Neural McKean-Vlasov Processes: Distributional Dependence in Diffusion Processes

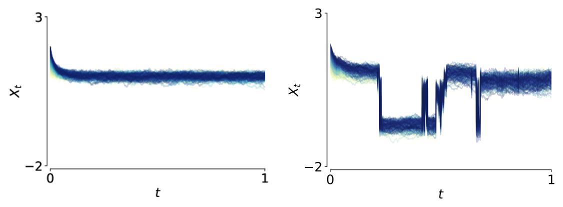
Haoming Yang[*] **Ali Hasan**[*†] **Yuting Ng** **Vahid Tarokh**
Duke University

Abstract

McKean-Vlasov stochastic differential equations (MV-SDEs) provide a mathematical description of the behavior of an infinite number of interacting particles by imposing a dependence on the particle density. We study the influence of explicitly including distributional information in the parameterization of the SDE. We propose a series of semi-parametric methods for representing MV-SDEs, and corresponding estimators for inferring parameters from data based on the properties of the MV-SDE. We analyze the characteristics of the different architectures and estimators, and consider their applicability in relevant machine learning problems. We empirically compare the performance of the different architectures and estimators on real and synthetic datasets for time series and probabilistic modeling. The results suggest that explicitly including distributional dependence in the parameterization of the SDE is effective in modeling temporal data with interaction under an exchangeability assumption while maintaining strong performance for standard Itô-SDEs due to the richer class of probability flows associated with MV-SDEs.

1 Introduction

Stochastic differential equations (SDEs) model the evolution of a stochastic process through two functions known as the *drift* and *diffusion* functions. With Itô-SDEs, individual sample paths are assumed to be independent, and neural representations of the drift and diffusion have achieved high performance in many applications such as in time series and generative modeling (Song et al., 2020; Tashiro et al., 2021).



(a) Itô-SDE (b) MV-SDE

Figure 1: SDE sample paths of a double-well potential, where the particles (a) do not interact and (b) exhibit complex phase transitions as a result only of interaction through weak attraction.

On the other hand, interacting particle systems are also used to model stochastic processes; they share many characteristics with Itô-SDEs but additionally dictate an interaction between the different sample paths (Liggett, 1997). This has been used as an interpretation of the dynamics of machine learning algorithms, most notably for the attention mechanism in the transformer architecture (e.g. in (Sander et al., 2022)). When the number of particles approaches infinity, these processes generalize Itô-SDEs to *nonlinear SDEs* known as McKean-Vlasov SDEs (MV-SDEs). The nonlinearity arises from the individual particle's dependence on the whole particle density, often in the form of a *mean-field* term represented by an expectation with respect to the particle density. This distributional dependence allows for greater flexibility in the time marginal distributions that the MV-SDE can represent versus the Itô-SDE. An example of the differences between the two frameworks is illustrated in Figure 1 where Figure 1a depicts an Itô-SDE where the sample paths are independent and Figure 1b depicts a MV-SDE where the sample paths interact through distributional dependence. While these models appear

[*]Equal contribution, junior author listed first.
[†]Correspondence: ali.hasan@duke.edu

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

in a variety of disciplines such as in finance (Feinstein and Søjmark, 2021), biology (Keller and Segel, 1971), and social sciences (Carrillo et al., 2020), relatively few works have considered the application in machine learning tasks. Moreover, representing such processes requires careful consideration, since the drift and diffusion parameterizations must now include *distributional information* rather than just the state as inputs.

This brings us to a motivating question:

(Q1) Can we develop neural architectures that satisfy the properties of MV-SDEs?

To answer *(Q1)*, we consider two ideas: (i) expressing a layer in a neural network as an expectation under a change of measure and (ii) using generative models to capture distributions and compute empirical expectations. Our second question is concerned with how the theoretical generality of MV-SDEs to Itô-SDEs occurs in practice:

(Q2) Does explicitly including distributional dependence improve modeling capabilities?

