underlying TPP distribution from empirical observations and generate new sequences from this distribution. In a sense, this second task can be viewed as a special case of the first with no observed history. Our overall experimental procedure is inspired by that of Lüdke et al. [2023].

Datasets We evaluate our model across a diverse set of datasets encompassing a wide range of possible point process behaviors. First, we use a collection of six synthetic datasets produced by Omi et al. [2019]. We additionally evaluate our model on seven real-world datasets, which are a standard benchmark for modeling unmarked TPPs [Shchur et al., 2020b, Bosser and Taieb, 2023, Lüdke et al., 2023]. See Appendix D.1 for additional information regarding our datasets.

Baseline Models As a point of comparison, we train and evaluate several baseline models. These models were selected as they constitute a set of diverse and highly performant models. For an intensity-based method, we compare against the Neural Hawkes Process (NHP) [Mei and Eisner, 2017]. We additionally compare against two intensity-free methods, namely the flow-based IFTPP model [Shchur et al., 2020a] and the diffusion-based model of Lin et al.

[2022]. Lastly, our strongest baseline is the recently proposed Add-and-Thin model of Lüdke et al. [2023], which can be loosely viewed as a non-autoregressive diffusion model. These models use an RNN-based history encoder, with the exception of Add-and-Thin which uses a CNN-based encoder. See Appendix D.3 for additional details.

Metrics Evaluating generative TPP models is challenging, as one must take into account both the variable locations and numbers of events. This is particularly challenging for the unconditional setting, where unlike forecasting, we do not have a ground-truth sequence to compare against. Our starting point is a metric [Xiao et al., 2017a] on the space of sequences Γ, allowing us to measure the distance between two sequences $\gamma = \sum_{k=1}^{n} \delta[t_k^\gamma]$ and $\eta = \sum_{k=1}^{m} \delta[t_k^\eta]$ with possibly different numbers of events. Without loss of generality, we assume $n \leq m$, so the distance is given by

$$d(\gamma, \eta) = \sum_{k=1}^{n} |t_k^\gamma - t_k^\eta| + \sum_{k=n+1}^{m} (T - t_k^\eta) \tag{6.16}$$

where we recall that we assume the sequences are supported on $\mathcal{T} = [0, T]$. This distance can be understood either as an L^1 distance between the corresponding counting processes of γ, η or as a generalization of the 1-Wasserstein distance to measures of unequal mass. This metric allows us to compare two sequences of possibly unequal lengths, and will be used in our forecasting experiment.

For our unconditional experiment, we require a metric which will capture the distance between the TPP distributions themselves. To do so, we follow the approach suggested by Shchur et al. [2020b] and Lüdke et al. [2023], and use the distance in Equation (6.16) to calculate an MMD [Gretton et al., 2012]. Namely, the MMD between two point process distributions

$\mu, \nu \in \mathbb{P}(\Gamma)$ is given by

$$\mathsf{MMD}(\mu, \nu) = \mathbb{E}_{\gamma,\gamma' \sim \mu}[k(\gamma, \gamma')] - 2\mathbb{E}_{\gamma \sim \mu, \eta \sim \nu}[k(\gamma, \eta)] + \mathbb{E}_{\eta,\eta' \sim \nu}[k(\eta, \eta')] \tag{6.17}$$

where $k : \Gamma \to \Gamma \to \mathbb{R}_{\geq 0}$ is a specified kernel. In accordance with prior work [Shchur et al., 2020b, Lüdke et al., 2023], we use an exponential kernel $k(\gamma, \eta) = \exp\left(-d(\gamma, \eta)/(2\sigma^2)\right)$ with σ chosen to be the median distance between all sequences.

6.4.1 Forecasting Event Sequences

We first evaluate our model on a multi-step conditional forecasting task, focusing on the real-world datasets. To do so, we set a forecast horizon ΔT for each of our real-world datasets, and generate event sequences in the range $[T_0, T_0 + \Delta T]$ for some given T_0, conditioned on the history of events $\mathcal{H}_{T_0}$. Up to a shift, this means we are taking $\mathcal{T} = [0, \Delta T]$ as the support for our model TPP. The forecast horizon ΔT is chosen such that the window typically contains multiple events.

At training time, we uniformly sample $T_0 \in [\Delta T, T - \Delta T]$ and split a given data sequence γ_1 into a history on $[0, T_0]$ and a future $[T_0, T_0 + \Delta T]$. We encode the history $\mathcal{H}_{T_0}$ before training the model to fit the events occurring in the future. At testing time, we perform the same splitting procedure, sampling 50 values of T_0 for each test set sequence. We then forecast the sequence in $[T_0, T_0 + \Delta T]$ via the model and compute the distance (6.16) between the ground-truth and generated sequences. Importantly, we note that the distance in Equation (6.16) is computed using $T_0 + \Delta T$ rather than T as the maximum event time, as using T would result in a distance which is sensitive to the location of the forecasting window within the support $[0, T]$. We further normalize Equation (6.16) by ΔT to allow for comparison across different window lengths.

Table 6.1: Sequence distance (6.16) between the forecasted and ground-truth event sequences on a held-out test set. Lower is better. We report the mean ± one standard deviation over five random seeds. The best mean distance on each dataset is indicated in bold, and the second best by an underline.

	PUBG	Reddit-C	Reddit-S	Taxi	Twitter	Yelp-A	Yelp-M
IFTPP	4.2±0.7	25.6±2.3	61.2±3.2	5.1±0.4	2.9±0.2	2.1±0.2	3.4±0.2
NHP	<u>2.8</u>±0.1	31.0±1.4	95.7±0.7	4.5±0.3	3.4±0.5	<u>1.8</u>±0.1	<u>3.0</u>±0.2
Diffusion	5.4±1.2	25.7±0.9	80.3±11.4	4.6±0.7	<u>2.4</u>±0.2	<u>1.8</u>±0.1	3.3±0.7
Add-and-Thin	**2.5**±0.04	**22.2**±4.6	<u>34.3</u>±0.4	**3.7**±0.1	3.1±0.2	<u>1.8</u>±0.1	<u>3.0</u>±0.2
EventFlow (25 NFEs)	<u>2.8</u>±0.7	<u>22.6</u>±2.7	**21.5**±0.4	**3.7**±0.1	**1.7**±0.1	**1.4**±0.04	**2.1**±0.1
EventFlow (10 NFEs)	2.8±0.7	22.6±2.7	21.3±0.4	3.5±0.2	1.7±0.1	1.4±0.04	2.1±0.1
EventFlow (1 NFE)	2.7±0.7	22.6±2.7	21.1±0.3	3.7±0.4	1.8±0.1	1.6±0.2	2.1±0.1
EventFlow (25 NFEs, true n)	1.2±0.01	5.5±0.3	8.8±0.2	1.8±0.02	0.7±0.01	0.7±0.02	1.1±0.02

We report the results of this experiment in Table 6.1. Our proposed `EventFlow` method obtains the lowest average forecasting error on 5/7 of the datasets, and closely matches the performance of Add-and-Thin on the remaining 2/7 datasets. Given that the non-autoregressive models (`EventFlow`, Add-and-Thin) consistently outperform the autoregressive baselines, this is clear evidence that autoregressive models can struggle on multi-step predictions. This is especially true on the Reddit-C and Reddit-S datasets, which exhibit long sequence lengths.

In Table 6.2, we provide additional evaluations of the event count distributions alone. To do so, we measure the mean absolute relative error (MARE) given by

$$\text{MARE} = \mathbb{E}_{n,\hat{n}} \left| \frac{\hat{n} - n}{n} \right| \tag{6.18}$$

where n represents the true number of points in a sequence, $\hat{n}$ represents the predicted number of points, and the expectation is estimated empirically on the testing set. As our method directly predicts the number of events n by sampling from the learned distribution $p_\phi(n \mid \mathcal{H})$, this serves as a direct evaluation of this component of our model. Here, we find that Add-and-Thin has strong performance (mean rank: 1.3), whereas our method (mean rank: 3), diffusion (mean rank: 3.1) perform comparably, while IFTPP (mean rank: 3.6) and NHP lag slightly behind (mean rank: 4). While our method has room for improvement,

Table 6.2: MARE values evaluating the predicted number of events when forecasting. Mean values ± one standard deviation are reported over five random seeds. The lowest MARE on each dataset is indicated and bold, and the second lowest is indicated by an underline.

	PUBG	Reddit-C	Reddit-S	Taxi	Twitter	Yelp-A	Yelp-M
IFTPP	1.05±0.14	1.69±0.39	0.79±0.20	0.60±0.11	0.88±0.08	0.76±0.07	0.76±0.05
NHP	1.02±0.08	**0.95**±0.01	1.00±0.0004	0.67±0.11	2.48±0.40	0.80±0.22	1.07±0.34
Diffusion	1.95±0.48	1.28±0.09	1.12±0.56	0.49±0.07	<u>0.66</u>±0.04	0.65 ± 0.07	<u>0.72</u>±0.07
Add-and-Thin	**0.43**±0.01	<u>0.99</u>±0.10	<u>0.38</u>±0.01	**0.33**±0.02	**0.60**±0.02	**0.42**±0.01	**0.46**±0.03
Ours	<u>0.69</u>±0.17	2.01±0.40	**0.26**±0.01	<u>0.47</u>±0.03	1.23±0.07	<u>0.66</u>±0.03	0.80±0.05

we note that even though our approach to learning $p(n \mid H)$ is quite simple it still achieves competitive results. Designing better techniques for predicting the event counts is an exciting direction for future work. However, we emphasize that our model shows clear gains on the forecasting metric which measures both the event counts and their times, and this is the primary relevant metric for the problem we address in this paper.

Ablations We additionally perform two ablations. First, we vary the number of function evaluations (NFEs) used at sampling time, i.e., steps in Equation (6.15). We find that 10 NFEs is sufficient, and increasing the NFEs further does not result in significant gains. Interestingly, with only one step, we observe only a small drop in forecasting performance. This is enabled by our carefully designed interpolant construction (Equation (6.10)). We emphasize that Add-and-Thin uses 100 NFEs at generation time and the diffusion model uses 1000 NFEs *per generated event*. The autoregressive baselines (NHP, IFTPP) require one NFE per generated event. Thus, our method is able to simultaneously obtain strong forecasting performance while only requiring a small number of model evaluations. In our second ablation, we do not sample $n \sim p_\phi(n \mid \mathcal{H})$, but rather set n to be the true number of events in the forecast window. While this is not practical, this serves to ablate the effect of errors in the event counts. We see that the forecasting error improves significantly, indicating that designing stronger techniques for modeling $p_\phi(n \mid \mathcal{H})$ can lead to improved forecasts.

Table 6.3: MMDs (1e-2) between the test set and 1,000 generated sequences on our synthetic datasets. Lower is better. We report the mean ± one standard deviation over five random seeds. The lowest MMD distance on each dataset is indicated in bold, and the second lowest is indicated by an underline.

	Hawkes1	Hawkes2	NSP	NSR	SC	SR
Data	1.3	1.3	1.8	3.0	5.7	1.1
IFTPP	**1.5**±0.4	**1.4**±0.5	**2.3**±0.7	6.2±2.1	**5.8**±0.5	**1.3**±0.3
NHP	1.9±0.3	5.2±1.6	3.6±1.3	12.6±1.8	25.4±11.5	5.0±0.7
Diffusion	4.8±2.7	5.5±3.3	10.8±7.5	15.0±3.6	9.1±1.8	5.1±2.8
Add-and-Thin	1.9±0.5	2.5±0.3	2.6±0.5	7.4±1.2	22.5±0.5	2.2±0.8
EventFlow (ours)	1.9±0.2	2.2±0.1	3.8±1.2	**4.2**±0.5	8.3±0.4	1.7±0.3

Table 6.4: MMDs (1e-2) between the test set and 1,000 generated sequences on our real-world datasets. Lower is better. We report the mean ± one standard deviation over five random seeds. The lowest MMD distance on each dataset is indicated in bold, and the second lowest is indicated by an underline.

	PUBG	Reddit-C	Reddit-S	Taxi	Twitter	Yelp-A	Yelp-M
Data	1.3	0.6	0.4	3.1	2.6	3.6	3.1
IFTPP	5.7±1.8	1.3±1.2	1.9±0.6	5.8±0.9	**2.9**±0.6	8.2±4.7	5.1±0.7
NHP	7.2±0.2	2.2±1.6	22.5±0.3	5.0±0.1	7.3±0.7	6.7±1.5	6.1±2.3
Diffusion	14.3±6.5	3.9±1.2	6.2±3.3	11.7±1.8	12.5±1.9	10.9±3.8	10.5±5.2
Add-and-Thin	2.8±0.5	1.2±0.27	2.7±0.3	5.2±0.6	4.8±0.4	**4.5**±0.2	**3.0**±0.5
EventFlow (ours)	**1.5**±0.2	**0.7**±0.1	**0.7**±0.1	**3.5**±0.1	4.9±0.7	6.6±1.2	**3.0**±0.5

6.4.2 Unconditional Generation of Event Sequences

Next, we evaluate our model on an unconditional generation task, where we aim to generate new sequences from the underlying data distribution. This task serves as a benchmark to evaluate the methods in terms of how well they are able to fit the underlying TPP. Moreover, learning a general-purpose TPP prior could enable downstream tasks, such as data augmentation [Graikos et al., 2022].

In Tables 6.3 and 6.4, we report MMD values (6.17) for each of the synthetic and real-world datasets. We tune the hyperparameters for all models and perform model selection based on the validation set MMD. MMDs are calculated by sampling 1,000 sequences from each

trained model, and estimating Equation (6.17) using the generated and test set samples. The first row ("data") is the MMD calculated between samples in the training and validation sets, giving us a sense of the best-case performance.

Overall, we find that our `EventFlow` method (mean rank: **1.8**) exhibits uniformly strong performance, obtaining either the best or second best MMD on 11 of the 13 datasets. This is particularly pronounced on the real-world datasets, where we obtain the lowest MMD on 5 of the 7 datasets. We see that IFTPP (mean rank: **2.1**) is a strong baseline, obtaining results which are competitive with our method. The Add-and-Thin method (mean rank: **2.4**) is often similarly strong, but struggles on the SC dataset. While the NHP (mean rank: **3.7**) can obtain good fits, this appears to be dataset dependent, with weak results on the NSR, SC, and Reddit-S datasets. The diffusion baseline (mean rank: **4.8**) is our weakest baseline, which is perhaps unsurprising as this model can only be trained to maximize the likelihood of a subsequent event and not the overall sequence likelihood.

6.5 Conclusion

In this chapter, we propose `EventFlow`, a non-autoregressive and likelihood-free generative model for temporal point processes. We demonstrate that `EventFlow` is able to achieve state-of-the-art results on a multi-step forecasting task and strong performance on unconditional generation.

There are several directions in which our work could be extended. First, we do not explicitly enforce that the support of our model TPP is $[0, T]$. This would necessitate moving beyond the Gaussian setup, which is non-trivial to carry out. Second, more sophisticated approaches to learning the event count distribution $p_\phi(n \mid \mathcal{H})$ could lead to improved performance.

Chapter 7

Conclusions and Outlook

This dissertation has explored several techniques for generative modeling in infinite-dimensional spaces. We began by developing a discrete-time diffusion model for function-valued data in Chapter 3, followed by a continuous-time flow-based model in Chapter 4. In Chapter 5, we developed a theory of dynamic conditional optimal transport, and leveraged this theory to build flow-based models for amortized likelihood-free inference. Finally, we studied an application of flow-based models for temporal point processes in Chapter 6.

As our ability to collect data from the world around us grows, so too will our need for building faithful models of this data. One principle that has guided this dissertation is that in order to do so, we must build models upon foundations that best support the task at hand, rather than on pervasive assumptions taken for granted. While this unavoidably introduces new challenges, solving these leads to new insights, techniques, and connections.

7.1 Open Problems

While we have explored several techniques for function-space generative modeling, this area of machine learning is still nascent and many important questions remain open. Here, we conclude with a brief discussion of a few key directions.

Beyond the Hilbert Setting Throughout this dissertation, we have made the fundamental assumption that our data distribution is supported on a Hilbert space. The defining feature of these spaces is that they are equipped with an inner product, and this inner product is invaluable for performing calculations. However, Hilbert spaces represent only a small subset of the zoo of function spaces, and many spaces of interest are not Hilbertian. For instance, even the conceptually appealing space of continuous functions (equipped with the usual L^∞ norm) does not fit into this framework. A natural open challenge is to generalize the techniques we have insofar developed to a more general class of spaces, such as Banach spaces.

The Role of the Cameron-Martin Space In Chapters 3 and 4, we noted the Cameron-Martin space of the Gaussian noise used to define our path of measures plays a key role in our methods. For instance, recall that we showed that a sufficient condition for our functional flow matching setup in Chapter 4 to be well-defined is that the data measure is supported in this subspace. However, this is a fairly restrictive condition, and several key questions remain regarding the role of this space. For instance, is this sufficient condition *necessary*? If so, can we develop methods for selecting the corresponding Cameron-Martin space in a principled fashion? Or, to sidestep this issue entirely, how can we design methods that do not require this assumption?

Constraints In many problems of practical interest, practitioners may want to impose various constraints or otherwise include prior knowledge into the generated functions. For instance, we may want to only generate functions having some special property, such as monotonicity or having some pre-specified integral. While this can be encouraged by choosing an appropriate space of functions, strictly enforcing such constraints largely relies on designing an appropriate neural architecture. To date, this design space is under explored.

Convergence Analyses Recent theoretical works calculate, under suitable assumptions, upper bounds between the true data distribution and the distribution learned by diffusion [Bortoli, 2022, Oko et al., 2023] or flow-based [Fukumizu et al., 2024] models. However, these analyses are all in the finite-dimensional setting, and moreover, the error bounds depend explicitly on the dimension of the state space. Analyzing the correctness of diffusion and flow-based models in infinite-dimensional spaces is an open and interesting challenge for future work.

Overall, this list of open problems represents only a small sample of the exciting and under-explored field of machine learning in function spaces. Many challenges, both methodological and theoretical, remain open, and addressing these may enhance both our understanding and the real-world applicability of these models.

Bibliography

Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J Fonnesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C Luhmann, Osvaldo A Martin, et al. PyMC: A modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9:e1516, 2023.

Jonas Adler and Ozan Öktem. Deep Bayesian inversion. *arXiv preprint arXiv:1811.05910*, 2018.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Michael Samuel Albergo, Mark Goldstein, Nicholas Matthew Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 921–937. PMLR, 21–27 Jul 2024.

Sergio Albeverio, Yu G Kondratiev, and Michael Röckner. Analysis and geometry on configuration spaces. *Journal of Functional Analysis*, 154(2):444–500, 1998.

Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2022.

Jason Alfonso, Ricardo Baptista, Anupam Bhakta, Noam Gal, Alfin Hou, Isa Lyubimova, Daniel Pocklington, Josef Sajonz, Giulio Trigila, and Ryan Tsai. A generative flow for conditional sampling via optimal transport. *arXiv preprint arXiv:2307.04102*, 2023.

Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Springer Science & Business Media, 2005.

Luigi Ambrosio, Alberto Bressan, Dirk Helbing, Axel Klar, Enrique Zuazua, Luigi Ambrosio, and Nicola Gigli. A users guide to optimal transport. *Modelling and Optimisation of Flows on Networks*, pages 1–155, 2013.

Brandon Amos. Tutorial on amortized optimization. *Foundations and Trends in Machine Learning*, 16(5):592–732, 2023.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, 06–11 Aug 2017.

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? Some theory and empirics. In *International Conference on Learning Representations*, 2018.

Sheldon Axler. *Measure, Integration and Real Analysis*. Springer Nature, 2020.

Ricardo Baptista, Bamdad Hosseini, Nikola B Kovachki, and Youssef Marzouk. Conditional sampling with monotone GANs: From generative models to likelihood-free inference. *arXiv preprint arXiv:2006.06755*, 2020.

Ricardo Baptista, Youssef Marzouk, and Olivier Zahm. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, pages 1–46, 2023.

Raphaël Barboni, Gabriel Peyré, and François-Xavier Vialard. Understanding the training of infinitely deep and wide ResNets with conditional optimal transport. *arXiv preprint arXiv:2403.12887*, 2024.

Mark A Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010.

Heli Ben-Hamu, Samuel Cohen, Joey Bose, Brandon Amos, Maximillian Nickel, Aditya Grover, Ricky TQ Chen, and Yaron Lipman. Matching normalizing flows and probability paths on manifolds. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1749–1763. PMLR, 17–23 Jul 2022.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with process diffusions. In *Workshop on Score-Based Methods*, 2022.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Vladimir I Bogachev, Nicolai V Krylov, Michael Röckner, and Stanislav V Shaposhnikov. *Fokker–Planck–Kolmogorov Equations*, volume 207. American Mathematical Society, 2022.

Vladimir Igorevich Bogachev. *Gaussian Measures*. American Mathematical Society, 1998.

Vladimir Igorevich Bogachev and Maria Aparecida Soares Ruas. *Measure Theory*, volume 2. Springer, 2007.