This question is related to the initial insight that the MV-SDE can be viewed as a generalization of powerful architectures, such as the transformer, and we are interested in whether adding distributional dependence improves modeling capabilities. We discuss a few theoretical properties that motivate this question and answer the question affirmatively by empirically comparing different architectures for applications in time series and in probabilistic modeling.

1.1 Related work

Estimating SDEs from data is an important component of many scientific disciplines with a number of methods developed specifically for this task, e.g. Chen et al. (2021); Hasan et al. (2021). Methods that estimate MV-SDEs from observations often assume known interaction kernels and drift parameters. They then rely on a large number of samples at regularly spaced time intervals to empirically approximate the expectation in the mean-field term (Messenger and Bortz, 2022; Della Maestra and Hoffmann, 2022; Yao et al., 2022; Della Maestra and Hoffmann, 2023). In Pavliotis and Zanoni (2022), the authors describe a method of moments estimator for the parameters of the MV-SDE. Other approaches concerned analyzing the partial differential equation (PDE) associated with MV-SDEs as in Gomes et al. (2019). In our work, we are primarily concerned with inference in regions where we have limited time-marginal data and the number of samples is not large.

In a machine learning context, Itô-SDEs have been extensively used for problems such as generative modeling (Song et al., 2020; De Bortoli et al., 2021). These models consider a forward SDE which maps data to noise and a reverse SDE which transforms noise to a target distribution, but the forward model notably does not include distributional dependence. Ansari et al. (2020) and Alvarez-Melis and Fusi (2021) consider using MV-SDEs for improving generative models and domain adaptation tasks, respectively.

Other applications of MV-SDEs in machine learning topics include estimating optimal trajectories in scRNA-Seq data (Chizat et al., 2022) and stochastic control problems relating to mean-field games (Han et al., 2022; Ruthotto et al., 2020). Inverse problems can also be solved by deriving an appropriate MV-SDE as the authors describe in Crucinio et al. (2022). Extensive analysis of the dynamics of the parameters of a neural network under stochastic gradient descent has been conducted using the theory from MV-SDEs, e.g. (Hu et al., 2021). These methods use a pre-described form of the drift to conduct their analyses whereas we're interested in learning a representation of the drift.

Our Contributions To address the lack of non-parametric MV-SDE estimators in the literature, this paper contributes the following: First, we present two neural architectures for representing MV-SDEs based on learned measures and generative networks; then, we present two estimators, based on maximum likelihood, used in conjunction with the architectures without prior knowledge on the structure of the drift; next, we characterize the properties of implicit regularization and richer probability flows of these architectures; finally, we empirically demonstrate the applicability of the architectures on time series and generative modeling.

2 Properties of MV-SDEs

We begin by describing the background and properties of the transition densities of MV-SDEs. Figure 2 illustrates some of these concepts qualitatively where we first consider non-local dynamics and then consider jumps in the sample paths.

2.1 Background

Consider a domain $\mathcal{D} \subset \mathbb{R}^d$ and let $\mathcal{P}_k(\mathcal{D})$ be the space of all probability distributions supported on $\mathcal{D}$ with finite kth moment. Let $W_t \in \mathbb{R}^d$ be a d-dimensional Wiener process and let $X_t \in \mathbb{R}^d$ be a solution to the following MV-SDE

$$\mathrm{d}X_t = b(X_t, p_t, t)\mathrm{d}t + \sqrt{\Sigma(X_t, p_t, t)}\mathrm{d}W_t \qquad (1)$$

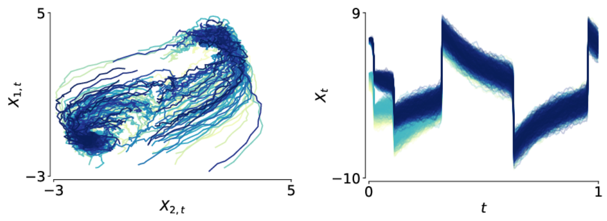


Figure 2: MV-SDE sample paths with non-local dynamics (left) and discontinuities (right).

where p_t denotes the law of X_t at time t and $\sqrt{\Sigma}$ denotes the Cholesky decomposition of Σ. We denote the *drift* as $b : \mathbb{R}^d \times \mathcal{P}_2(\mathcal{D}) \times \mathbb{R}_+ \to \mathbb{R}^d$ and the *diffusion* as $\Sigma : \mathbb{R}^d \times \mathcal{P}_2(\mathcal{D}) \times \mathbb{R}_+ \to \mathtt{SPD}(\mathbb{R}^{d\times d})$ with $\mathtt{SPD}$ denoting the space of symmetric, positive definite matrices.

We focus on the case where the diffusion coefficient is a known constant, σ, and focus on estimating the drift, b, from data. In addition, for simplicity in analysis, we suppose that b factors linearly into a non-interacting component, and an interacting component, where the mean-field term with dependence on p_t is often written in terms of an expectation, specifically

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \mathbb{E}_{y_t \sim p_t}\left[\varphi\left(X_t, y_t\right)\right]\mathrm{d}t + \sigma \mathrm{d}W_t \quad (2)$$

where $f : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ can be seen as the Itô drift, the expectation as the mean-field drift, and $\varphi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^k$ as the *interaction* function describing the interaction between particles, e.g. attraction with $\varphi(x, y) = -(x - y)$ in Figure 1b and the left side of Figure 2. This factorization is used in many practical models of interest, such as in bacterial chemotaxis, neural oscillators, and social networks (Carrillo et al., 2020). Note that the proposed architectures do not assume this factorization, we use this factorization for ease in exposition. As mentioned, unlike Itô-SDEs which only depend on the state of the current particle X_t and t, MV-SDEs additionally depend on the marginal time distribution of particles p_t. By introducing a dependence on the marginal law, the transition density of the process satisfies a richer class of functions. To motivate this further, we next describe the properties of non-local dynamics and jumps.

2.2 Non-locality of the transition density

Following the background, we describe a favorable property of the MV-SDE that induces non-local dependencies in the state space. The transition density of (2) can be written as the non-linear PDE

$$\partial_t p_t(x) = \\ -\nabla \cdot \left(\underbrace{p_t f(x)}_{\text{Itô Drift}} + \underbrace{p_t \int \varphi(x, y_t) p_t(y_t) \mathrm{d}y_t}_{\text{Non-Local Interactions}} - \underbrace{\frac{\sigma^2}{2} \nabla p_t}_{\text{Diffusion}} \right). \quad (3)$$

This non-local behavior has a variety of implications. For example, the distribution of particles "far away" from a reference particle can affect the behavior of the reference particle. This property is illustrated in the left side of Figure 2 with an example from the mean-field FitzHugh-Nagumo model used to model spikes in neuron activation, leading to interactions between distinct spikes (Crevat et al., 2019). Notably, this is not possible when considering only the Itô drift, since that operator acts locally on the density.

2.3 Discontinuous sample paths

The richer class of densities modeled by MV-SDEs has direct influence on individual sample paths. In a modeling scenario, we may wish to approximate a process that exhibits jumps. For example, in finance, a number of related entities may have common exposure and experience failure simultaneously (Nadtochiy and Shkolnikov, 2019; Feinstein and Søjmark, 2021). Similarly, in neuroscience, a number of neurons spiking simultaneously results in discontinuities in the sample paths (Carrillo et al., 2013). The fact that the interaction of many particles can cause blowups leads to a remarkable property of MV-SDEs that allows discontinuous paths. The major benefit of this property is that we do not need to consider an additional jump noise process – we only need to specify a particular interaction between the particles to induce the jump behavior. We can then represent point processes through the discontinuous points of the sample path, which provides a framework of interpreting a point process through particle interactions. A simple proof for the case of positive feedback is given in Hambly et al. (2019, Theorem 1.1). Intuitively, the jumps occur due to the drift inducing a simultaneous force against the particles, leading to the discontinuity.