Vladimir Igorevich Bogachev and Oleg Georgievich Smolyanov. Analytic properties of infinite-dimensional distributions. *Russian Mathematical Surveys*, 45(3):1, 1990.

Jutta Bolt and Jan Luiten Van Zanden. Maddison style estimates of the evolution of the world economy. A new 2020 update. *Maddison Project Working Paper*, 15, 2020.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Tanguy Bosser and Souhaib Ben Taieb. On the predictive accuracy of neural temporal point process models for continuous-time event data. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Cajo JF Ter Braak. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16:239–249, 2006.

Erik Buhmann, Cedric Ewen, Darius A Faroughy, Tobias Golling, Gregor Kasieczka, Matthew Leigh, Guillaume Quétant, John Andrew Raine, Debajyoti Sengupta, and David Shih. Epic-ly fast particle cloud generation with flow-matching and diffusion. *arXiv preprint arXiv:2310.00049*, 2023.

Charlotte Bunne, Andreas Krause, and Marco Cuturi. Supervised training of conditional Monge maps. *Advances in Neural Information Processing Systems*, 35:6859–6872, 2022.

David R Burt. Spectral methods in Gaussian process approximations. Master's thesis, University of Cambridge, 2018.

David R. Burt, Sebastian W. Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Guillaume Carlier, Victor Chernozhukov, and Alfred Galichon. Vector quantile regression: An optimal transport approach. *The Annals of Statistics*, 44(3):1165 – 1192, 2016.

Jannis Chemseddine, Paul Hagemann, and Christian Wald. Y-Diagonal couplings: Approximating posteriors with conditional Wasserstein distances. *arXiv preprint arXiv:2310.13433*, 2023.

Jannis Chemseddine, Paul Hagemann, Christian Wald, and Gabriele Steidl. Conditional Wasserstein distances with applications in Bayesian OT flow matching. *arXiv preprint arXiv:2403.18705*, 2024.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

Ricky TQ Chen. torchdiffeq, 2018. URL `https://github.com/rtqichen/torchdiffeq`.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Sheung Hun Cheng and Nicholas J Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM Journal on Matrix Analysis and Applications*, 19 (4):1097–1110, 1998.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14347–14356, 2021.

Simon L Cotter, Gareth O. Roberts, Andrew M Stuart, and David White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28(3):424 – 446, 2013.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Giuseppe Da Prato and Jerzy Zabczyk. *Stochastic Equations in Infinite Dimensions*. Cambridge University Press, 2014.

Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2003.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Masoumeh Dashti and Andrew M Stuart. The Bayesian approach to inverse problems. *arXiv preprint arXiv:1302.6989*, 2013.

Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23263–23274, 2023.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *Advances in Neural Information Processing Systems*, volume 35, pages 2406–2422, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021.

John R Dormand and Pete J Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6:19–26, 1980.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *Proceedings of the 39th International Conference on Machine Learning*, pages 5694–5725, 2022a.

Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. Generative models as distributions of functions. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pages 2989–3015, 2022b.

Paul Dupuis and Richard S Ellis. *A Weak Convergence Approach to the Theory of Large Deviations*. John Wiley & Sons, 2011.

Rick Durrett. *Probability: Theory and Examples*, volume 49. Cambridge University Press, 2019.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8990–9012. PMLR, 23–29 Jul 2023.

Nathaniel Eldredge. Analysis and probability on infinite-dimensional spaces. *arXiv preprint arXiv:1607.03591*, 2016.

Lawrence C Evans. *Partial Differential Equations*. Graduate Studies in Mathematics. American Mathematical Society, 2010.

Manuel Febrero-Bande and Manuel Oviedo de la Fuente. Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, 51:1–28, 2012.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Gerald B Folland. *Real Analysis: Modern Techniques and Their Applications*, volume 40. John Wiley & Sons, 1999.

Giulio Franzese, Giulio Corallo, Simone Rossi, Markus Heinonen, Maurizio Filippone, and Pietro Michiardi. Continuous-time functional diffusion processes. *Advances in Neural Information Processing Systems*, 36, 2024.

Kenji Fukumizu, Taiji Suzuki, Noboru Isobe, Kazusato Oko, and Masanori Koyama. Flow matching achieves minimax optimal convergence. *arXiv preprint arXiv:2405.20879*, 2024.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, Ali Eslami, and Yee Whye Teh. Neural processes. In *Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

Timothy D Gebhard, Jonas Wildberger, Maximilian Dax, Daniel Angerhausen, Sascha P Quanz, and Bernhard Schölkopf. Inferring atmospheric properties of exoplanets with flow matching and neural importance sampling. *arXiv preprint arXiv:2312.08295*, 2023.

Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*. Cambridge University Press, 2017.

Nicola Gigli. *On the Geometry of the Space of Probability Measures in Rn Endowed with the Quadratic Optimal Transport Distance*. PhD thesis, Scuola Normale Superiore, 2008.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773, 2012.

Gerd Grubb. *Distributions and Operators*. Springer Science & Business Media, 2008.

Paul Hagemann, Lars Ruthotto, Gabriele Steidl, and Nicole Tianjiao Yang. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation. *arXiv preprint arXiv:2303.04772*, 2023.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Sergio Hernandez, Pedro Alvarez, Javier Fabra, and Joaquin Ezpeleta. Analysis of users behavior in structured e-commerce websites. *IEEE Access*, 5:11941–11958, 2017.

Hans Hersbach. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570, 2000.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Nicholas J Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646, 2022.

Bamdad Hosseini, Alexander W Hsu, and Amirhossein Taghvaei. Conditional optimal transport on function spaces. *arXiv preprint arXiv:2311.05672*, 2023.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 2750–2761, 2022.

Robert Inklaar, Herman de Jong, Jutta Bolt, and Jan Luiten Van Zanden. Rebasing 'Maddison': New income comparisons and the shape of long-run economic development. *GGDC Research Memorandum*, 174, 2018.

International Financial Statistics. Prices, Production, and Labor, Labor Force, 2022. URL `https://data.imf.org/regular.aspx?key=63087884`.

Valerie Isham and Mark Westcott. A self-correcting point process. *Stochastic Processes and Their Applications*, 8(3):335–347, 1979.

Noboru Isobe, Masanori Koyama, Kohei Hayashi, and Kenji Fukumizu. Extended flow matching: a method of conditional generation with generalized continuity equation. *arXiv preprint arXiv:2402.18839*, 2024.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Olav Kallenberg. *Foundations of Modern Probability*, volume 2. Springer, 1997.

Olav Kallenberg. *Random Measures, Theory and Applications*, volume 1. Springer, 2017.

Gopinath Kallianpur. *Stochastic Filtering Theory*. Springer Science & Business Media, 2013.

Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942.

Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion generative models in infinite dimensions. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 9538–9563. PMLR, 2023.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3934–3942, 2024a.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Dynamic conditional optimal transport through simulation-free flows. In *Advances in Neural Information Processing Systems*, 2024b.

Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural SDEs as infinite-dimensional GANs. In *International Conference on Machine Learning*, volume 139, pages 5453–5463, 2021.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019.

Young-geun Kim, Kyungbok Lee, and Myunghee Cho Paik. Conditional Wasserstein generator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 7208–7219, 2022.

Young-geun Kim, Kyungbok Lee, Youngwon Choi, Joong-Ho Won, and Myunghee Cho Paik. Wasserstein geodesic generator for conditional distributions. *arXiv preprint arXiv:2308.10145*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 43:3964–3979, 2020.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

Alexander Kukush. *Gaussian Measures in Hilbert Space: Construction and Properties*. John Wiley & Sons, 2020.

Peter D Lax. *Functional Analysis*. John Wiley & Sons, 2014.

Olivier Le Maître and Omar M Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Springer Science & Business Media, 2010.

Peter AW Lewis and Gerald S Shedler. Simulation of nonhomogeneous Poisson processes with degree-two exponential polynomial rate function. *Operations Research*, 27(5):1026–1040, 1979.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. In *Advances in Neural Information Processing Systems*, volume 35, pages 16768–16781, 2022.

Jae Hyun Lim, Nikola B. Kovachki, Ricardo Baptista, Christoper Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, Christopher Pal, Arash Vahdat, and Anima Anandkumar. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023a.

Jen Ning Lim, Sebastian Vollmer, Lorenz Wolf, and Andrew Duncan. Energy-based models for functional data using path measure tilting. In *International Conference on Artificial Intelligence and Statistics*, pages 1904–1923, 2023b.

Sungbin Lim, Eun Bi Yoon, Taehyun Byun, Taewon Kang, Seungwoo Kim, Kyungjae Lee, and Sungjoon Choi. Score-based generative modeling through stochastic evolution equations in hilbert spaces. In *Advances in Neural Information Processing Systems*, volume 36, pages 37799–37812, 2023c.

Haitao Lin, Cheng Tan, Lirong Wu, Zhangyang Gao, Stan Li, et al. An empirical study: Extensive deep temporal point process. *arXiv preprint arXiv:2110.09823*, 2021.

Haitao Lin, Lirong Wu, Guojiang Zhao, Pai Liu, and Stan Z Li. Exploring generative neural temporal point process. *arXiv preprint arXiv:2208.01874*, 2022.

Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time series applications: A survey. *arXiv preprint arXiv:2305.00624*, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

David Lüdke, Marin Biloš, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and thin: Diffusion for temporal point processes. *Advances in Neural Information Processing Systems*, 36:56784–56801, 2023.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

Emile Mathieu, Vincent Dutordoir, Michael Hutchinson, Valentin De Bortoli, Yee Whye Teh, and Richard Turner. Geometric neural diffusion processes. In *Advances in Neural Information Processing Systems*, volume 36, pages 53475–53507, 2023.

Alexander G de G Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 231–239, 2016.

Robert J McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153–179, 1997.

Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30, 2017.

Hongyuan Mei, Guanghui Qin, and Jason Eisner. Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, pages 4475–4485. PMLR, 2019.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.

Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25858–25889. PMLR, 23–29 Jul 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

Yosihiko Ogata. On Lewis' simulation method for point processes. *IEEE transactions on Information Theory*, 27(1):23–31, 1981.

Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50:379–402, 1998.

Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 373–383, 2019.

Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

Takahiro Omi, naonori ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Donal O'Regan. *Differential Equations in Abstract Spaces*, pages 186–194. Springer Netherlands, 1997.

David A Orlando, Charles Y Lin, Allister Bernard, Jean Y Wang, Joshua ES Socolar, Edwin S Iversen, Alexander J Hartemink, and Steven B Haase. Global control of cell-cycle transcription by coupled cdk and network oscillators. *Nature*, 453:944–947, 2008.

Felix Otto. The geometry of dissipative evolution equations: The porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174, 2001.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics*, pages 837–848. PMLR, 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22:2617–2680, 2021.

Athansios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 2002.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Francesco Pedrotti, Jan Maas, and Marco Mondelli. Improved convergence of score-based diffusion models via prediction-correction. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Jakiw Pidstrigach, Youssef Marzouk, Sebastian Reich, and Sven Wang. Infinite-dimensional diffusion models for function spaces. *arXiv preprint arXiv:2302.10130*, 2023.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28100–28127. PMLR, 23–29 Jul 2023.

Md Ashiqur Rahman, Manuel A Florez, Anima Anandkumar, Zachary E Ross, and Kamyar Azizzadenesheli. Generative adversarial neural operators. *Transactions on Machine Learning Research*, 2022.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022a.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S. Greenberg, Pedro J. Goncalves, and Jakob H. Macke. GATSBI: Generative adversarial training for simulation-based inference. In *International Conference on Learning Representations*, 2022b.

Carlos Ramos-Carreño, Alberto Suárez, José Luis Torrecilla, Miguel Carbajo Berrocal, Pablo Marcos Manchn, Pablo Pérez Manso, Amanda Hernando Bernabé, David Garca Fernández, Yujian Hong, Pedro Martín Rodrguez-Ponga Eyriés, lvaro Sánchez Romero, Elena Petrunina, lvaro Castillo, Diego Serna, and Rafael Hidalgo. GAA-UAM/scikit-fda: Functional data analysis in Python, 2019.

James O. Ramsay and Bernhard W. Silverman. *Functional Data Analysis*. Springer New York, 2008.

Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. Springer, 2006.

Jakob Gulddahl Rasmussen. Temporal point processes: The conditional intensity function. *Lecture Notes, Jan*, 2011.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8857–8868, 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021b.

Deep Ray, Harisankar Ramaswamy, Dhruv V Patel, and Assad A Oberai. The efficacy and generalizability of conditional gans for posterior inference in physics-based inverse problems. *Numerical Algebra, Control and Optimization*, 14(1):160–189, 2024. ISSN 2155-3289. URL `https://www.aimsciences.org/article/id/639862c5b2114e413cb35cd4`.

Yulia Rubanova, Ricky TQ Chen, and David K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

W. Rudin. *Functional Analysis*. Higher Mathematics Series. McGraw-Hill, 1973.

Tim G. J. Rudner, Zonghao Chen, and Yarin Gal. Rethinking function-space variational inference in Bayesian neural networks. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494, 2022.

Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500, 2017.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55 (58-63):94, 2015.

Louis Sharrock, Jack Simons, Song Liu, and Mark Beaumont. Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44565–44602. PMLR, 21–27 Jul 2024.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020a.

Oleksandr Shchur, Nicholas Gao, Marin Biloš, and Stephan Günnemann. Fast and flexible temporal point processes with triangular maps. *Advances in Neural Information Processing Systems*, 33:73–84, 2020b.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Alessio Spantini, Ricardo Baptista, and Youssef Marzouk. Coupling techniques for nonlinear ensemble filtering. *SIAM Review*, 64(4):921–953, 2022.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46495–46513. PMLR, 21–27 Jul 2024.

Eugene Stepanov and Dario Trevisan. Three superposition principles: currents, continuity equations and curves of measures. *Journal of Functional Analysis*, 272(3):1044–1103, 2017.

Andrew M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, pages 24804–24816, 2021.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Ba-Hien Tran, Simone Rossi, Dimitrios Milios, and Maurizio Filippone. All you need is a good functional prior for Bayesian deep learning. *Journal of Machine Learning Research*, 23(74):1–56, 2022.

Giulio Trigila and Esteban G Tabak. Data-driven optimal transport. *Communications on Pure and Applied Mathematics*, 69(4):613–648, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017.

Cédric Villani. *Optimal Transport: Old and New*. Springer Berlin Heidelberg, 2009.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Zheyu Oliver Wang, Ricardo Baptista, Youssef Marzouk, Lars Ruthotto, and Deepanshu Verma. Efficient neural network approaches for conditional optimal transport with applications in Bayesian inference. *arXiv preprint arXiv:2310.16975*, 2023.

Veit David Wild, Robert Hu, and Dino Sejdinovic. Generalized variational inference in function spaces: Gaussian measures meet Bayesian deep learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 3716–3730, 2022.

Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. *Advances in Neural Information Processing Systems*, 36, 2024.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

Lemeng Wu, Dilin Wang, Chengyue Gong, Xingchao Liu, Yunyang Xiong, Rakesh Ranjan, Raghuraman Krishnamoorthi, Vikas Chandra, and Qiang Liu. Fast point cloud generation with straight flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9445–9454, 2023.

George Wynne and Veit Wild. Variational Gaussian processes: A functional analysis view. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 4955–4971, 2022.

Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. *Advances in Neural Information Processing Systems*, 30, 2017a.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017b.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. EasyTPP: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations*, 2024.

Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *arXiv preprint arXiv:2106.10121*, 2021.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022.

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10), 2023. ISSN 1099-4300.

Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Samuel Zaidman. *Functional Analysis and Differential Equations in Abstract Spaces*. Taylor & Francis, 1999.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In Hal Daum III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11183–11193. PMLR, 13–18 Jul 2020.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. *arXiv preprint arXiv:2212.12749*, 2022.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11692–11702, 2020.

Appendix A

Supplementary Material: Chapter 3

This section contains additional details regarding Chapter 3, largely to facilitate reproducibility. In particular, Section A.1 discusses various hyperparameter choices and details needed to reproduce our models. Section A.2 studies the effect of various kernel hyperparameters on the resulting quality of the generated functions. Section A.3 contains pseudocode for our methods, and Section A.4 contains experiments on a few additional datasets and with one additional baseline.

A.1 Model Details

In all of our experiments, our model architecture is the Graph Neural Operator (GNO) of Li et al. [2020]. We use a width of 64, a kernel width of 256, and a depth of 6. Inputs to the GNO are graphs, constructed from discrete functional observations. In particular, for every function we construct a graph where each node corresponds to a single observation of the function. Each node has features corresponding to the observation location (i.e. point in X), function value (i.e. scalar in $\mathbb{R}$), and additionally time step $t \in [1, T]$. Nodes are connected if

the Euclidean distance between their observation locations is smaller than a fixed radius r. We use $r = 0.5$ in all of our experiments, and we additionally scale X to $[0, 1] \subset \mathbb{R}$. Each edge in our graph has features corresponding to the observation locations and function values of the respective nodes. While using $r = 1$ would be ideal in this setting, we find this to be prohibitively expensive in terms of computation and memory usage. The Fourier Neural Operator (FNO) [Li et al., 2021] has a significantly reduced computation and memory cost compared to the GNO, but this model is limited to functional observations which are on a uniform gridding of X.

Our models are all trained for 50 epochs and a learning rate of 0.001. We use $T = 1000$ time steps in all of our experiments. We set $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$, and we linearly interpolate between these two values for other settings of β_t. We parametrize the Gaussian measure in our forward process via a mean-zero Gaussian process with a Matérn kernel of unit variance and lengthscale $\ell = 0.1$. In particular, we use a Matérn kernel with $\nu = 1/2$ (i.e. the exponential kernel) when $H = L^2(X, \omega)$ and $\nu = 3/2$ when $H = H^1(X, \omega)$. This choice was made to ensure that the Gaussian measure was sufficiently rough to remove any information contained in the functional data, yet regular enough to be square-integrable (and differentiable in the $\nu = 3/2$ case) such that we obtain a valid Gaussian measure on H.

A.2 Kernel Ablation

In Tables A.1-A.2, we study the effect of the kernel choice in the forward process on the MoGP and AEMET datasets. In particular, we train models as above (using the discrete $L^2(X, \omega)$ loss function), but choose between values of $\nu = 1/2$ and $\nu = 3/2$ and sweep across various length scales between 0.005 and 0.5. We then sample 500 generated functions from our model, and compute the average pointwise mean and variance curves, as well as the average autocorrelation curve – see Figures (3.1) and (A.1) for a visualization. We report the

MSE between these generated functional statistics and the true functional statistics given by the training data.

We see that choosing a length scale that is either significantly larger or smaller than the length scale of the underlying functional data can have negative effects on the statistics, but for reasonable choices of the length scale, the statistics are comparable. Although $\ell = 0.1$ does not produce the best MSE values on the AEMET datsaet, we still use $\ell = 0.1$ in our main experiments as this produced the most qualitatively realistic generated curves.

Table A.1: Effect of kernel choice on the MoGP dataset. We report the MSE between various functional statistics on the training data and data generated via our model with the listed kernel hyperparameters.

ν	ℓ	Mean	Var.	Autocorr.
1/2	0.005	3.0333	6.1184	1.211e-4
	0.01	0.4474	1.2174	5.552e-06
	0.1	0.0032	0.2328	9.169e-06
	0.2	0.4496	1.2752	9.183e-06
	0.5	0.0318	0.2772	1.080e-05
3/2	0.005	0.5225	0.5783	3.638e-05
	0.01	1.6557	4.7887	5.699e-05
	0.1	0.4645	0.1239	1.928e-05
	0.2	0.1046	0.2300	3.947e-06
	0.5	0.2651	0.2586	6.677e-05

A.3 Pseudocode

In this section we detail pseudocode for model training, unconditional sampling, and conditional sampling. Note that during training, we assume $u_0(\vec{x}) = \vec{y}$, i.e. we treat the observations as if they were noiseless. Thus the likelihood term $q(\vec{y} \mid \vec{x}, u_0)$ does not contribute to the loss, and we need only optimize the terms L_{t-1} (see Equation (3.35). Moreover, as mentioned in the main paper, we set $\lambda_t = 1$ as is standard in diffusion modeling [Ho et al., 2020].

Table A.2: Effect of kernel choice on the AEMET dataset. We report the MSE between various functional statistics on the training data and data generated via our model with the listed kernel hyperparameters. For $\nu = 3/2$ with a length scale of $\ell = 0.5$ our training failed to produce a reasonable model.

ν	ℓ	Mean	Var	Autocorr
1/2	0.005	0.1118	74.8143	1.813-06
	0.01	0.0646	2.2001	4.563e-06
	0.1	0.7284	2.2519	5.805e-05
	0.2	0.0152	1.0748	2.551e-06
	0.5	0.0832	3.0590	1.516e-05
3/2	0.005	0.1393	8.5001	1.700e-05
	0.01	0.0899	1.28130	5.638-06
	0.1	0.9317	148.4542	0.0021
	0.2	7.3748	15634.6889	0.0398
	0.5	-	-	-

Note that the given pseudocode for conditional generation covers both hard and soft conditioning. Hard conditioning is obtained when $n_{\text{free}} = 0$, and soft conditioning is obtained by setting $n_{\text{free}} \geq 1$, i.e. the parameter n_{free} indicates how many generation steps are not conditioned on the given information.