3 Mean-field architectures

We first consider a modification of the architecture proposed in Pham and Warin (2022) that empirically computes the expectation using observations, and denote it as the empirical measure (EM) architecture. We then propose two architectures – an architecture based on representing a learned measure with neural weights,

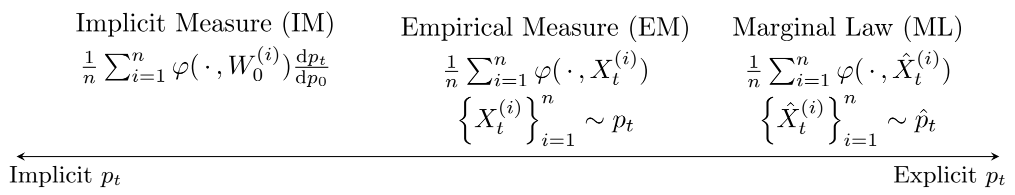

Implicit Measure (IM)
$\frac{1}{n}\sum_{i=1}^n \varphi(\,\cdot\,, W_0^{(i)})\frac{\mathrm{d}p_t}{\mathrm{d}p_0}$
Empirical Measure (EM)
$\frac{1}{n}\sum_{i=1}^n \varphi(\,\cdot\,, X_t^{(i)})$
$\left\{X_t^{(i)}\right\}_{i=1}^n \sim p_t$
Marginal Law (ML)
$\frac{1}{n}\sum_{i=1}^n \varphi(\,\cdot\,, \hat{X}_t^{(i)})$
$\left\{\hat{X}_t^{(i)}\right\}_{i=1}^n \sim \hat{p}_t$
Implicit p_t
Explicit p_t

Figure 3: Schematic comparing neural architectures for modeling MV-SDEs. Implicit measure (IM) architecture uses a mean-field layer that represents particles as learned weights and computes the expectation under a learned change of measure; the empirical measure (EM) architecture computes the expectation with the observed particles; the marginal law (ML) architecture learns the particle density, and computes an empirical expectation with samples from the learned density.

denoted as the implicit measure (IM) architecture, and a generative architecture based on representing the marginal law of the samples, denoted as the marginal law (ML) architecture. Figure 3 provides a schematic of the mean-field representations with different architectures. To motivate modeling the expectation in the mean-field term with the EM, IM and ML architectures, we first assume that the drift and diffusion coefficients of the MV-SDE in (1) exhibit sufficient regularity such that the empirical law converges to the true law of the system (i.e. $\frac{1}{n}\sum_{i=1}^n \delta_{X_t^{(i)}} \to_{n\to\infty} p_t(X_t)$), i.e. propagation of chaos holds (Méléard, 1996). We relax this assumption for the experiments since these are difficult to verify in practice.

3.1 Empirical measure architecture

Suppose we observe n particles at each time t given by $\{X_t^{(i)}\}_{i=1}^n$ and denote the discrete measure associated with these observations as $p_t^\delta = \frac{1}{n}\sum_{i=1}^n \delta_{X_t^{(i)}}$. Then, we can use p_t^δ to approximate the expectation in (2) as

$$\begin{aligned}\mathbb{E}_{y_t\sim p_t}\left[\varphi\left(X_t, y_t\right)\right] &\approx \mathbb{E}_{y_t\sim p_t^\delta}\left[\varphi\left(X_t, y_t;\theta\right)\right] \\ &= \frac{1}{n}\sum_{i=1}^n \varphi\left(X_t, X_t^{(i)};\theta\right) \qquad (4)\end{aligned}$$

for a neural network $\varphi(\cdot,\cdot;\theta)$ describing the interaction function between the particles (Pham and Warin, 2022). Suppose the non-mean field component f is also represented with a neural network $f(\cdot,t;\theta)$. Assuming that φ and f are well learned, this architecture can represent the MV-SDE drift in the limit as the number of observations $n\to\infty$. We refer to this architecture as the *empirical measure* (EM) architecture since at each time step the expectation is taken with respect to the empirical measure derived from the observations.

3.2 Implicit measure architecture

To empirically compute the expectation, the EM architecture in (4) relies on obtaining the empirical measure at each time point. This may be difficult in practice for a variety of reasons such as having few samples or the lack of data at some time points. Instead, in the implicit measure (IM) architecture, we represent particles as learned weights and compute the expectation under a learned change of measure.

Let us first recall that a single layer in a multilayer perceptron (MLP) can represent an expectation as

$$\mathrm{MLP}^{W,b}(x) = \int \sigma\left(Wx+b\right)\mathrm{d}\nu\left(W,b\right) \qquad (5)$$

where the expectation is taken over $\nu(\cdot)$, a measure over the space of parameters $y=(W,b)$, and σ is an activation function.

When $\nu = \frac{1}{n}\sum_{i=1}^n \delta_{y^{(i)}}$, a discrete measure with n particles, the expectation is exactly a single layer of width n, suggesting a correspondence between an empirical measure with n samples and a single layer of width n. Building on this correspondence, we propose a mean-field layer:

Definition 3.1 (Mean-field Layer)**.** Define the weight of the mean-field layer with width n as the matrix $W_0\in\mathbb{R}^{n\times d}$ and denote its ith row as $W_0^{(i)}$. The mean-field layer then is defined by the operation

$$\mathrm{MF}_{(n)}(\varphi(X_t)) := \frac{1}{n}\sum_{i=1}^n \varphi(X_t, W_0^{(i)})\frac{\mathrm{d}p_t}{\mathrm{d}p_0}. \qquad (6)$$

The mean-field layer (MF) can be thought of as another layer within the network architecture that approximates the law p_t. Each row $W_0^{(i)}$ is of size $\mathbb{R}^d$, corresponding to the dimensions of $X_t^{(i)}\in\mathbb{R}^d$. The activation function of the mean-field layer is the average over the augmented dimension over which MF operates. The change of measure $\frac{\mathrm{d}p_t}{\mathrm{d}p_0}$, such that $\mathbb{E}_{p_t}[\,\cdot\,] = \mathbb{E}_{p_0}[\,\cdot\,\frac{\mathrm{d}p_t}{\mathrm{d}p_0}]$, can be learned as part of the estimator of the interaction function, $\varphi(\cdot,\cdot,t;\theta)$. Thus

$$\mathbb{E}_{y_t\sim p_t}\left[\varphi\left(X_t, y_t\right)\right] \approx \frac{1}{n}\sum_{i=1}^n \varphi(X_t, W_0^{(i)}, t;\theta).$$

Importantly, the above representation allows modeling mean-field interactions without the need for a full set of observations at each time point and without the need to explicitly represent the distribution p_t at each time point. Assuming that φ and MF are well learned, this architecture can represent the true MV-SDE drift in the limit as the width $n \to \infty$. We note empirically that a finite n is sufficient and we provide examples of ablations in the appendix.

A similar analysis can be made for the standard MLP architecture. However, the explicit separation of f and φ is not expressed and φ may take a simpler convolutional form. This leads us to the following remark:

Remark 3.2. *From the above discussion, the expectation of the form $\mathbb{E}_{y\sim p_t}[\varphi(X_t - y)]$ with respect to the law p_t may be implicitly represented by a MLP.*

Our motivation is then concerned with how a relatively more explicit distribution dependence with φ and MF affect modeling capabilities. This explicit structure lends to an implicit regularization that promotes a smaller norm of the mean-field component under a maximum likelihood estimation framework, which we detail later in Section 5.1.