Algorithm 3: Training Step

1 Sample $(\vec{x}, \vec{y})$ from training data;
2 Sample t uniformly from $\{2, \ldots, T\}$;
3 Sample $\xi \sim \mathcal{GP}(0, k)$, evaluated at $\vec{x}$ to obtain $\xi(\vec{x})$;
4 Construct $u_t \mid u_0$, evaluated at $\vec{x}$, via Lemma (15): $u_t(\vec{x}) = \sqrt{\gamma_t} u_0(\vec{x}) + \sqrt{1-\gamma_t}\xi(\vec{x})$;
5 Compute model output $\xi^\theta(\vec{x} \mid u_t, t)$;
6 Take a θ-gradient step on $L_{t-1} = \left(\xi(\vec{x}) - \xi^\theta(\vec{x} \mid u_t, t)\right)^T A \left(\xi(\vec{x}) - \xi^\theta(\vec{x} \mid u_t, t)\right)$, where

$$A = \begin{cases} K_{\vec{x}\vec{x}}^{-1} & H = L^2(X, \omega) \\ \pi_{\text{PSD}}\left([I + D^T D][K_{\vec{x}\vec{x}} + K'_{\vec{x}\vec{x}} D]^{-1}\right) & H = H^1(X, \omega) \end{cases} \tag{A.1}$$

Algorithm 4: Unconditional Sampling

Specify query points $\vec{x} \subset X$;
Sample $u_T \sim \mathcal{GP}(0, k)$, evaluated at $\vec{x}$ to obtain $u_T(\vec{x})$;
for $t = T, T-1, \ldots, 1$ **do**
    Sample $\xi_t \sim \mathcal{GP}(0, k)$, evaluated at $\vec{x}$ to obtain $\xi_t(\vec{x})$;
    $u_{t-1}(\vec{x}) \leftarrow \frac{1}{\sqrt{1-\beta_t}} \left(u_t(\vec{x}) - \frac{\beta_t}{\sqrt{1-\gamma_t}} \xi^\theta(\vec{x} \mid u_t, t) \right) + \sqrt{\widetilde{\beta}_t} \xi_t(\vec{x})$;
Return $u_0(\vec{x})$

Algorithm 5: Conditional Sampling

Given: conditioning information $\mathcal{D} = \{(x_c^{(i)}, y_c^{(i)})\}_{i=1}^{n_c} = \{\vec{x}_c, \vec{y}_c\}$;
Specify query points $\vec{x} = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \subset X$;
Create augmented support $\vec{z} = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}, x_c^{(1)}, \ldots, x_c^{(n_c)}\}$;
Sample $u_T \sim \mathcal{GP}(0, k)$, evaluated at $\vec{z}$ to obtain $u_T(\vec{z})$;
for $t = T, T-1, \ldots, 1$ **do**
    Sample $\xi_t \sim \mathcal{GP}(0, k)$, evaluated at $\vec{z}$ and $\vec{x}_c$ to obtain $\xi_t(\vec{z})$;
    Sample reverse process unconditionally on $\vec{z}$:

$$\widetilde{u}_{t-1}(\vec{z}) \leftarrow \frac{1}{\sqrt{1-\beta_t}} \left(u_t(\vec{z}) - \frac{\beta_t}{\sqrt{1-\gamma_t}} \xi^\theta(\vec{z} \mid u_t, t) \right) + \sqrt{\widetilde{\beta}_t} \xi_t(\vec{z}) \tag{A.2}$$

    if $t > n_{free}$ **then**
        Sample $\xi_t' \sim \mathcal{GP}(0, k)$, and evaluate at $\vec{x}_c$ to obtain $\xi_t'(\vec{x}_c)$;
        Perturb conditioning information via the forward process:

$$\vec{y}_{c,t} = \sqrt{\gamma_t} \vec{y}_c + \sqrt{1-\gamma_t} \xi_t'(\vec{x_c}) \tag{A.3}$$

        For each $x \in \vec{z}$, conditioned on perturbed conditioning information by setting

$$u_{t-1}(x) = \begin{cases} \widetilde{u}_{t-1}(x) & x \notin \mathcal{D} \\ y_{c,t}(x) & x \in \mathcal{D} \end{cases} \tag{A.4}$$

    else
        Do no conditioning: $u_{t-1}(\vec{z}) \leftarrow \widetilde{u}_{t-1}(\vec{z})$;
Return u_0

A.4 Additional Experiments

A.4.1 Unconditional Samples

In Figure (A.1), we provide additional examples of our model on various datasets not discussed in the main paper. The first dataset (*Linear*) is a synthetic dataset consisting of random linear functions $u_0(x) = ax + b$ where $a \sim \mathcal{N}(2, 0.25^2)$ and $b \sim \mathcal{N}(-1, 0.07^2)$. Note that, although the pointwise variance of the generated samples in this dataset appear to be significantly smaller than the that of the true samples, this is largely due to the small scale of the variance. The other datasets (*Growth, Canadian, Octane*) are well-known functional data analysis datasets, which are available in the Python package scikit-fda [Ramos-Carreño et al., 2019].

A.4.2 FPCA Baseline

We additionally include a simple unconditional baseline based on functional principal component analysis (FPCA). In particular, we approximate the first $M = 5$ functional principal components by discretizing the training data (see Ramsay and Silverman [2008, Chapter 6] for details and Ramos-Carreño et al. [2019] for an implementation), followed by fitting a multivariate Gaussian to the resulting scores. To sample from this model, we sample from the Gaussian distribution over scores and project back to function space by taking linear combinations of the principal components with these sampled scores.

See Figure A.2 for an illustration of this approach on all of the datasets we have thus far considered. We see that while the FPCA baseline is able to accurately match the functional statistics of the training data, the generated samples often fail to match the qualitative performance of our FuncDiff model (Figures 3.1 and A.1). Note that, unlike our FuncDiff model, we are unable to perform conditional generation with this FPCA baseline.

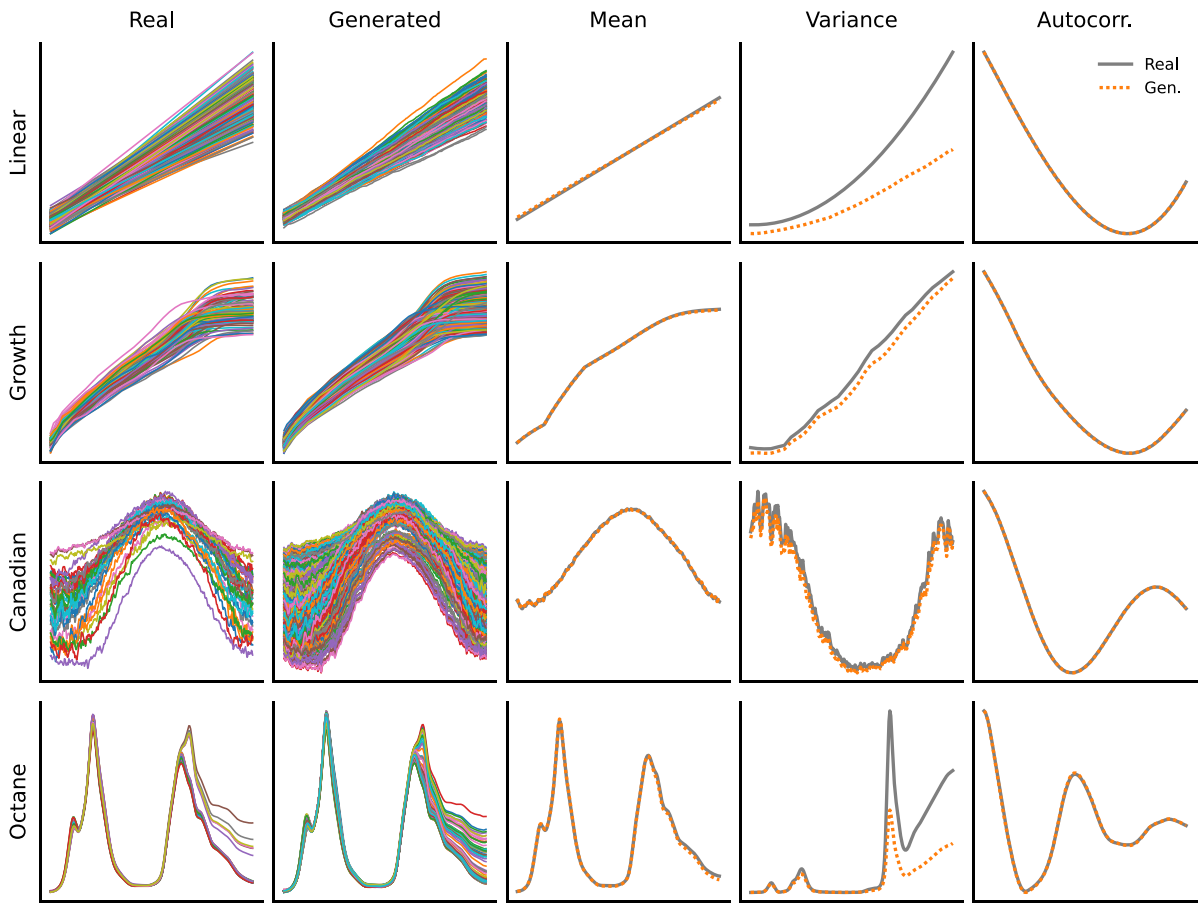


Figure A.1: Unconditional function generation on a synthetic (Linear) and several real-world (Growth, Canadian, Octane) datasets. For each dataset, a GNO model was trained on the plotted functions (first column), and a total of 500 functions were sampled from the model (second column).

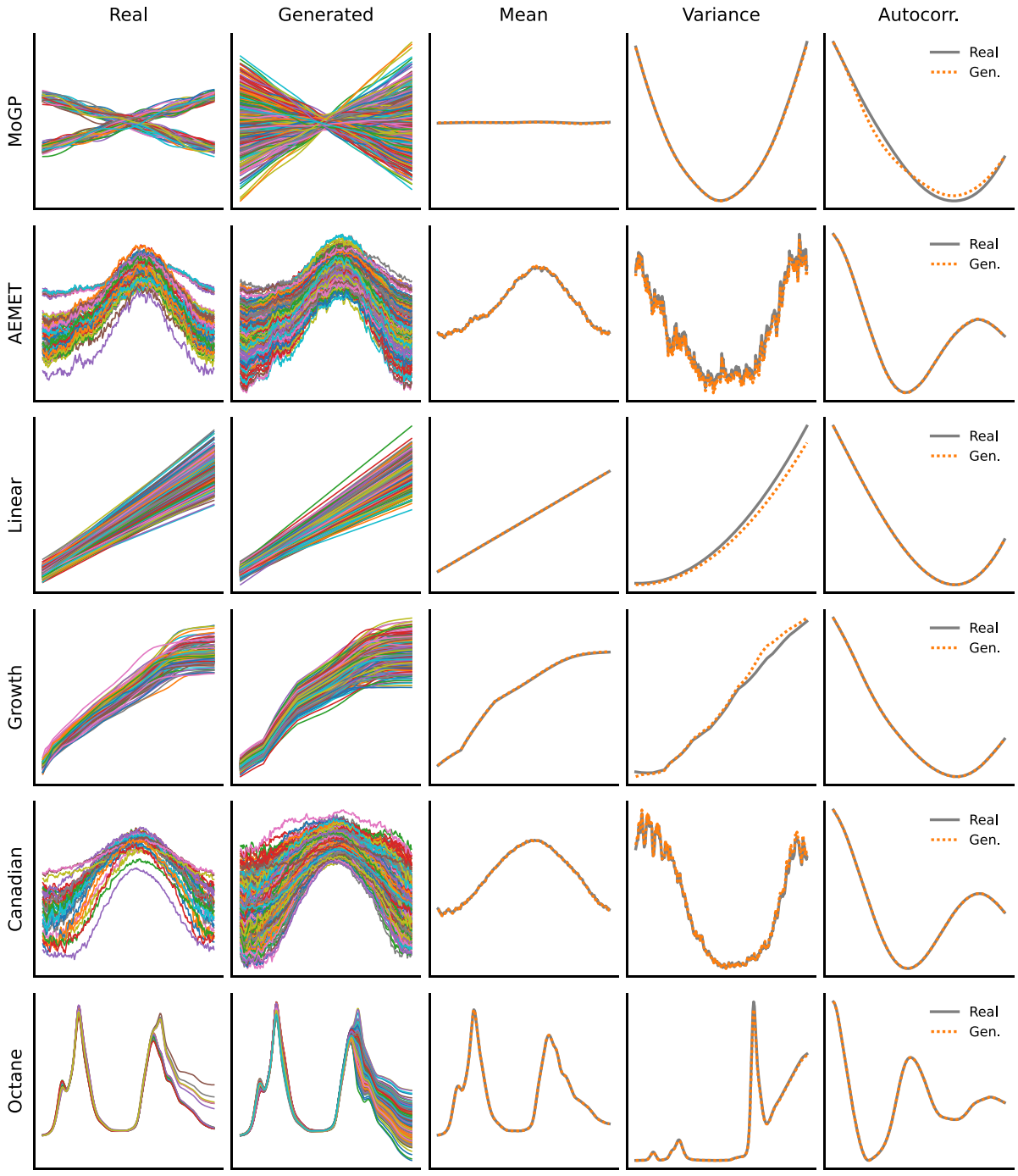


Figure A.2: Unconditional samples from an FPCA-based model on various datasets. For each dataset, we estimate the first $M = 5$ functional principal components and fit a Gaussian distribution to the resulting scores. Generation is performed by sampling from said Gaussian and taking the resulting linear combination of functional principal components. Although the functional statistics closely match those of the training data, the perceptual quality of the generated curves is worse than our FuncDiff model.

Appendix B

Supplementary Material: Chapter 4

This section contains additional supplementary material for Chapter 4. In particular, Section B.1 contains details necessary to reproduce our experiments, e.g., hyperparameters and information regarding datasets. Section B.2 contains additional experimental results, including a zero-shot super-resolution task. Lastly, Section B.3 describes some additional details regarding conditional simulation.

B.1 Experiment Details

B.1.1 Parametrizations

The FM-OT and FM-VP model require specifying a variance schedule via the hyperparameter σ_t^f. In this work, we parametrize FFM-OT by setting $\sigma_{\min} = 1\text{e}{-4}$. For FFM-VP, we set $\alpha_t = \cos\left(\frac{t+s}{1+s}\frac{\pi}{2}\right)$ where $s = 0.08$, following a formulation similar to the cosine schedule introduced by Nichol and Dhariwal [2021].

Model-specific hyperparameters have been extensively fine-tuned via grid search, and we

found the following parametrizations to consistently perform optimally across several domains:

- DDPM: the noise schedule, following the notation of Kerrigan et al. [2023], is set to linearly interpolate between $\beta_0 = 1e-4$ and $\beta_T = 0.02$ in $T = 1000$ timesteps. The code for this implementation was taken directly from the official repository[1].
- DDO: following the notation of Lim et al. [2023a], we set the time interval to $T = 10$, and the noise schedule geometrically interpolates between $\sigma_{10} = 1e-3$ and $\sigma_1 = 1$ on the 1D datasets and $\sigma_{10} = 1e-2$ and $\sigma_1 = 100$ on the 2D datasets. Sampling is performed by running their annealed Langevin dynamics algorithm with $\epsilon = 2 \times 10^{-5}$ and $M = 200$.
- GANO: the generator is trained every 5 epochs, and gradient penalty set to $\lambda = 0.1$ in 1D and $\lambda = 10$ in 2D. [Rahman et al., 2022]. The code for this implementation was taken directly from the official repository[2].

Model Architectures For FFM, DDPM, and DDO, the architecture used is the FNO implemented in the `neuraloperator` package [Li et al., 2021, Kovachki et al., 2021]. For GANO, we directly use the FNO-based model architectures for both the discriminator and generator implemented by Rahman et al. [2022] for the 2D dataset, while for the 1D datasets we use the same FNO architecture as the other methods.

Gaussian Measures Each model experimented with relies on noise sampled from a Gaussian measure. In our work, we consider a mean-zero Gaussian process (GP) parametrized by a Matrn kernel with $\nu = 1/2$. In 1D, the kernel hyperparameters are set to have a variance $\sigma^2 = 0.1$ and length scale $\ell = 1e-2$. In 2D, the variance is $\sigma^2 = 1$ and the length scale was set to $\ell = 1e-3$ for DDO and GANO, and $\ell = 1e-2$ for FFM and DDPM.

[1] https://github.com/GavinKerrigan/functional_diffusion
[2] https://github.com/neuraloperator/GANO

Training All models are trained using the Adam optimizer. In 1D, we use an initial learning rate of 1e−3, scheduled to decrease by one order of magnitude after 50 epochs for all datasets but AEMET, where it is decreased every 25 epochs. In 2D, we use an initial learning rate of $5e-4$ for FFM, DDPM, and DDO, and an initial learning rate of $1e-4$ for GANO, and this initial learning rate was decayed by one order of magnitude every 25 epochs.

B.1.2 Dataset Details

Navier-Stokes. This dataset consists of solutions to the Navier-Stokes equations on a 2D torus at a resolution of 64x64. For the sake of efficient training, we randomly selected 20,000 datapoints for training from the original dataset [Li et al., 2022] as there is a high degree of redundancy in the data. For FFM, DDPM, and DNO, we use 4 Fourier layers of 32 modes and 64 hidden channels, 256 lifting channels, 256 projection channels, and the GeLU activation function [Hendrycks and Gimpel, 2016]. For GANO, we use 32 modes and set the number of hidden channels to 16 due to memory constraints. All models were trained for 300 epochs at a batch size of 128.

AEMET dataset. This dataset consists of a set of functions describing the mean curve of the average daily temperature (in Celsius) for the period 1980-2009 recorded by 73 weather stations in Spain [Febrero-Bande and de la Fuente, 2012]. Each function is observed on a uniform grid at a resolution of 365. The neural architecture we use for this dataset is an FNO with a width of 256 and 64 modes, kept constant for all models considered in this experiment. The model is trained for 50 epochs, with batch size set to 73.

Gene expression. The original dataset consists of 10,928 time series at 20 uniformly spaced time points, recording the amplitude of gene expression for 4 different genes. The genes are concatenated to create the visual effect of spikes occurring periodically in time,

while maintaining the structure of the original dataset. The data was log-transformed and centered before being fed to the model. We restrict our focus to a subset of 156 functions exhibiting large gene expression, determined by the standard deviation averaged across time for each centered function being greater than 0.3. For this dataset, we use an FNO with a width of 256 and 16 modes across all models. The model is trained for 200 epochs, with batch size set to 16.

Economic datasets. The first two datasets are taken from the Maddison Project database [Bolt and Van Zanden, 2020, Inklaar et al., 2018], and the third from the IMF [International Financial Statistics, 2022]. The datasets were picked specifically for their distinct visual characteristics, explored in greater detail in Appendix B.2.

- *Population*: time series of the evolution of the population for 169 countries across the globe from the year 1950 to 2018 (that is, discretized at 69 points in time). For a clearer visual representation, each time series was divided by its mean, so each curve represents the population for each country, relative to the mean population for that country over the 69 years under consideration. The functions in this dataset exhibit linear growth over time, with a change point shared across observations.

- *GDP*: time series representing the evolution of GDP per capita from 1950 to 2018. The original dataset consists of 169 countries, but time series presenting missing values were removed yielding 145 observations. The same preprocessing as that described above was applied to the data. While the functions seem to exhibit the same change point as that observed for the population datasets, the growth over time is noisier and exhibits irregular patterns.

- *Labor*: size of labor force per quarter between Q1 2017 and Q4 2022 (for a total of 24 points in time), for a subset of 35 countries (obtained removing those with missing

values from the original 105 observations). The same preprocessing as that described for the population dataset was applied to the data. This dataset tests the ability of the generative models to learn from small and multimodal data.

The models for the population and GDP datasets have width set to 256 and 32 modes, while the one for the labor dataset has width set to 128 and 8 modes. All models were trained for 100 epochs, with a batch size of 16.

B.1.3 Sampling Details

We use the `torchdiffeq` [Chen, 2018] package for all ODE solvers. The specific solver we use is `dopri5`, an implementation of the Dormand-Prince method [Dormand and Prince, 1980] of order 5. We set the absolute and relative tolerance parameters to 1e−10 for the 1D datasets and 1e−5 for the 2D datasets. Note that setting such a tolerance gives us an explicit way of trading off sample quality for sampling efficiency.

B.2 Additional Experimental Results

This section contains several additional experimental results and figures. First, in Section B.2.1 we explore an additional synthetic dataset consisting of a mixture of Gaussian processes (MoGP). Next, we provide additional results on super-resolution in Section B.2.2. Sections B.2.3 and B.2.4 provide further visualizations of our results on the 1D and 2D datasets considered in the main paper respectively.