3.3 Marginal law architecture

In the marginal law (ML) architecture, in conjunction with the drift, we introduce a generative model for representing the time-varying density. In this case, we approximate the expectation in (2) as

$$\begin{aligned}\mathbb{E}_{y_t\sim p_t}\left[\varphi\left(X_t, y_t\right)\right] &\approx \mathbb{E}_{y_t\sim \hat{p}_t}\left[\varphi\left(X_t, y_t; \theta\right)\right]\\ &= \frac{1}{n}\sum_{i=1}^{n}\varphi\left(X_t, \hat{X}_t^{(i)}; \theta\right) \qquad (7)\end{aligned}$$

where the expectation is taken with respect to the discrete measure derived from samples $\{\hat{X}_t^{(i)}\}_{i=1}^n$ from the generative model $\hat{p}_t(\cdot;\phi)$. In addition, we regularize $\hat{p}_t(\cdot;\phi)$ such that the marginal law is consistent with the flow relating to the drift, which we detail later in Section 4.3 using the PDE in (3).

4 Parameter estimation

Having presented the neural architectures, we now present estimators, based on maximum likelihood, used in conjunction with the architectures without prior knowledge on the structure the drift. We first describe the likelihood function for use in cases with regularly sampled data. We then describe a bridge estimator for cases of irregularly sampled data. In addition, we describe an estimator for the generative architecture based on both the likelihood function and the transition density. For this section, we assume that we observe multiple paths, i.e., $\left\{\{X_{t_j}\}_{j=1\dots K}^{(i)}\right\}_{i=1\dots N}$. Full details of all algorithms are in the appendix.

4.1 Maximum likelihood estimation

We use an estimator based on the path-wise likelihood derived from Girsanov's theorem and an Euler-Maruyama discretization for the likelihood, considered in Sharrock et al. (2021). The likelihood function is given as

$$\begin{aligned}\mathcal{L}(\theta; t_1, t_K) := \exp\Bigg(&\frac{1}{\sigma^2}\int_{t_1}^{t_K} b\left(X_s, p_s, s;\theta\right)\mathrm{d}X_s\\ &-\frac{1}{2\sigma^2}\int_{t_1}^{t_K} b\left(X_s, p_s, s;\theta\right)^2\mathrm{d}s\Bigg),\end{aligned} \qquad (8)$$

where b is the unknown drift represented as one of the presented architectures and σ is the diffusion coefficient in (1) and (2). Following discretization, with the approximations $\Delta X_{t_j} = X_{t_{j+1}} - X_{t_j}$ and $\Delta t_j = t_{j+1} - t_j$, the log-likelihood is approximated by

$$\begin{aligned}\log \mathcal{L}(\theta; t_1, t_K) \approx &\sum_{j=1}^{K-1} b\left(X_{t_j}, p_{t_j}, t_j;\theta\right)\left(X_{t_{j+1}} - X_{t_j}\right)\\ &-\frac{1}{2}\sum_{j=1}^{K-1} b\left(X_{t_j}, p_{t_j}, t_j;\theta\right)^2\left(t_{j+1} - t_j\right).\end{aligned}$$

Optimization is performed using standard gradient based optimizers with the drift b represented as one of the presented architectures.

4.2 Estimation with Brownian bridges

Often data are not collected at uniform intervals in time, but rather, the time marginals may be collected at irregular intervals. In that case, we consider an interpolation approach to maximizing the likelihood building on the results of Lavenant et al. (2021) and Cameron et al. (2021) in the Itô-SDE case. We can write the likelihood conditioned on the set of observations (dropping the particle index for ease of notation) as

$$\mathcal{L}_{BB}(\theta) = \mathbb{E}_{\mathbb{Q}}\left[\prod_{j=1\dots K-1}\mathbb{1}_{\{Z_{t_{j+1}} = X_{t_{j+1}}\}}\mathcal{L}(\theta; t_j, t_{j+1})\right]$$

where $\{Z_s : s\in[t_j, t_{j+1}]\}$ is a Brownian bridge from X_{t_j} to $X_{t_{j+1}}$ and $\mathbb{Q}$ is the Wiener measure. Brownian bridges can easily be sampled and reused for computing the expectation.By applying Jensen's inequality, we can write an evidence lower bound (ELBO) as

$$\log \mathcal{L}_{BB} \geq \mathbb{E}_{\mathbb{Q}}\left[\sum_{j=1\dots K-1} \log \mathcal{L}(\theta; t_j, t_{j+1}) \mathbb{1}_{\left\{Z_{t_j}=X_{t_j}\right\}_{j=1}^{K}}\right]. \tag{9}$$

In this case, the estimator aims to fit the observed marginal distributions exactly while penalizing deviations from the Brownian bridge paths in regions without data.

4.3 Estimation with explicit marginal law

Returning to the ML architecture described in Section 3.3, where we explicitly model the density p_t with a generative network $\hat{p}_t$, our estimator should enforce the consistency between $\hat{p}_t$ and the flow relating to the drift. We do so using the PDE in (3). Let the parameters of the drift be θ and the parameters of the generative model be ϕ, we solve the optimization problem

$$\max_{\theta,\phi} \quad \mathbb{E}\left[\mathcal{L}(\theta, \phi \mid \{X_{t_j}\}_{j=1\dots K})\right] \quad s.t. \tag{10}$$

$$\int_{t_j}^{t_{j+1}} \left|\hat{p}_s(x;\phi) - \mathbb{E}\left[\hat{p}_{t_{j+1}}\left(\hat{X}_{t_{j+1}};\phi\right) \mid \hat{X}_s = x\right]\right| \mathrm{d}s = 0 \tag{11}$$

for time intervals indexed by $j = 1, \dots, K-1$, state space $x \in \mathrm{supp}(X_t)$, and where the trajectories of $\hat{X}_t$ are given by the dynamics of the ML architecture, specifically

$$\mathrm{d}\hat{X}_t = f(\hat{X}_t, t;\theta)\mathrm{d}t + \mathbb{E}_{y_t\sim\hat{p}_t(\cdot;\phi)}\left[\varphi\left(\hat{X}_t, y_t;\theta\right)\right]\mathrm{d}t + \sigma \mathrm{d}W_t. \tag{12}$$

The likelihood at the observed margins is first maximized in (10). In (11), the marginals at previous times are regularized using the correspondence between the PDE and its associated SDE via the nonlinear Kolmogorov backwards equation (Buckdahn et al., 2017), which describes p_t as an expectation of trajectories at a terminal time, i.e. $p_t(x) = \mathbb{E}[p_T(X_T) \mid X_t = x]$ for $t < T$.

5 Modeling properties

Having discussed the architectures and estimators, we now discuss specific properties of the modeling framework, which follow from the theoretical discussion presented in Section 2. We first discuss how the factorization into φ and MF lends to an implicit regularization of the IM architecture. We then compare the gradient flows of Itô-SDEs and MV-SDEs. We additionally provide more intuition on the proposed architectures in Appendix D.

5.1 Implicit regularization of the implicit measure architecture

Closely related to the IM architecture are MLP representations of Itô-SDEs, where we previously remarked may model MV-SDEs. On the other hand, the factorization of the IM architecture into an interaction function and a learned measure leads to a type of implicit regularization when the parameters are estimated using gradient descent.