B.2.1 Additional Results: MoGP

Mixture of Gaussian Processes (MoGP) Dataset. The experiment considers the task of generating samples from a mixture of two GPs. The two components, with equal weights, have mean functions $m_1 = 10x - 5$ and $m_2 = -10x + 5$, and a squared-exponential kernel with variance $\sigma^2 = 0.04$ and length scale $\ell = 0.1$. The synthetic samples used for training are observed on a uniform grid at a resolution of 64 on the interval $[0, 1]$. All models were trained on the same sample of $N = 5000$ realizations of the mixture of GPs. Figure B.1 illustrates a visual comparison of 500 samples from each model, while Table B.1 presents a quantitative comparison of the mean squared error (MSE) on various pointwise statistics. Additionally, Figure B.1 provides a comprehensive depiction of the variations in these pointwise statistics across different models.

Table B.1: Average MSEs between true and generated samples for various pointwise statistics on the MoGP dataset, along with the standard deviation (across ten random seeds). The average number of function evaluations (NFEs) for each model is also reported. Variants of our proposed FFM model obtain the best or second best average performance across all metrics. DDPM outperforms FFM in terms of variance, but only by a small margin.

	Mean	Variance	Skewness	Kurtosis	Autocorrelation	NFEs
FFM-OT (ours)	**2.2e-2** (3.e-2)	2.9e-1 (3.2e-1)	1.6e-2 (1.1e-2)	**1.1e-2** (1.2e-2)	**7.e-6** (6.e-6)	740
FFM-VP (ours)	3.9e-2 (3.6e-2)	3.6e-1 (5.6e-1)	**1.4e-2** (5.2e-3)	1.5e-2 (1.2e-2)	8.e-6 (8.e-6)	716
DDPM	3.0e-2 (2.4e-2)	**1.4e-1** (1.9e-1)	1.5e-2 (9.6e-3)	1.2e-2 (8.1e-3)	1.9e-5 (2.2e-5)	1000
DDO	7.3e-1 (9.6e-1)	2.7e+0 (5.3e+0)	4.2e-1 (8.7e-1)	2.8e-1 (3.9e-1)	1.3e-5 (8.e-6)	2000
GANO	1.9e-1 (1.6e-1)	8.1e+0 (6.0e+0)	3.4e-1 (2.5e-1)	4.6e-2 (3.9e-2)	6.2e-4 (6.7e-4)	1

B.2.2 Additional Results: Super-Resolution

Here, we provide additional visualizations regarding the ability of all models considered to perform super-resolution, i.e. to sample at a resolution greater than the training dataset. All models considered are trained at the original dataset resolution, but due to the neural operator architectures being used, we may sample at arbitrary resolutions. We note that quantitatively evaluating these super-resolved samples is difficult as we do not have access to a notion of higher-resolution ground truth here.

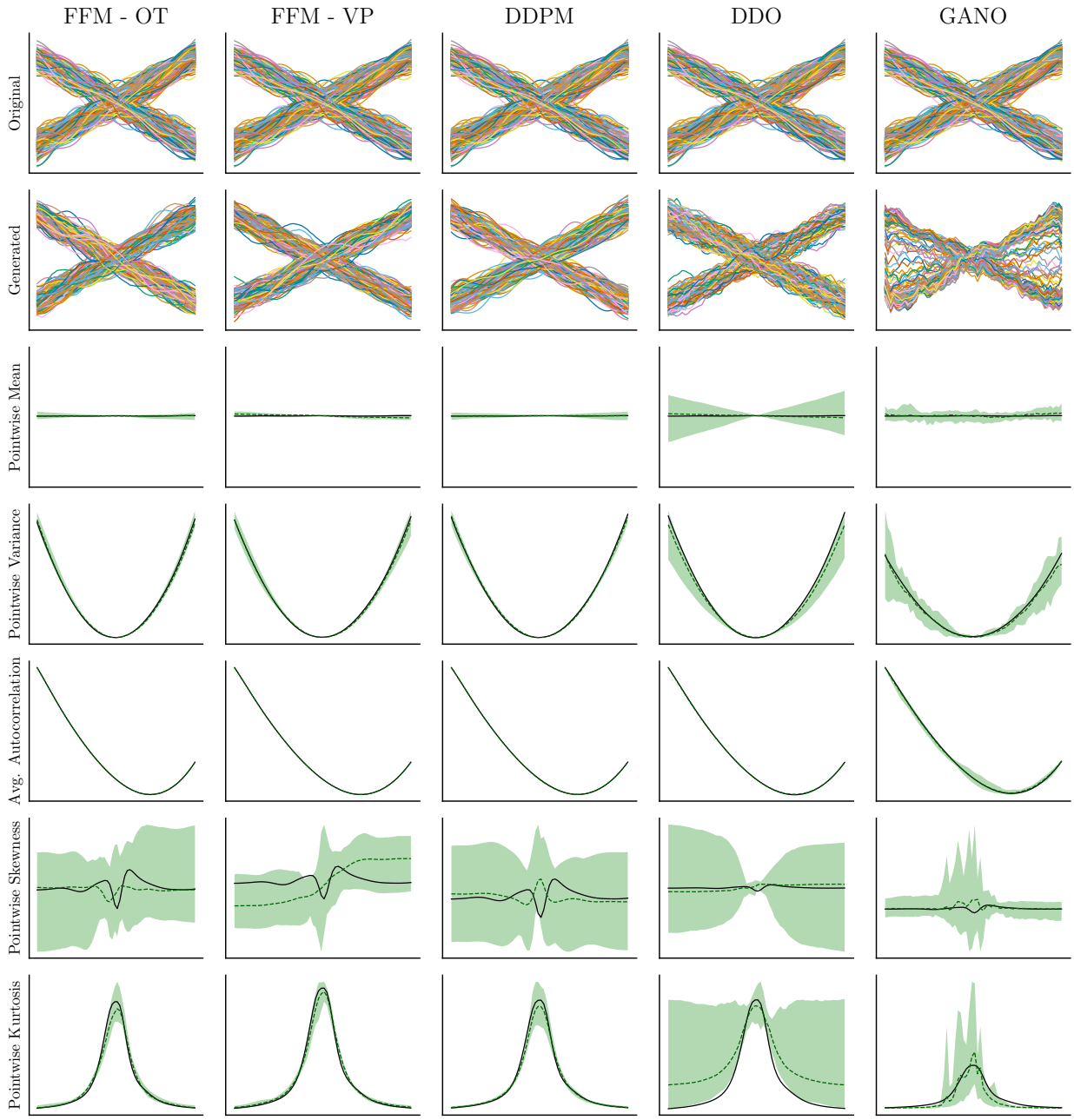


Figure B.1: Various pointwise statistics on the MoGP dataset. Curves in black indicate the corresponding pointwise statistic for the original dataset (top row). The green error bands represent the minimal and maximal value of the pointwise statistic from 500 samples across ten random seeds of the corresponding model. The dashed green lines indicate the mean pointwise statistic across these ten runs for each model. See Table B.1 for a quantitative comparison.

Figure B.2 shows samples from the original Gene expression time series dataset [Orlando et al., 2008] at a resolution of 20, as well as samples from each model at a 5x resolution,

i.e. a resolution of 100. We see that, qualitatively, samples from FFM, DDPM, and GANO resemble those of the original dataset. The samples from DDO appear overly rough, and the samples from GANO are smoother than those from FFM and DDPM.

Figure B.3 shows qualitatively similar results on the econometrics datasets [Bolt and Van Zanden, 2020, Inklaar et al., 2018, International Financial Statistics, 2022], with the exception of FFM-VP generating samples that are rougher than the original data. To further explore the quality of these super-resolved samples, we additionally provide the correlation matrices of the original data and super-resolved samples in Figure B.4. We generally see that FFM-OT, FFM-VP, DDPM, and GANO are able to qualitatively capture the original correlation structures, whereas DDO fails to do so. We additionally note that on the Population and GPD datasets, FFM-VP, DDO, and GANO display a consistent strong diagonal band, indicating that these models generate samples at a variance which is too large when super-resolved. All models display this failure mode on the Labor dataset.

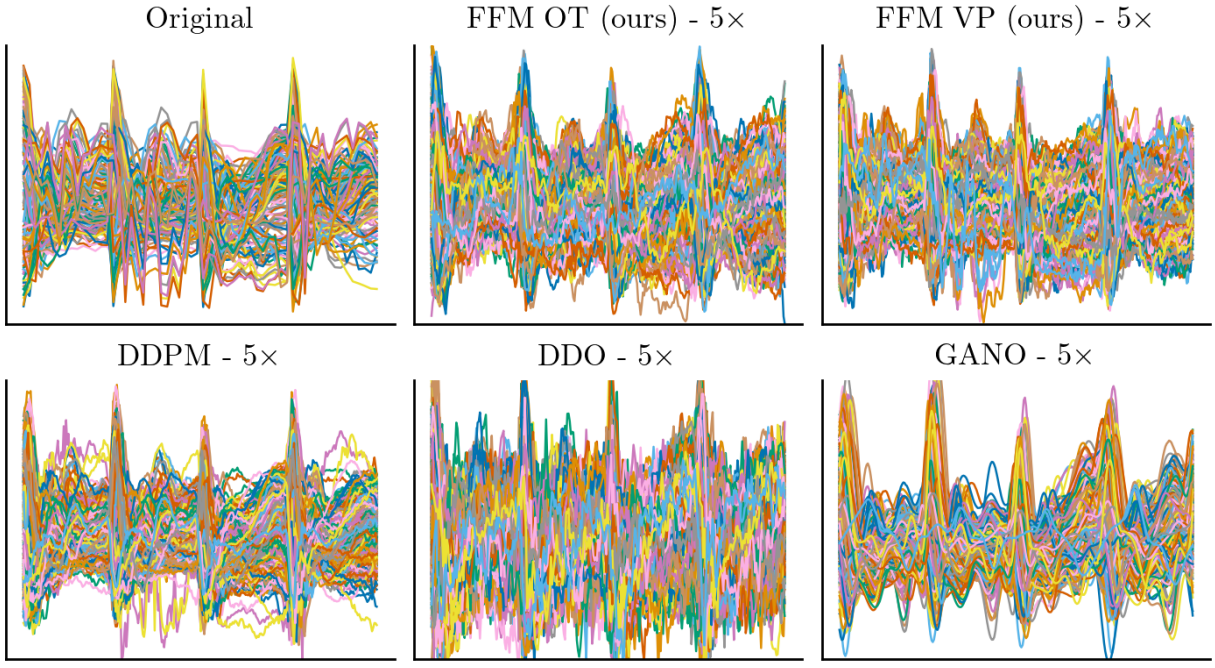


Figure B.2: Samples from the gene expression dataset and samples from the various models at a 5x super-resolution.

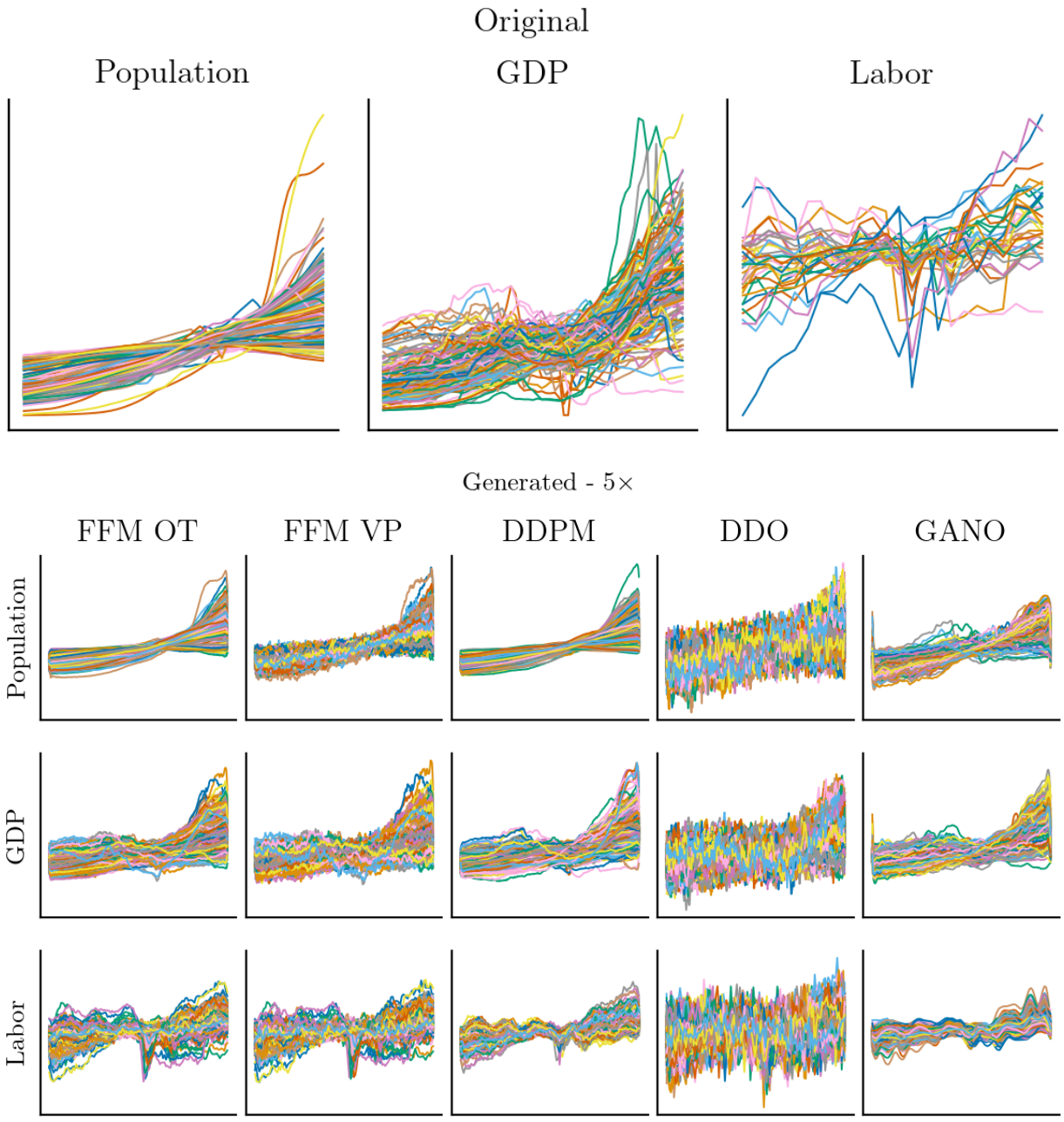


Figure B.3: Samples from the three economics datasets and samples from the various models at a 5x super-resolution.

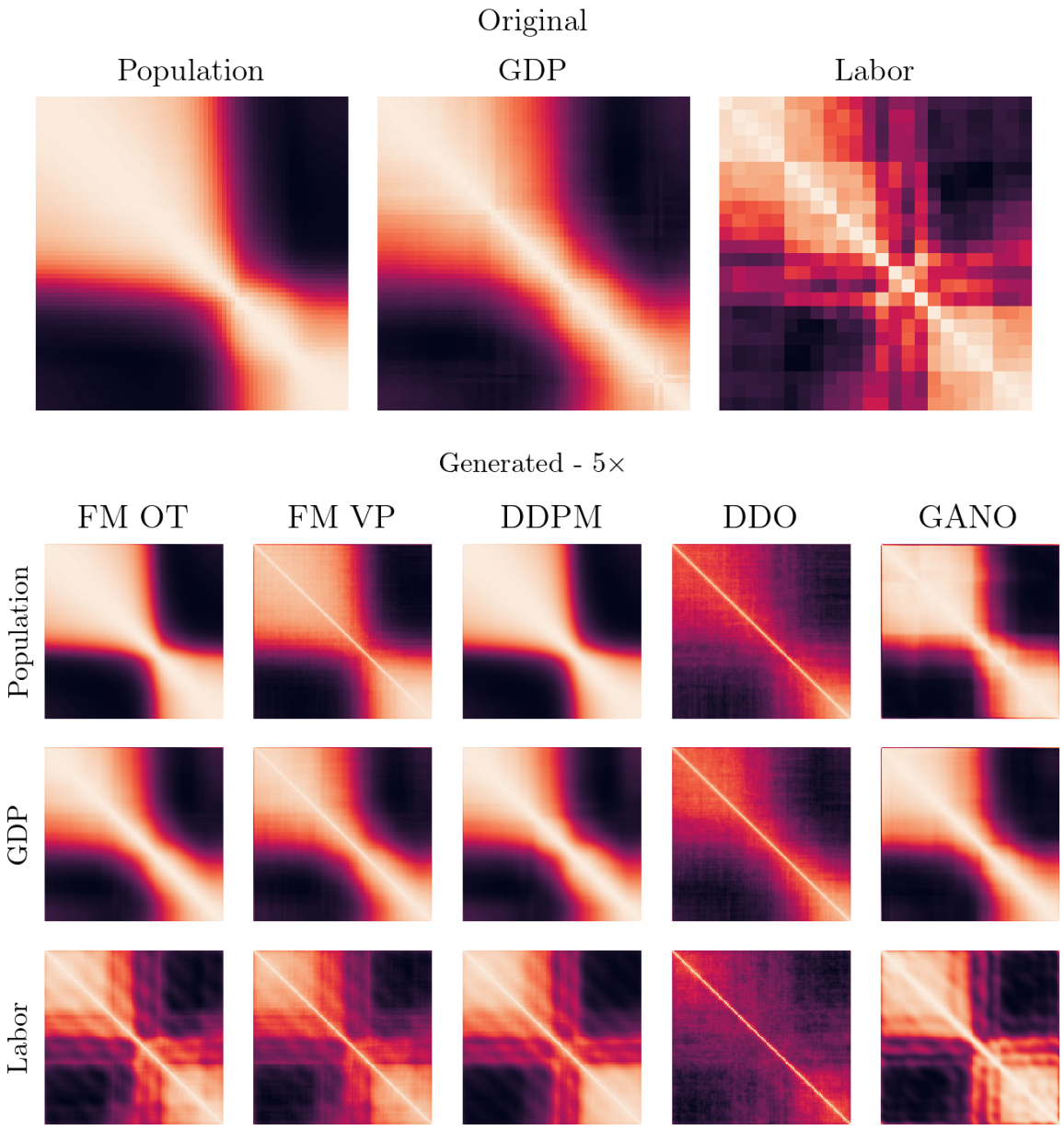


Figure B.4: Correlation matrices for the three economics datasets (top row), as well as correlation matrices for each dataset for the generated samples from each model at a 5x super-resolution.

B.2.3 Additional Results: 1D Datasets

This section provides a further analysis and visualization of the generated samples for the 1D datasets considered in the main paper. In Figures B.5 through B.9, we plot the original data, generated samples from each model, and various pointwise statistics for the real and generated samples. Curves in black indicate the corresponding pointwise statistic for the original dataset (top row). The green error bands represent the minimal and maximal value of the pointwise statistic from 500 samples across ten random seeds of the corresponding model. The dashed green lines indicate the mean pointwise statistic across these ten runs for each model. See Table 4.1 for a quantitative comparison derived from these figures.

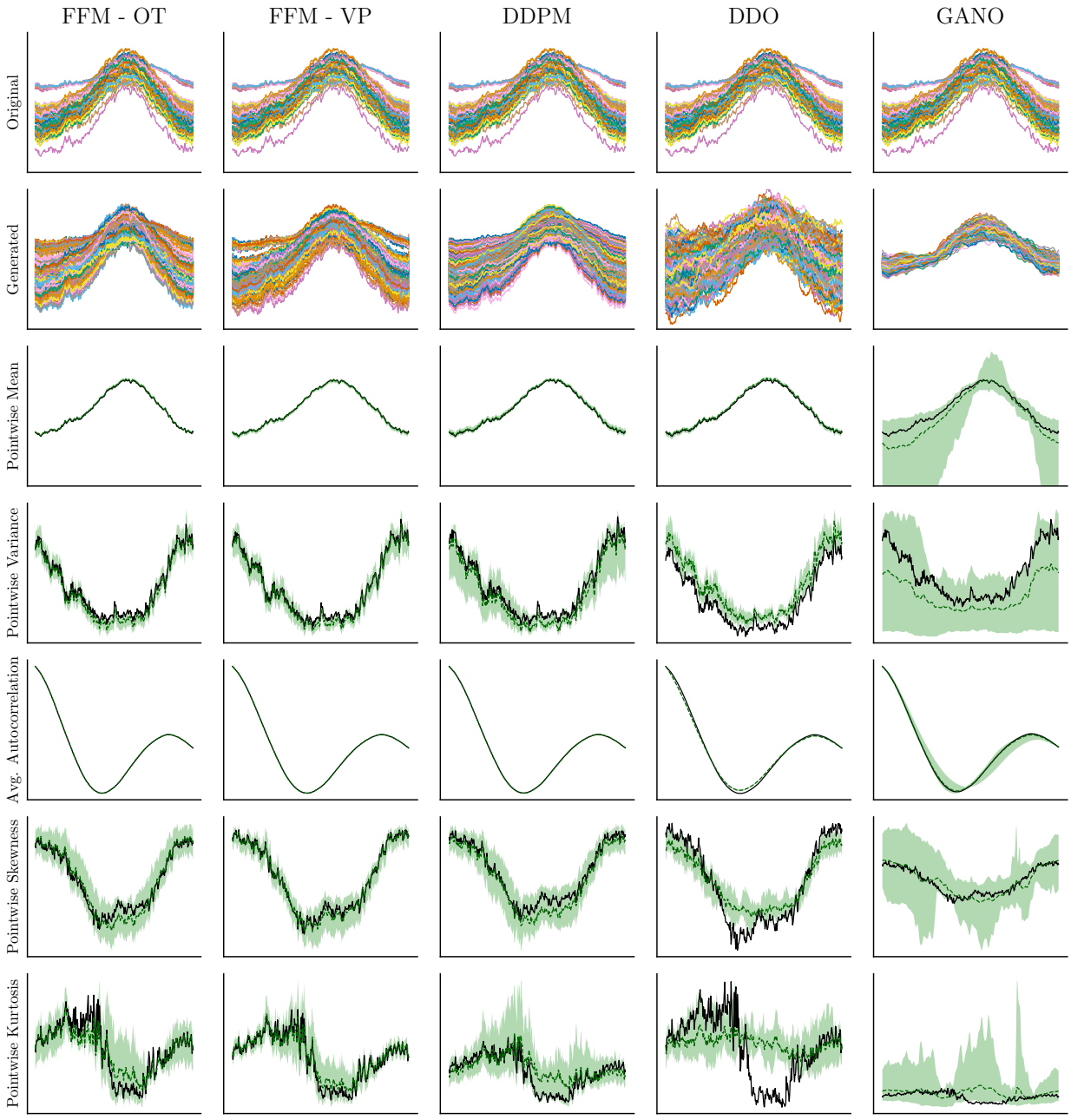


Figure B.5: Various pointwise statistics on the AEMET dataset.

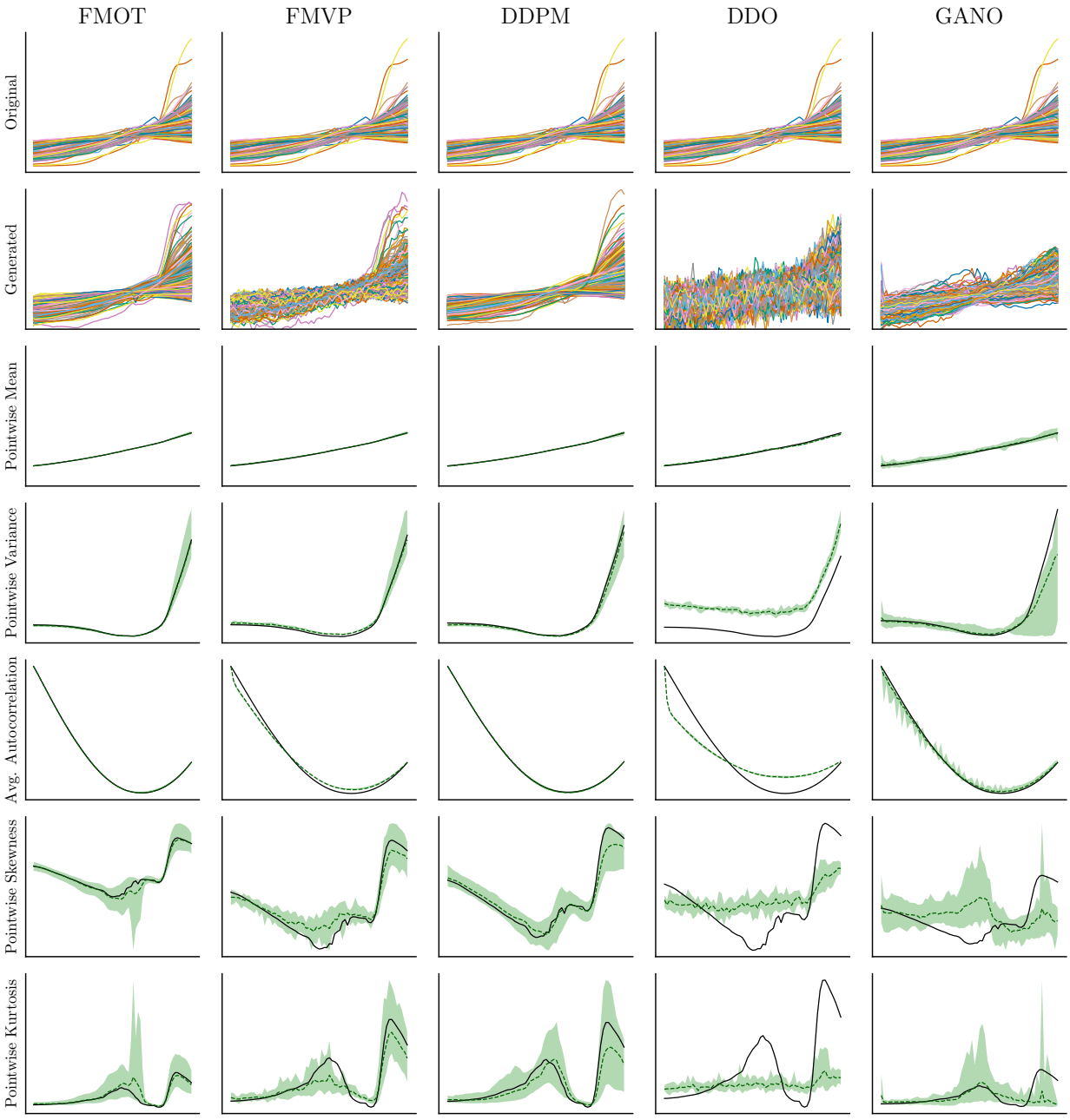


Figure B.6: Various pointwise statistics on the Population dataset.

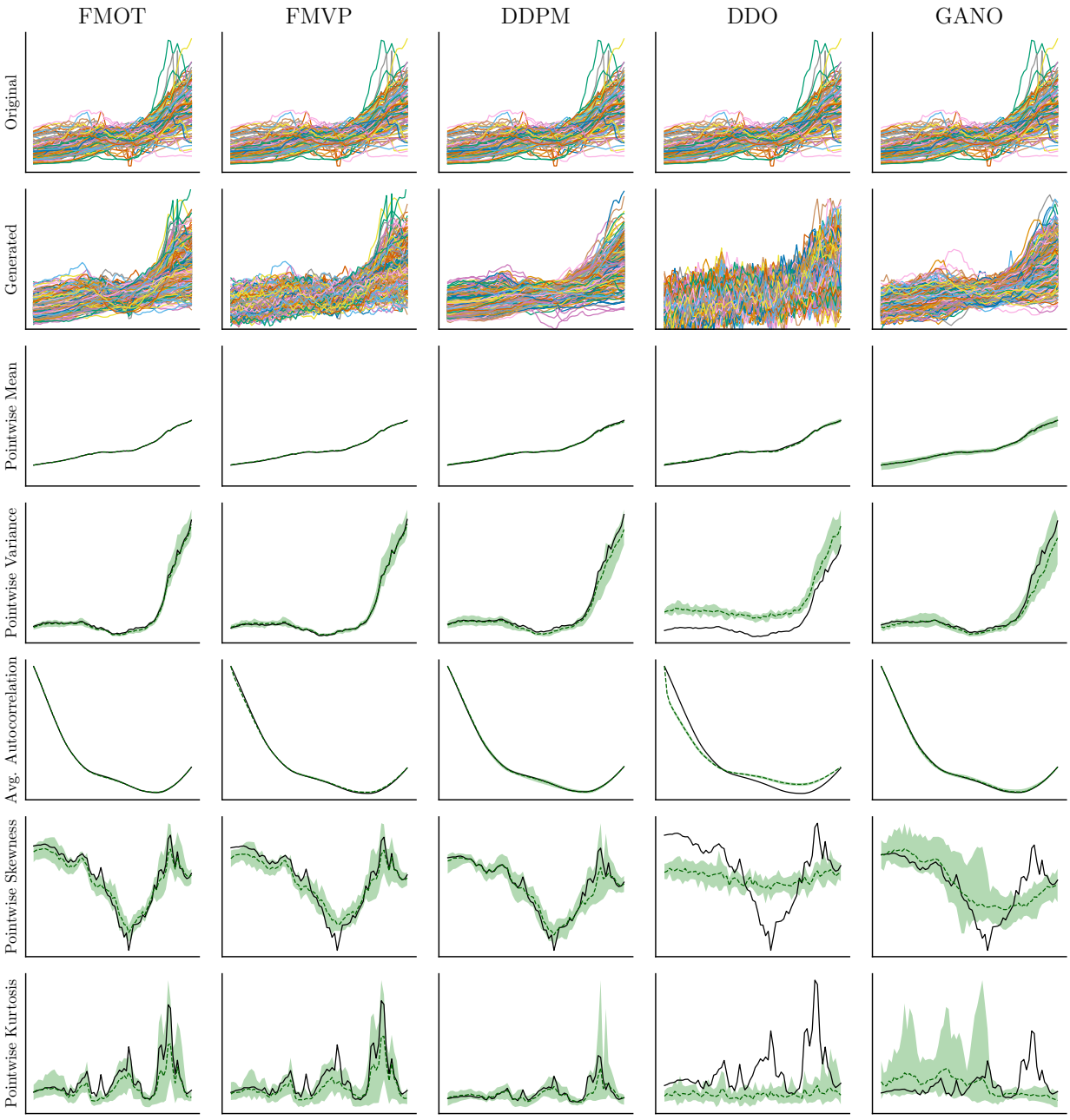


Figure B.7: Various pointwise statistics on the GDP dataset.

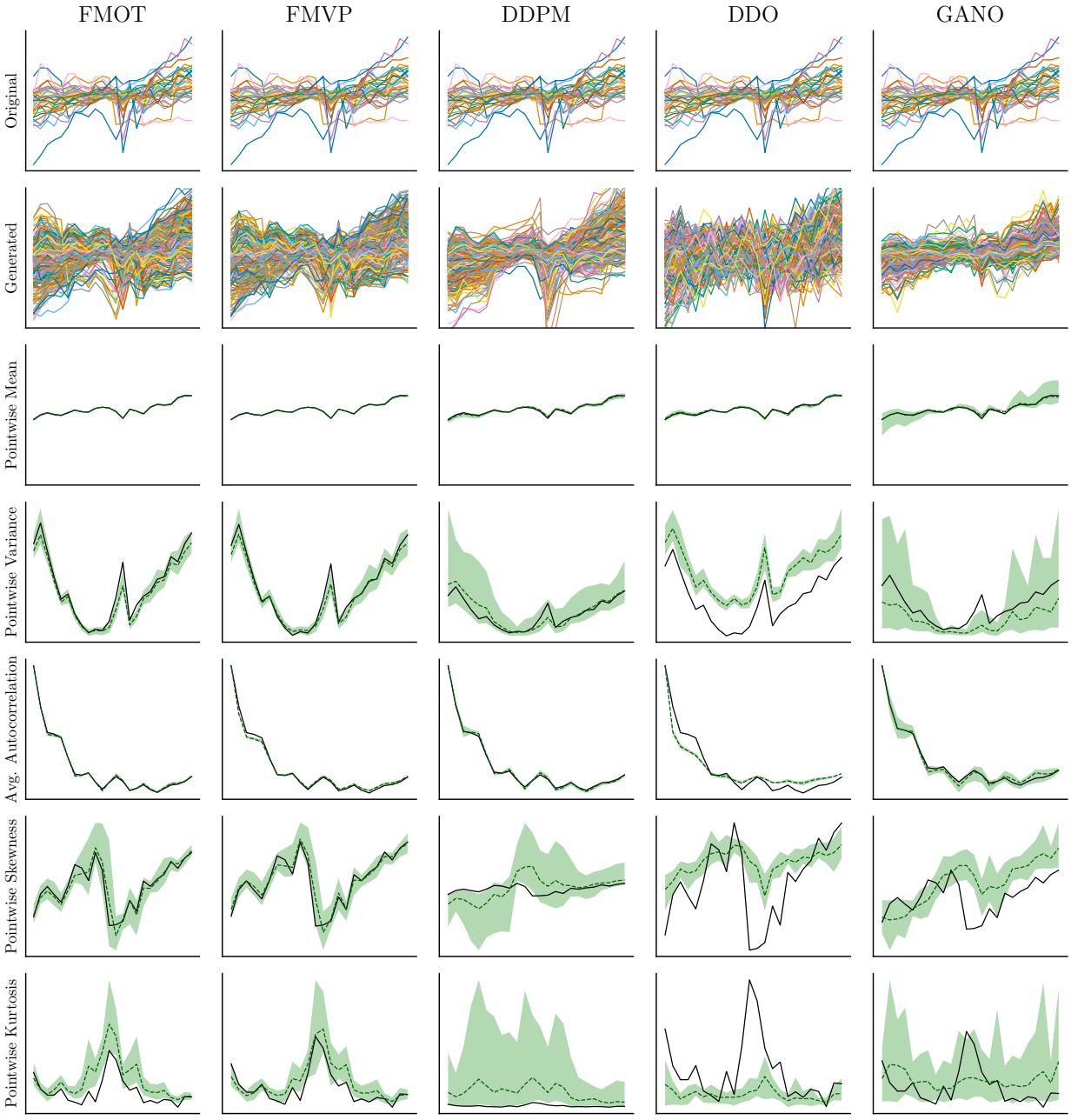


Figure B.8: Various pointwise statistics on the Labor dataset.

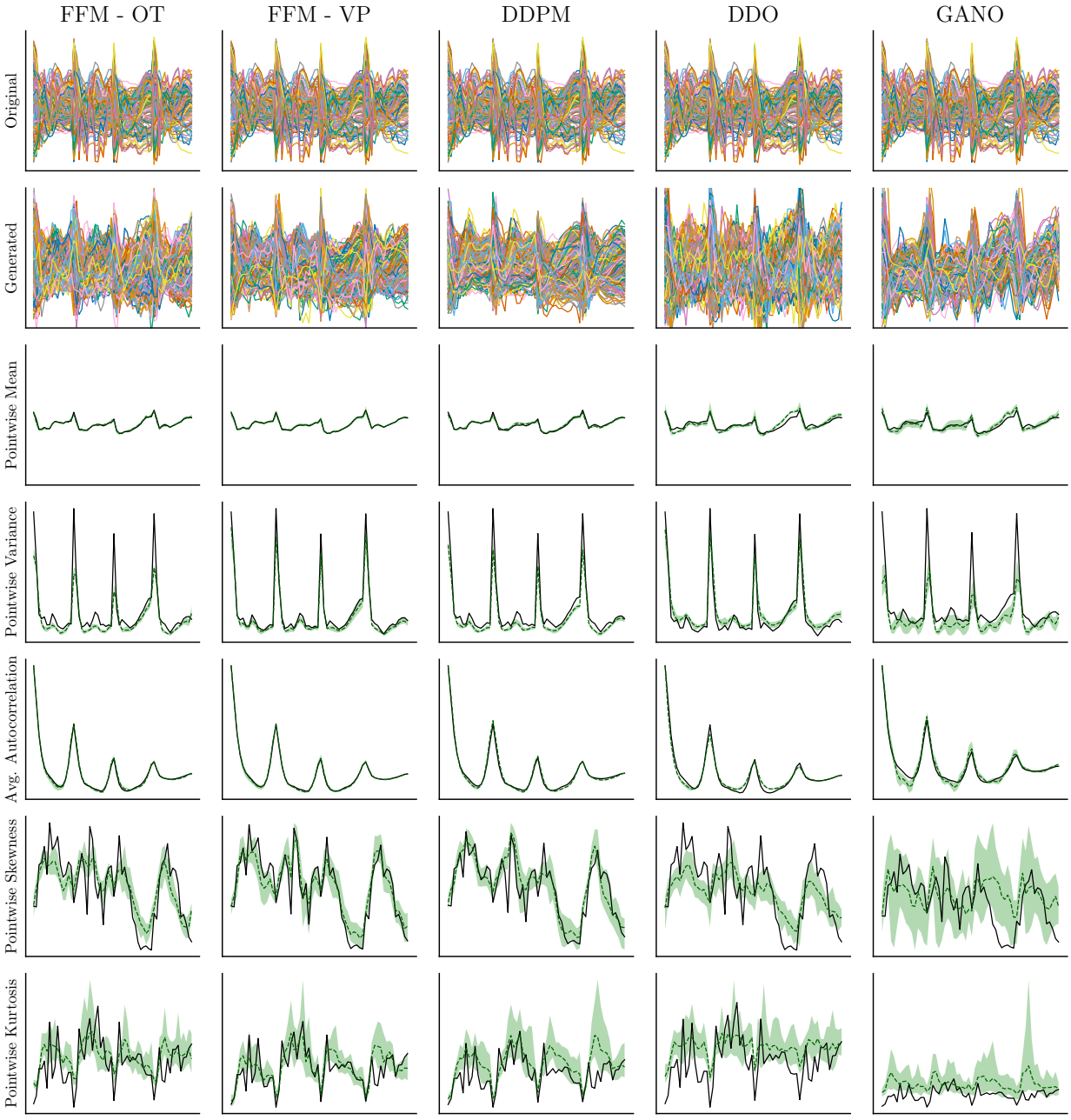


Figure B.9: Various pointwise statistics on the Genes dataset.

B.2.4 Additional Results: Navier-Stokes Dataset

In this section, we provide additional visualizations and evaluation on the Navier-Stokes dataset, corresponding to the samples in Figure 4.2 and Table 4.2a in the main paper. The first row of Figure B.10 plots a Gaussian KDE with a fixed bandwidth of 0.5 for the pixel-wise values of both the real and generated samples across all methods. We observe that FFM and DDPM closely match the ground-truth distribution, whereas DDO places too much mass around zero, and GANO learns a multimodal distribution. In the second row, we plot the spectrum of both the real and generated samples, i.e. the log-energy as a function of the wavenumber. We see that FFM and DDPM closely match the true spectrum for low wavenumbers, whereas the fits of DDO and GANO are less close. For all models, the generated samples fail to match the true spectrum at high wavenumbers. We obtain quantitative metrics from these visualizations by considering the pointwise MSE between the ground truth and generated curves to obtain the metrics in Table 4.2a.

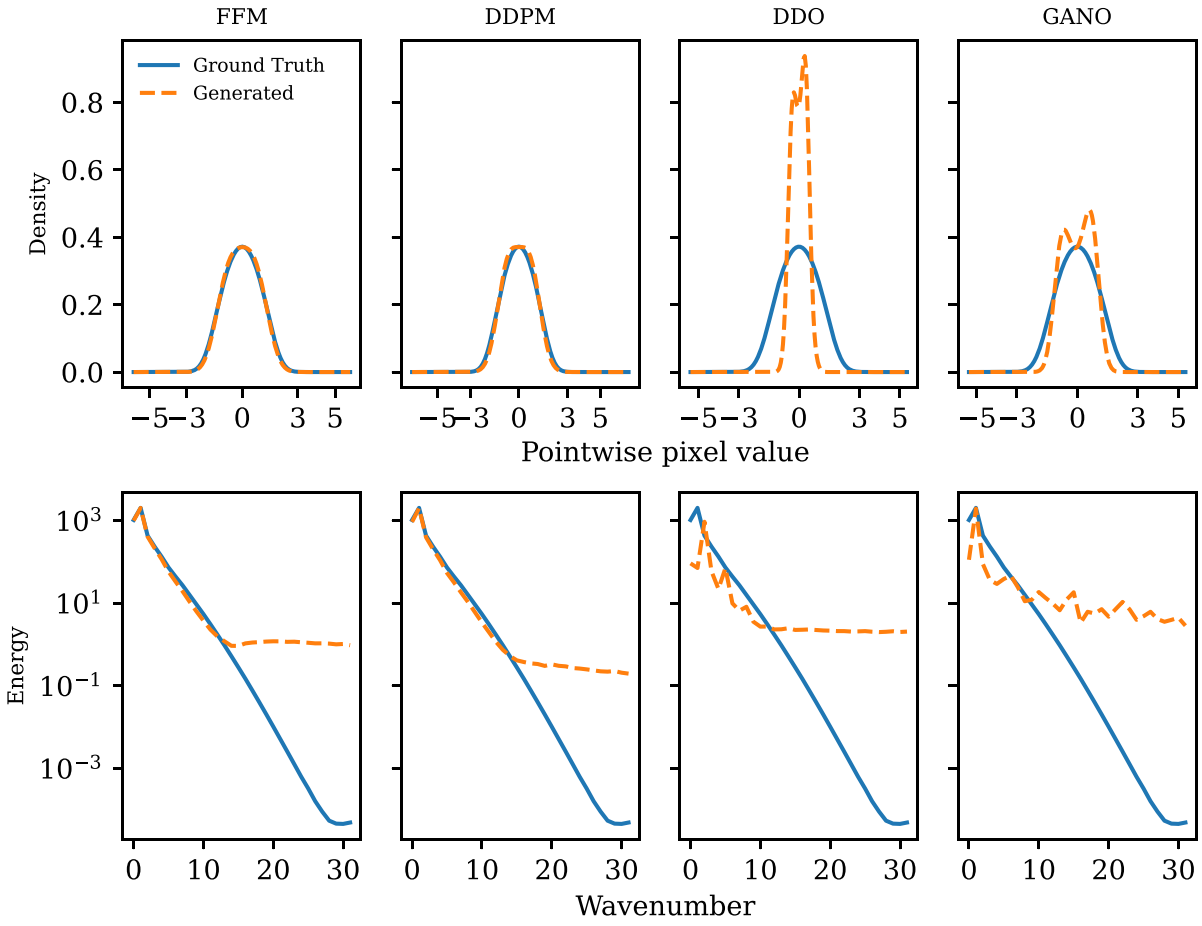


Figure B.10: Additional visualizations corresponding to the samples in Figure 4.2 and Table 4.2a. We use 1000 samples from each of the models.

B.3 Conditional Models

In addition to unconditional generation of functions, we demonstrate that our method can be extended to perform conditional generation. That is, we have access to side information $z \in \mathbb{R}^d$ (assumed to be finite dimensional) and we are interested in sampling from the conditional data measure $\nu(\,\mathrm{d}f \mid z)$. For instance, z could be a collection of observed values of some function, and we may be interesting in generating functions which interpolate (or extrapolate) these given observations. We describe two approaches to performing conditional generation: one based on a modified training process, and a second based on a modified sampling process. In Figure 3.2, we demonstrate these two approaches on the AEMET dataset.

Conditional Training. Using the unconditional paths of measures μ_t^f as described in the unconditional setting, we may define a conditional marginal $\mu_t(\,\mathrm{d}f \mid z)$ by mixing over $\mathrm{d}\nu(f \mid z)$, i.e. $\mu_t(A \mid z) := \int_H \mu_t^f(A)\,\mathrm{d}\nu(f \mid z)$.

As long as μ_t^f is concentrated around f, then $\mu_t(\,\mathrm{d}f \mid z) \approx \nu(\,\mathrm{d}f \mid z)$. Note that this condition is satisfied for the paths of measures constructed for unconditional generation, and hence no modification is necessary. However, modifying the conditional measures to account for the information z could potentially be beneficial, and we leave exploration of such design choices to future work. In all, we obtain a modified loss function

$$\mathcal{J}_{\mathrm{C}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],z\sim q(z),f\sim\nu(\,\mathrm{d}f|z),g\sim\mu_t^f}\left[\left\|v_t^f(g) - u_t(f \mid z,\theta)\right\|^2\right]. \tag{B.1}$$

In other words, we simply adapt our model architecture to also take in conditioning information z at training time. In practice, because z is assumed to be finite dimensional, we concatenate

z to the input of our FNO model [Li et al., 2021]. We note that a similar loss appears in the context of Flow Matching generative models for video, as proposed by [Davtyan et al., 2023].

Conditional Sampling. As an alternative, we may instead modify the sampling process to account for z. This allows one to train an unconditional model and sample conditionally at generation time (in contrast to the conditional training setup, which only allows you to condition on the particular form of z you have trained on). Here, we assume that $z = (\vec{x}, \vec{y})$ consists of a collection of function observations, and that we would like to generate functions whose values match those observed in z.

In order to achieve this, at time $t \in [0, 1]$, we take a step as dictated by our ODE solver and model vector field to obtain a function $\tilde{f}_t$. Next, we flow the information contained in z forwards for t seconds along the conditional vector field designated by our model to obtain $z_t = (\vec{x}, \vec{y}_t)$. Then, we set $f_t(\vec{x}) = \vec{y}_t$. This approach can be seen as an extension of the ILVR method, which has been successfully applied to diffusion models for conditional image generation [Choi et al., 2021] and diffusion models for conditional function generation [Kerrigan et al., 2023].

Appendix C

Supplementary Material: Chapter 5

This section contains additional material relating to Chapter 5. Section C.1 contains details necessary to reproduce our experiments (e.g., hyperparameter settings). Section C.2 contains additional details regarding the experiments on the 2D datasets used in this chapter, with additional figures illustrating the results appearing in the main body of Chapter 5. Sections C.3 and C.4 describe similar content for the Lotka-Volterra system and inverse Darcy flow, respectively.

C.1 Experiment Details

In this section, we provide additional details regarding all of our experiments, as well as additional results not contained within Chapter 5. All models can be trained on a single GPU with less than 24 GB of memory, and our experiments were parallelized over 8 such GPUs on a local server. We first describe our setting for the 2D and Lotka-Volterra experiments, as these share a similar setup. Details for the Darcy flow inverse problem are described in the corresponding section.

Table C.1: Hyperparameter grid used for random search of the FM and COT-FM models on the 2D and Lotka-Volterra datasets.

Hyperparameter	Description	Values
ϵ	COT coupling strength	[1e-6, 1e-4, 1e-2, 1e-1]
σ	Variance for $C = \sigma^2 I$ in (5.91)	[1e-3, 1e-2, 1e-1, 5e-1]
Batch Size	Training batch size	[256, 512, 1024]
Width	Layer width in MLP	[256, 512, 1024, 2048]
LR	Learning rate	[1e-4, 3e-4, 7e-4, 1e-3]
Layers	Number of MLP layers	[4, 6, 8]

Models. For FM and COT-FM, our model architecture is an MLP with SeLU activations [Klambauer et al., 2017]. Time conditioning is achieved by concatenating the time variable as an input to the network. The covariance operator C chosen in the path of measures in Equation (5.91) is taken to be $C = \sigma^2 I$ where σ is a hyperparameter.

Our implementation of FM is adapted from the `torchcfm` package Tong et al. [2024], available under the MIT License. For PCP-Map and COT-Flow, we adapt the open-source implementations from Wang et al. [2023], available under the MIT License.

Training and Model Selection. Hyperparameter tuning of the PCP-Map and COT-Flow models was performed directly using the code of Wang et al. [2023], essentially implementing grid-search with an early stopping procedure. We refer to the paper and codebase of Wang et al. [2023] for further details. For COT-FM and FM, we perform a random grid search over 100 hyperparameter settings using the grid described in Table C.1. For all model types, we select the best model used to generate the results in the paper as the training checkpoint that resulted in the lowest W_2 error to the joint target distribution on a held-out validation set. For training, we use the Adam optimizer where we only tune the learning rate, leaving all other settings as their defaults in `pytorch`.

C.2 2D Synthetic Data

Data Generation. This experiment consists of four 2D synthetic datasets, where $Y = U = \mathbb{R}$. The datasets moons, circles, swissroll are available through `scikit-learn` [Pedregosa et al., 2011]. The moons dataset is generated with `noise=`0.05 followed by standard scaling with a mean of $m = (0.5, 0.25)$ and standard deviation of $\sigma = (0.75, 0.25)$. The circles dataset is generated with `factor=0.5` and `noise=0.05`. The swissroll dataset is generated with `noise=0.75`, followed by projection to the first two coordinates and re-scaling by a factor of 12. All other unstated parameters are left as their default values. We use the code available from Hosseini et al. [2023] to generate the checkerboard dataset. For all datasets, we generate a training set (i.e., samples from the target distribution) of $20,000$ samples and $1,000$ held-out validation samples for model selection. Means and standard deviations in Table 5.1 are reported across five independent testing sets of 5,000 samples for the best representative of each model type.

In COT-FM, to generate samples from the source distribution, we sample an additional $20,000$ points from the target distribution and keep only the Y coordinates. This ensures that the source and target have equal Y marginals. During training, standard Gaussian noise $\mathcal{N}(0, 1)$ is sampled for the U coordinate of these source points at each minibatch.

We use minibatch COT couplings [Tong et al., 2024] in this experiment as computing the full COT plan was prohibitively expensive in terms of memory usage. However, we note that we use large batch sizes, meaning that the COT plan we find in this way should not be too far from optimal. All couplings are computed using the `POT` Python package [Flamary et al., 2021].

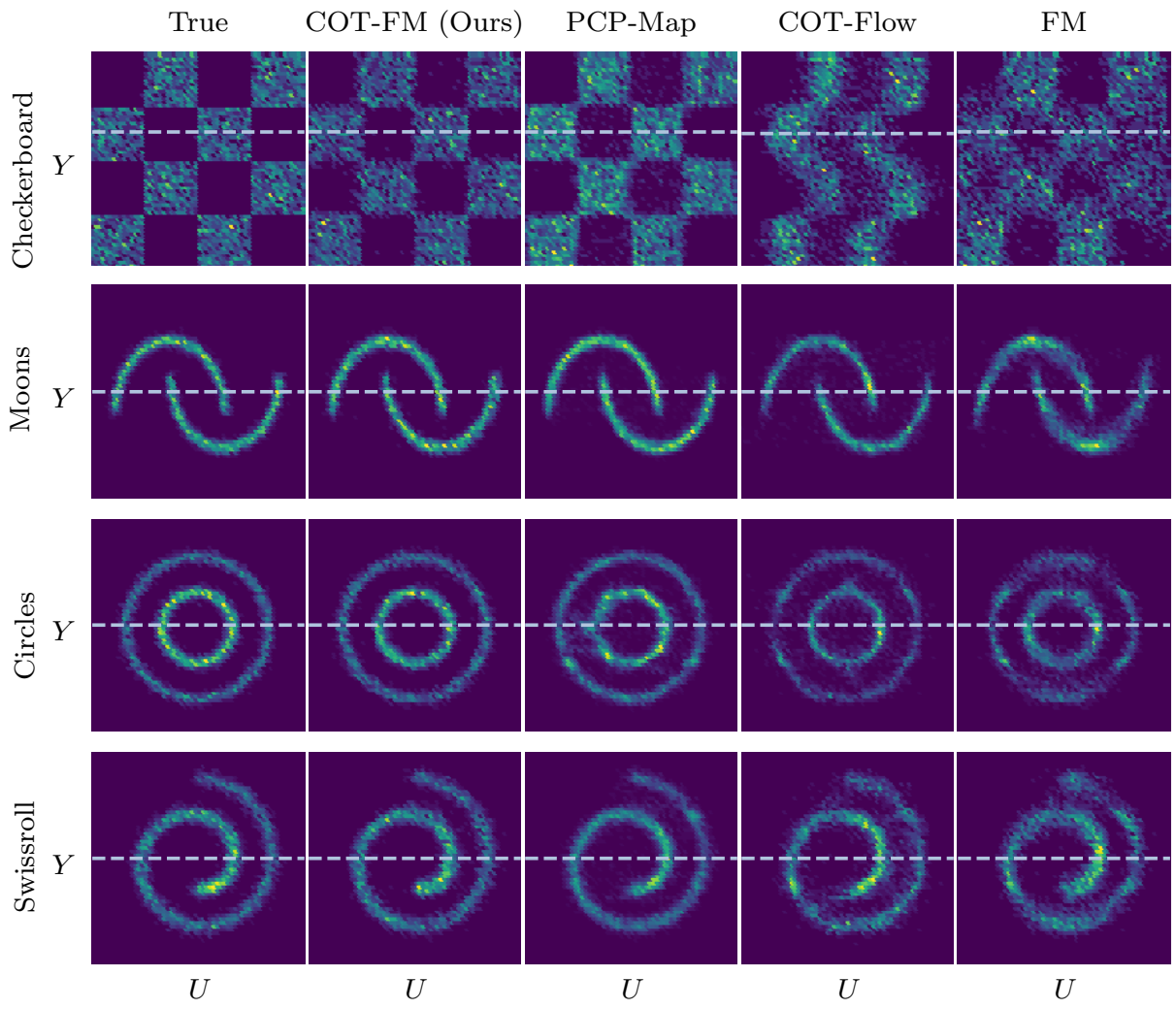


Figure C.1: Samples from the ground-truth joint target distribution and the various models for the 2D datasets. Samples from COT-FM more closely match the ground-truth distribution than the baselines. A common failure mode for the baselines is to generate samples from regions with zero support under the true data distributions. Table 5.1 contains a quantitative evaluation.

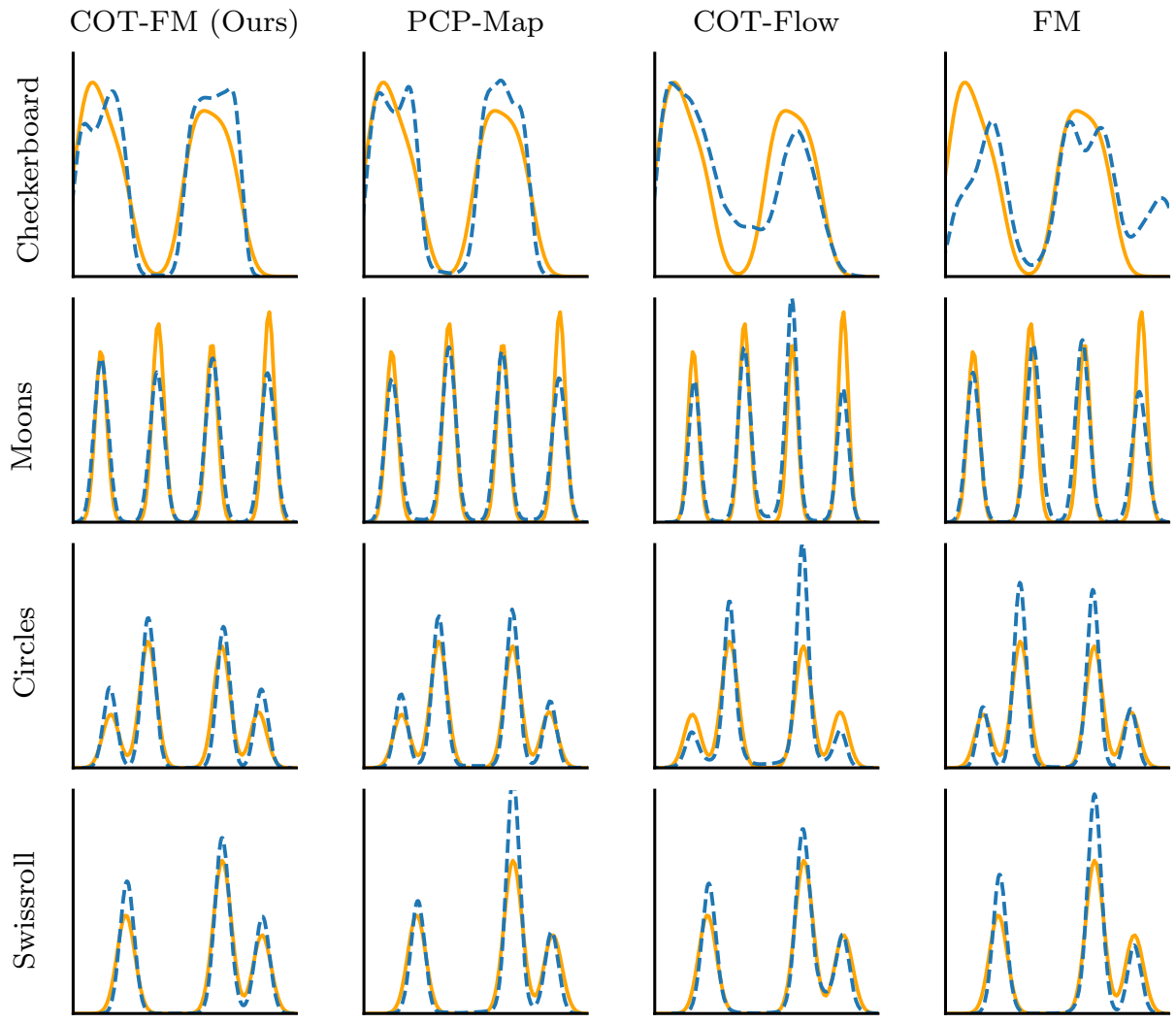


Figure C.2: Conditional KDEs shown for each of the methods on the 2D datasets. The conditioning variable y is fixed at the horizontal dashed line shown in Figure C.1. In all plots, the orange solid line indicates the CKDE of the ground-truth joint samples. In each column, the dashed blue line indicates the CKDE of samples generated from the respective method.

C.3 Lotka-Volterra Dynamical System

Data Generation. We adopt the settings of Alfonso et al. [2023] for this experiment. As described in the main paper, we assume $p(0) = (30, 1)$ and that $\log(u) \sim \mathcal{N}(m, 0.5I)$ with $m = (-0.125, -3, -0.125, -3)$. Given parameters $u \in \mathbb{R}^4_{\geq 0}$, we simulate Equation (5.94) for $t \in \{0, 2, \ldots, 20\}$ to obtain a solution $z(u) \in \mathbb{R}^{22}_{\geq 0}$. An observation $y \in \mathbb{R}^{22}_{\geq 0}$ is obtained by the addition of log-normal noise, i.e. $\log(y) \sim \mathcal{N}(\log(z(u), 0.1I)$. We thus may simulate many (y, u) pairs from the target measure for training.

We generate a training set of $10,000$ (y, u) pairs using the procedure described above and a held-out validation set of $10,000$ (y, u) pairs for model selection. Means and standard deviations in Table 5.2 are reported across five independent testing sets of 5,000 samples for the best representative of each model type. Figure 5.3 and Figures C.5, C.4, C.3, C.6 show $10,000$ samples from each model, as well as $10,000$ samples from the differential evolution Metropolis MCMC sampler [Braak, 2006] after a burn-in of $50,000$ samples. This is implemented through the `PyMC` Python package [Abril-Pla et al., 2023].

For COT-FM we use the full COT couplings, i.e. without minibatches. This is available to use due to the smaller size of the training set used in this experiment. The COT couplings are computed in the same way as the previous section, and as described in Section 5.6.

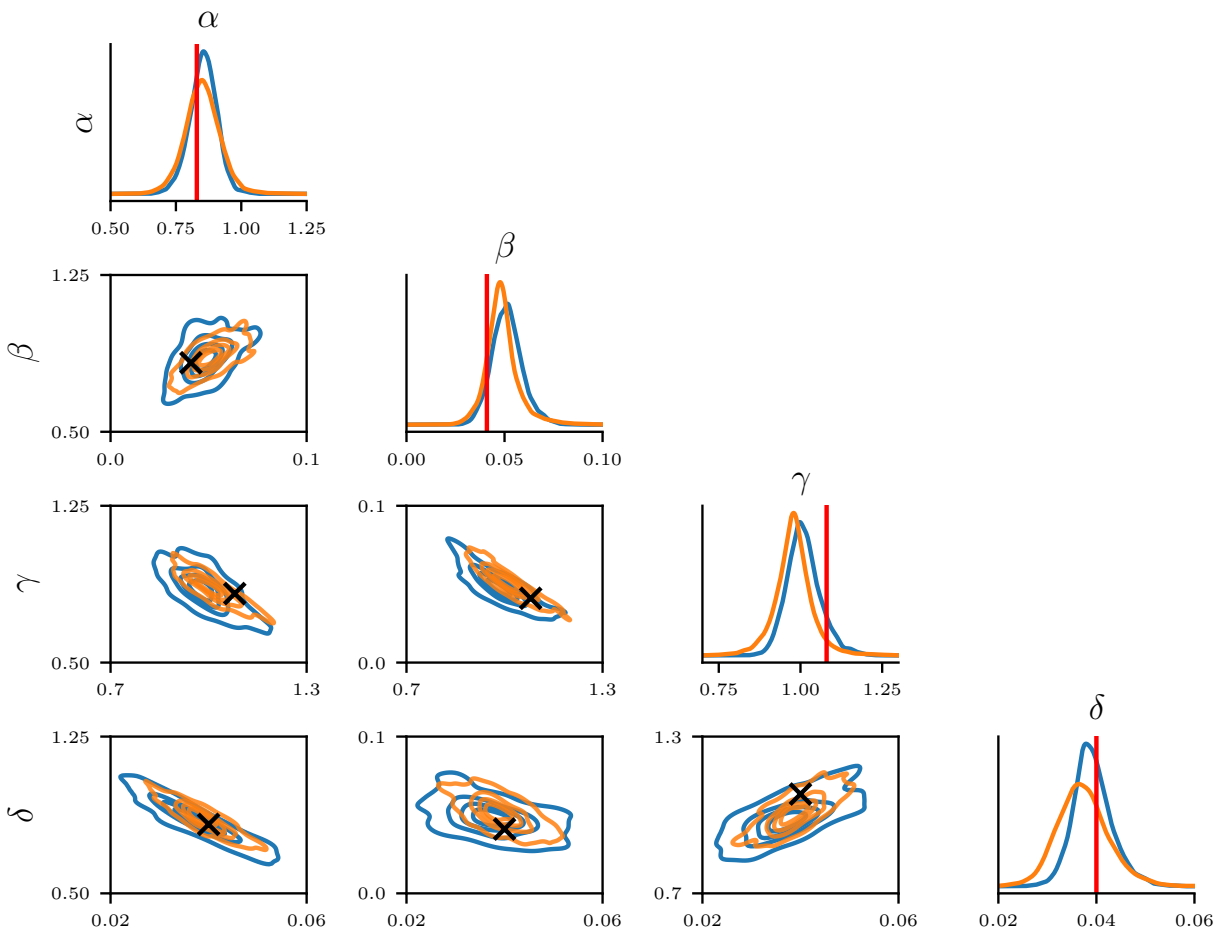


Figure C.3: KDE plots of the samples on the Lotka-Volterra system, using the settings described in Section 5.6. Plots include one-dimensional KDEs on the diagonal, as well as all two-dimensional pairs. In all plots, samples from MCMC are drawn in orange, and samples from our method (COT-FM) are indicated in blue. The true unknown parameters are indicated by the red vertical line in the diagonal plots, or the black x in the off-diagonal plots.

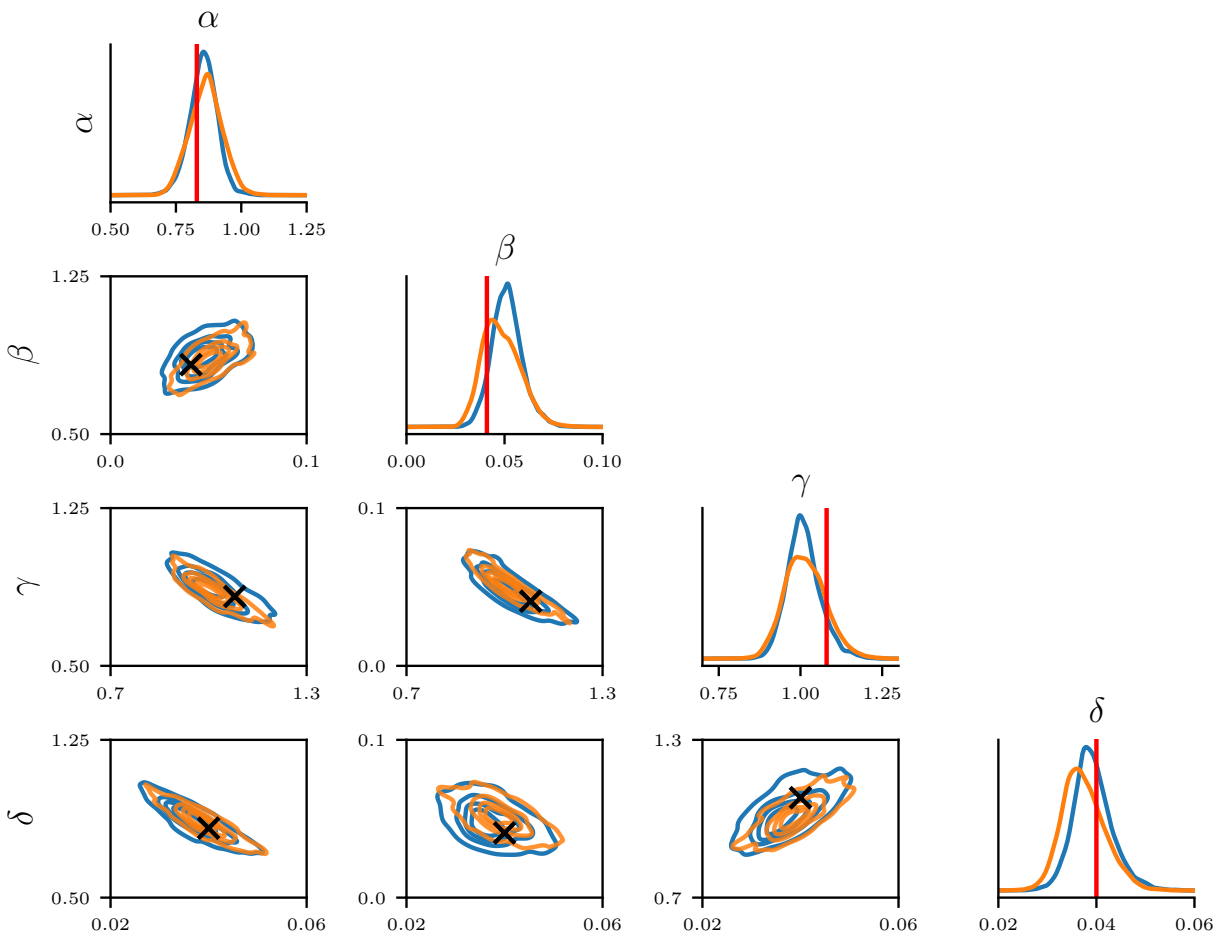


Figure C.4: KDE plots of the samples on the Lotka-Volterra system, using the settings described in Section 5.6. Plots include one-dimensional KDEs on the diagonal, as well as all two-dimensional pairs. In all plots, samples from MCMC are drawn in orange, and samples from PCP-Map are indicated in blue. The true unknown parameters are indicated by the red vertical line in the diagonal plots, or the black x in the off-diagonal plots.

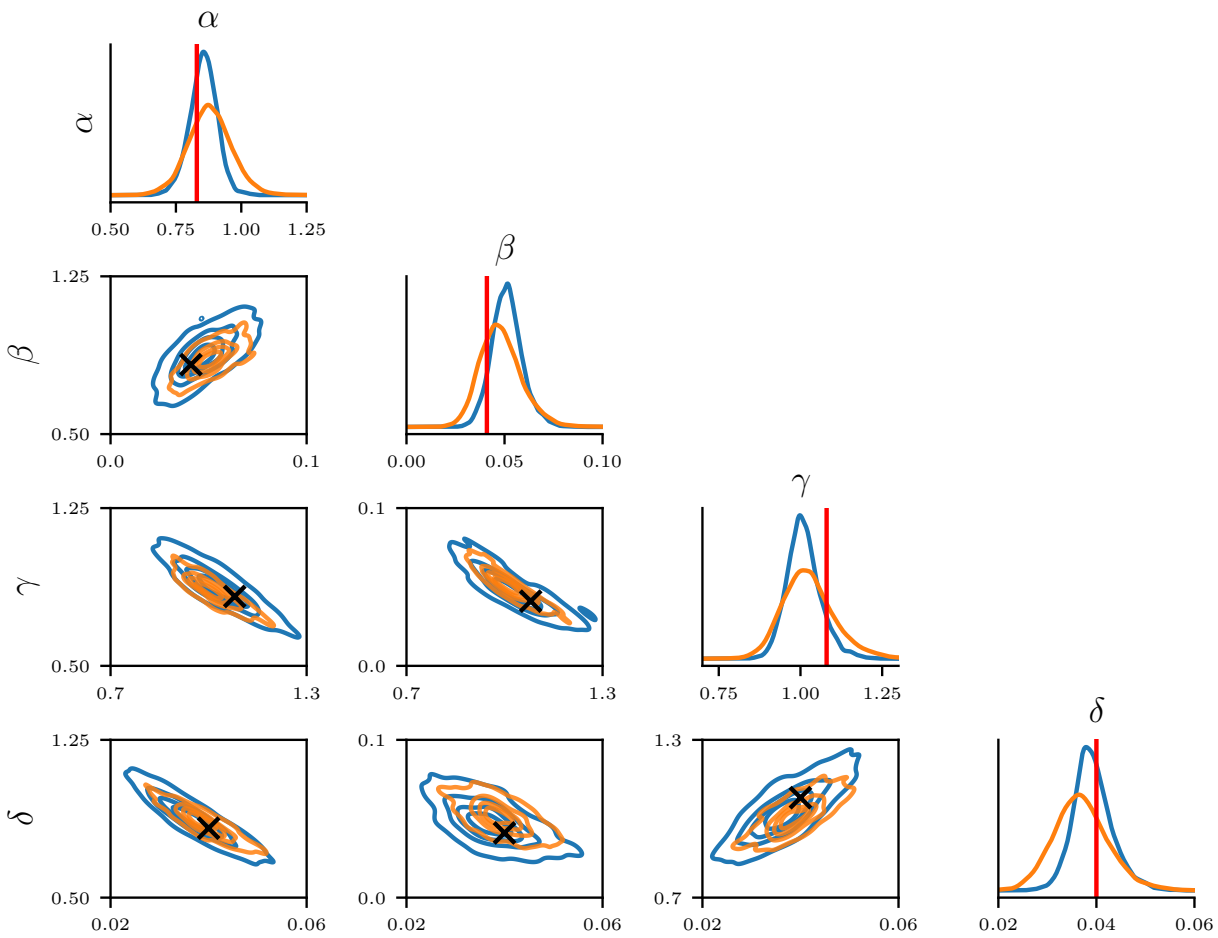


Figure C.5: KDE plots of the samples on the Lotka-Volterra system, using the settings described in Section 5.6. Plots include one-dimensional KDEs on the diagonal, as well as all two-dimensional pairs. In all plots, samples from MCMC are drawn in orange, and samples from COT-Flow are indicated in blue. The true unknown parameters are indicated by the red vertical line in the diagonal plots, or the black x in the off-diagonal plots.

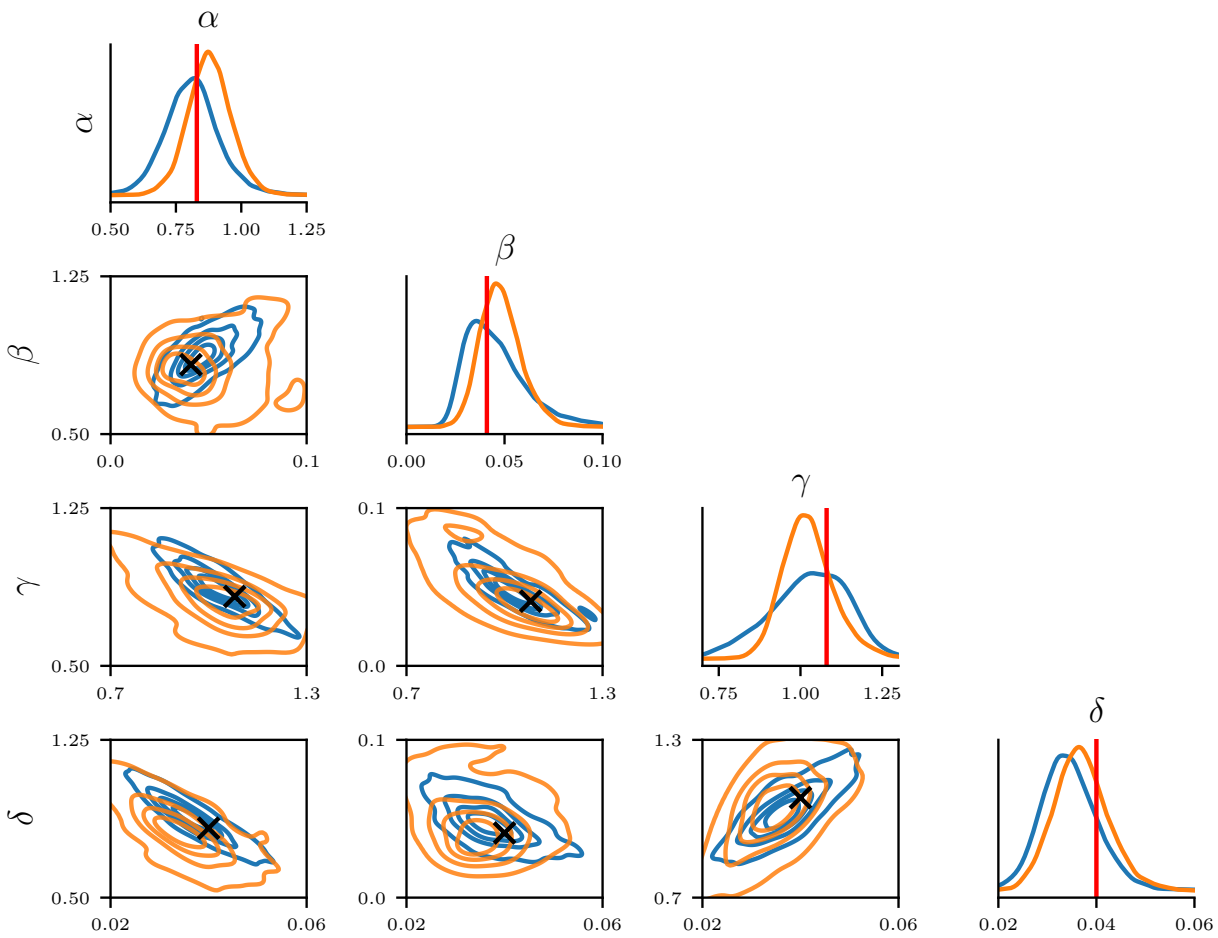


Figure C.6: KDE plots of the samples on the Lotka-Volterra system, using the settings described in Section 5.6. Plots include one-dimensional KDEs on the diagonal, as well as all two-dimensional pairs. In all plots, samples from MCMC are drawn in orange, and samples from flow matching (FM) are indicated in blue. The true unknown parameters are indicated by the red vertical line in the diagonal plots, or the black x in the off-diagonal plots.

C.4 Inverse Darcy Flow

Dataset. The training and test datasets are generated following the same procedure as Hosseini et al. [2023]: pressure fields u are sampled from a Gaussian process with Matérn kernel having $\nu = 3/2$ and lengthscale $\ell = 1/2$, on a regular 40×40 grid. The parameters are then exponentiated and used to simulate the permeability fields p from the forward model $\mathfrak{F}$ solving the Darcy flow PDE, using FEniCS [Alnæs et al., 2015]. Stochasticity arises from adding Gaussian noise to the permeability fields, obtaining $y = \mathfrak{F}(u) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. For our experiments we observe y on a 100×100 grid, and we use $\sigma = 2.5 \times 10^{-2}$. We note that this level of noise is quite considerable, as it accounts for roughly 60% of the variability in the y. Figure C.7 showcases a data point for reference. Our source and target training sets contain 1×10^4 samples each, and our test set comprises 5×10^3 samples. We remark that although y and u are observed on a grid their resolution does not need to be fixed, allowing for training at different resolutions.

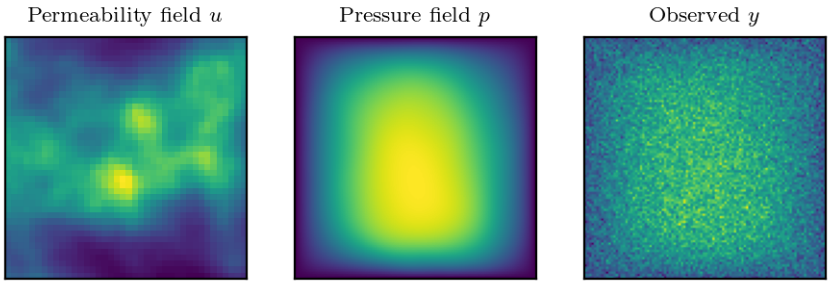


Figure C.7: Example of one random data point from the Darcy flow dataset.

Models. In order to make learning feasible in infinite-dimensional Hilbert spaces, we adapt the architecture of a Fourier Neural Operator (FNO) [Li et al., 2021] from the `neuraloperator` package [Kovachki et al., 2021] to accommodate for conditioning information observed at an arbitrary resolution. We do so by introducing a projection layer mapping the conditioning information to match the hidden channels of the input lifting block, and a pooling operation to project to the input dimensions. The two are then concatenated and passed through

an `FNOBlock` mapping from $(2 \times$ `hidden_channels`$) \times$ `input_dim` to `hidden_channels` $\times$ `input_dim`, before following the original architecture. For all of the models in consideration, we fix the architecture to be have `hidden_channels` $= 64$, `projection_channels` $= 256$, and 32 Fourier modes. We train each model for 1500 epochs, and hyperparameters for each architecture are selected as follows:

- WaMGAN [Hosseini et al., 2023]: using an adaptation to the FNO architecture of the original code[1], we perform a grid search as detailed in Table C.2. We found the training procedure to be rather unstable, and for this reason we checkpoint the model every 100 epochs and report the results for the best performing model at its best checkpoint. We found this to be a model with learning rate 1×10^{-4}, 2 full critic iterations, and monotone penalty of 1×10^{-3}. The gradient penalty parameter did not seem to significantly affect performance on the test set, and was set to 5.

- FFM [Kerrigan et al., 2024a]: the learning rate is fixed to 5×10^{-4}, and the covariance operator C is set to match that of the prior, but rescaled by a factor of $\sigma = 1 \times 10^{-3}$. We use the code from the original repository[2].

- COT-FFM: we set $\epsilon = 1 \times 10^{-5}$ in the cost function used to build the COT plan. The learning rate and C are chosen to be the same as FFM. In order to build COT couplings, we take the source measure to be the product measure $\pi^Y_\# \eta \times \mathcal{N}(0, C)$. Approximate couplings are obtained on minibatches of size 256.

It should be noted that in any scenario where the source and the target $U-$marginals are identical, using the OT coupling would yield the identity mapping as the optimal vector field minimizing (5.93). Hence, the OT-CFM model [Tong et al., 2024] is inapplicable here.

[1] https://github.com/TADSGroup/ConditionalOT2023
[2] https://github.com/GavinKerrigan/functional_flow_matching

Table C.2: Hyperparameter search space for WaMGAN

Parameter	Search Space
Learning rate	$\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$
Full critic iter.	$\{2, 5, 10\}$
Monotone penalty	$\{1 \times 10^{-3}, 5 \times 10^{-2}, 1 \times 10^{-1}\}$
Gradient penalty	$\{1, 5, 10\}$

Sampling. The resulting amortized sampler, denoted for simplification by the mapping $(y, u_0) \mapsto u_1 = \tilde{T}_U(y, u_0)$, will parameterize an approximate posterior measure. Notice that, in contrast to classical variational inference techniques, no distributional assumptions are made about the approximate posterior. In turn, integrals are obtained numerically by Monte Carlo sampling K samples from the prior, resulting in the approximation

$$\nu^y(f) \approx \int f \, \mathrm{d}\delta_{\tilde{T}_U(y,u_0)} \, \mathrm{d}\mathcal{N}(0, C) \approx \frac{1}{K} \sum_{k=1}^{K} f(\tilde{T}_U(y_k, u_{0,k})), \quad \{u_{0,k}\}_{k=1}^{K} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, C). \tag{C.1}$$

Appendix D

Supplementary Material: Chapter 6

This section contains additional material related to Chapter 6. More specifically, Section D.1 describes the datasets used in this chapter with some additional exploratory analyses. Section D.2 contains details required to reproduce our results, including model training and architecture details. Finally, Section D.3 provides a detailed discussion of the baseline models used in this chapter.

D.1 Datasets

In this section, we provide some additional details regarding the datasets used in this work. In Table D.1, we report the number of sequences in each dataset, some basic statistics regarding the number of events in each sequence, and their support $[0, T]$ and chosen forecast window ΔT. In all datasets, we use 60% of the data for training, 20% for validation, and the remaining 20% for testing.

Synthetic Datasets Our synthetic datasets are adopted from those proposed by Omi et al. [2019]. Each of these datasets consists of $1,000$ sequences supported on $\mathcal{T} = [0, 100]$. These synthetic datasets are chosen as they exhibit a wide range of behavior, ranging from i.i.d. inter-arrival times to self-correcting processes which discourage rapid bursts of events. We refer to Section 4 of Omi et al. [2019] for details.

Real-World Datasets We use the set of real-world datasets proposed in Shchur et al. [2020b], which constitute a set of standard benchmark datasets for unmarked TPPs. We refer to Appendix D of Shchur et al. [2020b] for additional details. With the exception of PUBG, these datasets are supported on $\mathcal{T} = [0, 24]$, i.e. each sequence corresponds to a single day. For the PUBG dataset, $\mathcal{T} = [0, 38]$ corresponds to the maximum length (in minutes) of an online game of PUBG. We note that PUBG has the largest number of sequences (which can lead to slow training), and the Reddit-C and Reddit-S datasets have long sequences (which can lead to slow training and high memory costs).

Table D.1: Some basic summary statistics of the datasets we consider in this work.

	Sequences	Mean length	Std length	Range length	Support	ΔT
Hawkes1	1000	95.4	45.8	[14, 300]	[0, 100]	–
Hawkes2	1000	97.2	49.1	[18, 355]	[0, 100]	–
Nonstationary Poisson	1000	100.3	9.8	[71, 134]	[0, 100]	–
Nonstationary Renewal	1000	98	2.9	[86, 100]	[0, 100]	–
Stationary Renewal	1000	109.2	38.1	[1, 219]	[0, 100]	–
Self-Correcting	1000	100.3	0.74	[98, 102]	[0, 100]	–
PUBG	3001	76.5	8.8	[26, 97]	[0, 38]	5
Reddit-C	1356	295.7	317.9	[1, 2137]	[0, 24]	4
Reddit-S	1094	1129	359.5	[363, 2658]	[0, 24]	4
Taxi	182	98.4	20	[12, 140]	[0, 24]	4
Twitter	2019	14.9	14	[1, 169]	[0, 24]	4
Yelp-Airport	319	30.5	7.5	[9, 55]	[0, 24]	4
Yelp-Miss.	319	55.2	15.9	[3, 107]	[0, 24]	4

D.2 EventFlow Architecture and Training Details

Here, we provide additional details regarding the parametrization and training of our `EventFlow` model. In general, our model is based on the transformer architecture [Vaswani et al., 2017, Yang et al., 2022], due to its general ability to handle variable length inputs and outputs, high flexibility, and ability to incorporate long-range interactions. In all settings, our reference measure μ_0 is specified with $q = \mathcal{N}(0, I)$.

Model Parametrization For our unconditional model, we first embed the sequence times γ_s, the flow-time s, and the sequence position indices using sinusoidal embeddings followed by an additional linear layer. There are three linear layers in total – one for the flow time, one shared across the sequence times, and one for the position indices. These embeddings are added together to create a representation of the sequence, and we apply a standard transformer to this sequence to produce a sequence of vectors of length $N(\gamma_s)$. Finally, each of these vectors is projected to one dimension via a final linear layer with shared weights to produce the vector field $v_\theta(\gamma_s, s)$. See Figure D.1.

For the conditional model, we use a standard transformer encoder-decoder architecture. We first embed the history sequence times $\mathcal{H}$ and the sequence position indices in a manner analogous to the above. In addition, the model was provided the start of the prediction window T_0 by concatenating it as the final event in $\mathcal{H}$. This yielded better results than encoding the start of the prediction window separately. We feed these embeddings through the transformer encoder produce an intermediate representation $e_{\mathcal{H}}$.

For the decoder, we provide the model with the current state γ_s (corresponding to the generated event times at flow-time s), the flow-time s, and the corresponding positional indices. These are embedded as previously described, before being passed into the transformer decoder. The history encoding $e_{\mathcal{H}}$ is provided to the decoder via cross-attention in the intermediate layer.

This produces a sequence of $N(\gamma_s)$ vectors, which we again pass through a final linear layer to produce the final conditional vector field $v_\theta(\gamma_s, s, e_{\mathcal{H}})$. See Figure D.2.

Our architecture for predicting the number of future events given a history, i.e. $p(n \mid \mathcal{H})$, is again based on the transformer decoder, sharing the same overall architecture as our unconditional model. However, the key difference is that we instead take a mean of the final sequence embeddings before passing this through a small MLP to produce the final logit. See Figure D.3.

Training and Tuning We normalize all sequences to the range $[-1, 1]$, using the overall min/max event time seen in the training data. All sequences are generated on this normalized scale, prior to re-scaling the sequence back to the original data range before evaluation. Our model is trained with the Adam [Kingma and Ba, 2014] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for $30,000$ steps with a cosine scheduler, which cycled every $10,000$ steps. Final hyperparameters were selected by best performance on the validation dataset achieved at any point during the training, where models were evaluated 10 times throughout their training.

To tune our model, we performed a grid search over learning rates in $\{5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}\}$ and dropout probabilities in $\{0, 0.1, 0.2\}$. Overall, we found that learning rates of 10^{-2} or larger often caused the model to diverge, and a dropout of 0.1 yielded the best results across all settings. We use 6 transformer layers, 8 attention heads, and an embedding dimension of 512 across all settings, except for the Reddit-C and Reddit-S datasets where we use 4 heads and an embedding dimension of 128 due to the increased memory cost of these datasets.

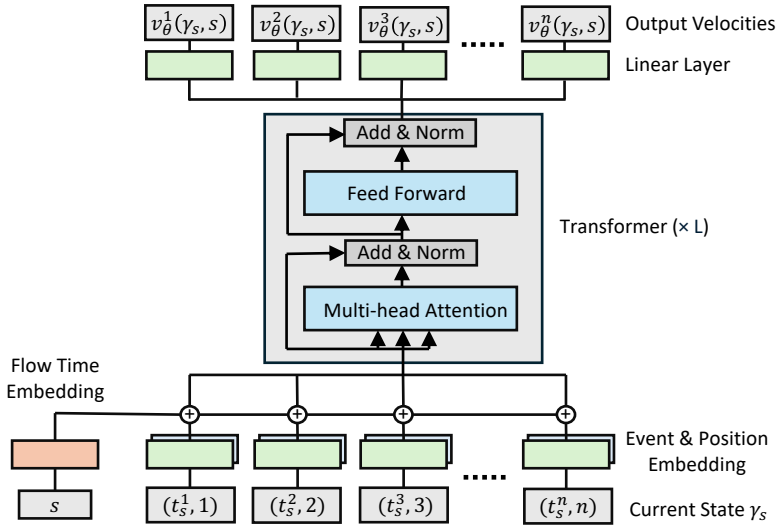


Figure D.1: Overview of our model architecture for unconditional generation. The model takes as input the flow time s and current sequence state $\gamma_s = \sum_{k=1}^{n} \delta[t_s^k]$. Each input is projected to a fixed-length vector via a learnable embedding. The resulting embeddings are added and passed to the transformer model, which produces a sequence of output velocities $v_\theta(\gamma_s, s)$ with $N(\gamma_s)$ components.

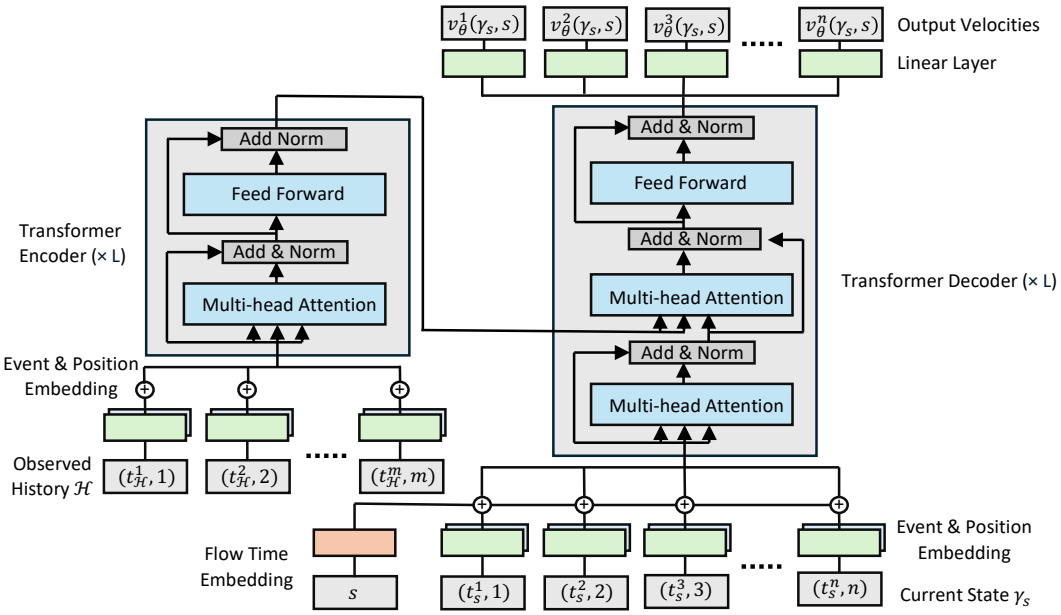


Figure D.2: Overview of our model architecture for conditional generation. The encoder (left) takes as input the observed history $\mathcal{H}$, which is embedded in a fashion analogous to our unconditional model. The decoder (right) takes as input the flow time s and current state $\gamma_s = \sum_{k=1} \delta[t_s^k]$. These are embedded and passed through the decoder, which applies cross attention to produce the conditional velocities $v_\theta(\gamma_s, s, e_{\mathcal{H}})$.

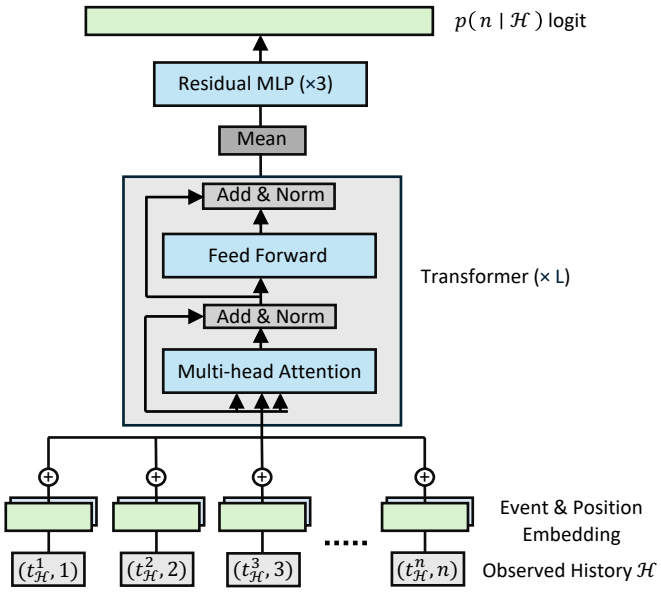


Figure D.3: Overview of our architecture modeling the event count distribution $p_\phi(n \mid \mathcal{H})$. The model takes as input an observed history $\mathcal{H}$. As in our other architectures, the events are embedded and passed through a transformer. Here, the final sequence embedding output by the transformer is averaged and passed through an additional residual MLP with three layers to produce the logit corresponding to $p(n \mid \mathcal{H})$.

Table D.2: The best hyperparameter settings found for the vector field v_θ in our EventFlow method on the unconditional generation task.

	Learning Rate	Emb. Dim.	MLP Dim	Heads	Transformer Layers
Hawkes1	10^{-3}	512	2048	8	6
Hawkes2	10^{-3}	512	2048	8	6
Nonstationary Poisson	10^{-3}	512	2048	8	6
Nonstationary Renewal	10^{-3}	512	2048	8	6
Stationary Renewal	10^{-3}	512	2048	8	6
Self-Correcting	10^{-3}	512	2048	8	6
PUBG	5×10^{-4}	512	2048	8	6
Reddit-C	10^{-3}	128	256	4	6
Reddit-S	5×10^{-3}	128	256	4	6
Taxi	5×10^{-4}	512	2048	8	6
Twitter	10^{-3}	512	2048	8	6
Yelp-Airport	5×10^{-4}	512	2048	8	6
Yelp-Miss.	10^{-3}	512	2048	8	6

Table D.3: The best hyperparameter settings found for the vector field v_θ in our EventFlow method on the forecasting task.

	Learning Rate	Emb. Dim.	MLP Dim.	Heads	Transformer Layers
PUBG	10^{-3}	512	2048	8	6
Reddit-C	10^{-3}	128	256	4	6
Reddit-S	10^{-3}	128	256	4	6
Taxi	10^{-3}	512	2048	8	6
Twitter	5×10^{-4}	512	2048	8	6
Yelp-Airport	10^{-3}	512	2048	8	6
Yelp-Miss.	10^{-3}	512	2048	8	6

Table D.4: The best hyperparameter settings found for the event count predictor $p(n \mid \mathcal{H})$ in our EventFlow method on the forecasting task.

	Learning Rate	Emb. Dim.	MLP Dim.	Heads	Transformer Layers
PUBG	5×10^{-4}	512	2048	8	6
Reddit-C	10^{-3}	128	256	4	6
Reddit-S	10^{-3}	128	256	4	6
Taxi	5×10^{-4}	512	2048	8	6
Twitter	5×10^{-4}	512	2048	8	6
Yelp-Airport	5×10^{-4}	512	2048	8	6
Yelp-Miss.	5×10^{-4}	512	2048	8	6

D.3 Additional details on baselines

In this section, we provide additional details regarding our baseline methods. All methods are trained at a batch size of 64 for $1,000$ epochs, using early stopping on the validation set loss. In early experiments, we also evaluated AttNHP [Zuo et al., 2020], a variant of the NHP which uses an attention-based encoder, but found it to be prohibitively expensive in terms of memory cost (requiring more than 24 GB of VRAM) and, as a result, do not include it in our results.

IFTPP Our first baseline is the intensity-free TPP model of Shchur et al. [2020a]. This model uses an RNN encoder and a mixture of log-normal distributions to parametrize the decoder. We directly use the implementation provided by the authors.[1]. We train for $1,000$ epochs with early stopping based on the validation set loss. To tune this baseline, we performed a grid search over learning rates in $\{10^{-4}, 10^{-3}, 10^{-2}\}$, weight decays in $\{0, 10^{-6}, 10^{-5}, 10^{-4}\}$, history embedding dimensions $\{32, 64, 128\}$, and mixture component counts $\{8, 16, 32, 64\}$. Our best hyperparameters can be found in Table D.5 and Table D.6.

NHP We additionally compare against the Neural Hawkes Process of Mei and Eisner [2017]. This model uses an LSTM encoder and a parametric form, whose weights are modeled by a neural network, to model the conditional intensity function. In practice, we use the implementation proved by the EasyTPP benchmark [Xue et al., 2024], as this version implements the necessary thinning algorithm for sampling.[2] We perform a grid search over learning rates in $\{10^{-4}, 10^{-3}, 10^{-2}\}$ and embedding dimensions in $\{32, 64, 128\}$. These hyperparameters are chosen as the EasyTPP implementation allows these to be configured easily. Our best hyperparameters are reported in Table D.7 and Table D.8.

[1] URL: `https://github.com/shchur/ifl-tpp`
[2] URL: `https://github.com/ant-research/EasyTemporalPointProcess`

Table D.5: The best hyperparameter settings found for IFTPP on the unconditional generation task.

	Learning Rate	Weight Decay	Embedding Dimension	Mixture Components
Hawkes1	10^{-3}	10^{-4}	32	8
Hawkes2	10^{-2}	0	32	8
Nonstationary Poisson	10^{-3}	10^{-6}	128	8
Nonstationary Renewal	10^{-2}	10^{-6}	64	16
Stationary Renewal	10^{-3}	10^{-4}	32	8
Self-Correcting	10^{-3}	10^{-6}	32	64
PUBG	10^{-2}	0	128	32
Reddit-C	10^{-3}	10^{-4}	64	16
Reddit-S	10^{-2}	10^{-4}	64	16
Taxi	10^{-2}	10^{-5}	128	64
Twitter	10^{-3}	10^{-4}	64	6
Yelp-Airport	10^{-2}	10^{-6}	64	64
Yelp-Miss.	10^{-3}	10^{-4}	32	8

Table D.6: The best hyperparameter settings found for IFTPP on the forecasting task.

	Learning Rate	Weight Decay	Embedding Dimension	Mixture Components
PUBG	10^{-4}	10^{-6}	32	32
Reddit-C	10^{-2}	0	64	8
Reddit-S	10^{-2}	0	64	16
Taxi	10^{-3}	10^{-6}	128	8
Twitter	10^{-2}	10^{-5}	32	8
Yelp-Airport	10^{-2}	10^{-6}	128	32
Yelp-Miss.	10^{-2}	10^{-6}	32	8

Diffusion Our diffusion baseline is based on the implementation of Lin et al. [2022], and our decoder model architecture is taken directly from the code of Lin et al. [2022].[3] At a high level, this model is a discrete-time diffusion model [Ho et al., 2020] trained to generate a single inter-arrival time given a history embedding. Note that as the likelihood is not available in diffusion models, the CDF in the likelihood in Equation (6.2) is not tractable. Instead, the model is trained by maximizing an ELBO of only the subsequent inter-arrival time.

In preliminary experiments, we found that the codebase provided by Lin et al. [2022] often produced `NaN` values during sampling, prompting us to make several changes. First, we use

[3]URL: `https://github.com/EDAPINENUT/GNTPP`

the RNN encoder from Shchur et al. [2020a], i.e. the same encoder as the IFTPP baseline, to reduce the memory requirements of the model. Second, we do not log-scale the inter-arrival times as suggested by Lin et al. [2022], as we found that this often led to overflow and underflow issues at sampling time. Third, we do not normalize the data via standardization (i.e., subtracting off the mean inter-arrival time and dividing by the standard deviation), but rather, we scale the inter-arrival times so that they are in the bounded range $[-1, 1]$. This is aligned with standard diffusion implementations [Ho et al., 2020], and allows us to perform clipping at sampling time to avoid the accumulation of errors. With these changes, our diffusion baseline is competitive, and able to obtain stronger results than previous work has reported [Lüdke et al., 2023].

We use 1000 diffusion steps and the cosine beta schedule [Nichol and Dhariwal, 2021], and we train the model on the simplified ϵ-prediction loss of Ho et al. [2020]. We train for $1,000$ epochs with early stopping based on the validation set loss. To tune this baseline, we performed a grid search over learning rates in $\{10^{-4}, 10^{-3}, 10^{-2}\}$, weight decays in $\{0, 10^{-6}, 10^{-5}, 10^{-4}\}$, history embedding dimensions $\{32, 64, 128\}$, and layer numbers $\{2, 4, 6\}$. Our best hyperparameters can be found in Table D.9 and Table D.10.

Add-and-Thin We compare to the `Add-and-Thin` model of Lüdke et al. [2023] as a recently proposed non-autoregressive baseline. We directly run the code provided by the authors without additional modifications.[4] We do, however, perform a slightly larger hyperparameter sweep than Lüdke et al. [2023], in order to ensure a fair comparison between the methods considered. We train for $1,000$ epochs with early stopping on the validation loss. Tuning is performed via a grid search over learning rates in $\{10^{-4}, 10^{-3}, 10^{-2}\}$ and number of mixture components in $\{8, 16, 32, 64\}$. We choose to tune only these hyperparameters in order to follow the implementation provided by the authors. Our best hyperparameters can be found in Table D.11 and Table D.12.

[4]URL: `https://github.com/davecasp/add-thin`

Table D.7: The best hyperparameter settings found for NHP on the unconditional generation task.

	Learning Rate	Embedding Dimension
Hawkes1	10^{-3}	64
Hawkes2	10^{-3}	64
Nonstationary Poisson	10^{-3}	64
Nonstationary Renewal	10^{-4}	64
Stationary Renewal	10^{-3}	64
Self-Correcting	10^{-3}	64
PUBG	10^{-4}	64
Reddit-C	10^{-2}	64
Reddit-S	10^{-2}	64
Taxi	10^{-2}	64
Twitter	10^{-4}	64
Yelp-Airport	10^{-3}	128
Yelp-Miss.	10^{-2}	64

Table D.8: The best hyperparameter settings found for NHP on the forecasting task.

	Learning Rate	Embedding Dimension
PUBG	10^{-3}	128
Reddit-C	10^{-2}	64
Reddit-S	10^{-2}	64
Taxi	10^{-2}	128
Twitter	10^{-2}	128
Yelp-Airport	10^{-3}	64
Yelp-Miss.	10^{-2}	64

Table D.9: The best hyperparameter settings found for diffusion on the unconditional generation task.

	Learning Rate	Weight Decay	Embedding Dimension	Layers
Hawkes1	10^{-3}	10^{-6}	64	2
Hawkes2	10^{-2}	10^{-5}	64	4
Nonstationary Poisson	10^{-3}	10^{-5}	128	2
Nonstationary Renewal	10^{-3}	10^{-4}	64	2
Stationary Renewal	10^{-2}	0	32	6
Self-Correcting	10^{-3}	0	32	6
PUBG	10^{-3}	0	64	2
Reddit-C	10^{-3}	10^{-6}	128	4
Reddit-S	10^{-3}	0	64	4
Taxi	10^{-2}	0	128	4
Twitter	10^{-3}	10^{-4}	64	6
Yelp-Airport	10^{-2}	0	32	2
Yelp-Miss.	10^{-2}	10^{-5}	128	2

Table D.10: The best hyperparameter settings found for diffusion on the forecasting task.

	Learning Rate	Weight Decay	Embedding Dimension	Layers
PUBG	10^{-4}	10^{-5}	32	6
Reddit-C	10^{-2}	10^{-6}	64	6
Reddit-S	10^{-3}	0	64	4
Taxi	10^{-3}	10^{-6}	32	2
Twitter	10^{-4}	10^{-5}	64	6
Yelp-Airport	10^{-4}	10^{-5}	64	6
Yelp-Miss.	10^{-3}	10^{-5}	32	4

Table D.11: The best hyperparameter settings found for Add-and-Thin on the unconditional generation task.

	Learning Rate	Mixture Components
Hawkes1	10^{-3}	32
Hawkes2	10^{-2}	32
Nonstationary Poisson	10^{-2}	16
Nonstationary Renewal	10^{-2}	8
Stationary Renewal	10^{-2}	8
Self-Correcting	10^{-4}	8
PUBG	10^{-3}	8
Reddit-C	10^{-2}	32
Reddit-S	10^{-2}	16
Taxi	10^{-2}	8
Twitter	10^{-4}	32
Yelp-Airport	10^{-4}	8
Yelp-Miss.	10^{-2}	64

Table D.12: The best hyperparameter settings found for Add-and-Thin on the forecasting task.

	Learning Rate	Mixture Components
PUBG	10^{-2}	64
Reddit-C	10^{-2}	16
Reddit-S	10^{-2}	64
Taxi	10^{-2}	8
Twitter	10^{-3}	8
Yelp-Airport	10^{-2}	32
Yelp-Miss.	10^{-3}	16