Theorem 5.1 (Implicit Regularization). *Suppose f and φ are known and fixed. Further, assume that φ is twice differentiable. Then, for each time step t, the minimizing finite width MF with weight matrix $W_0 \in \mathbb{R}^{n\times d}$ and ith row $W_0^{(i)}$ under gradient descent satisfies the following optimization problem*

$$\min_{W_0} \quad \sum_{i=1\dots n}\sum_{j=1\dots d} \varphi(X_t, W_0^{(i)})_j$$
$$\text{s.t.} \quad \mathbb{E}\left[\frac{1}{2\Delta t}\left\|X_{t+\Delta t} - X_t - b(X_t, p_t, t)\right\|^2\right] = 0.$$

Proof. We follow the blueprint in Belabbas (2020) and give full details in the appendix. □

Theorem 5.1 effectively says that the mean-field system approximated is the one that has the least influence from the other particles. In the case where φ can be decomposed as a norm, this amounts to finding the drift parameterized by weight W_0 with smallest norm while still matching the marginals.

5.2 Gradient flows of the MV-SDE

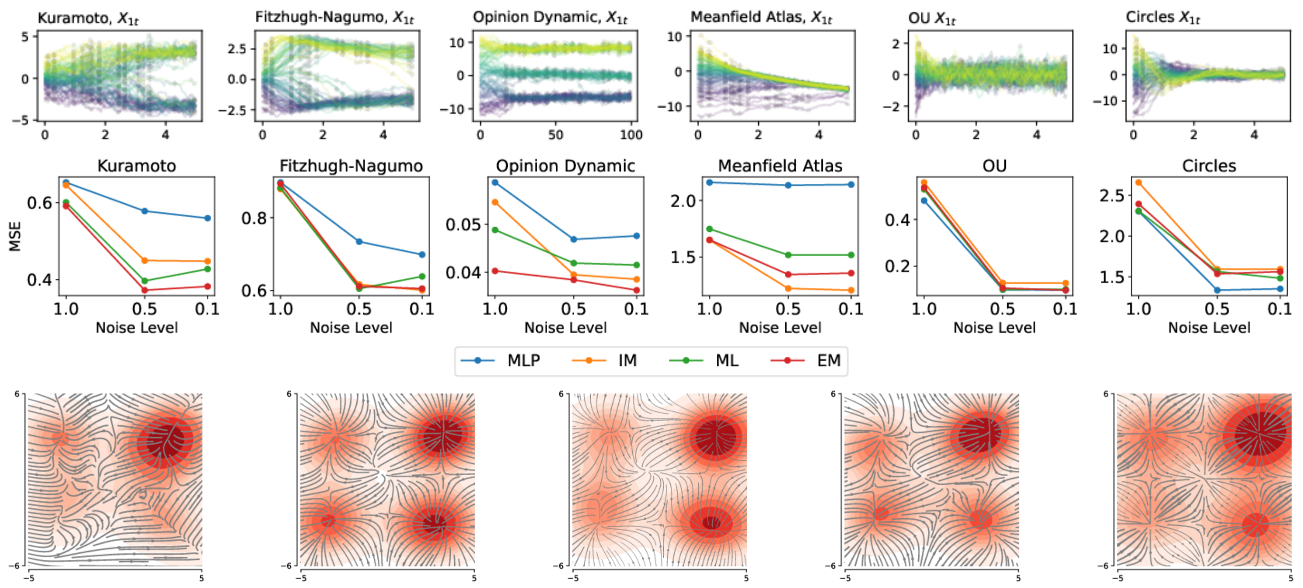
To illustrate the difference between the particle flows of MV-SDEs and Itô-SDEs, we consider a gradient flow perspective to describe the functionals that are minimized according to the different SDEs (Villani, 2021, Section 8.3). In the following remark, we apply this idea to a gradient flow that minimizes the energy distance.

Remark 5.2 (Minimizing the Energy Distance). *Consider two densities p_t, q such that $p_t \ll q$ for all t. The gradient flow induced by the MV-SDE with the drift*

$$b(X_t, p_t, t) = \mathbb{E}_{y_t\sim p_t}\left[\nabla\left(2\|X_t - y_t\|\frac{\mathrm{d}q}{\mathrm{d}p_t} - \|X_t - y_t\|^2 - \|X_t - y_t\|^2\left(\frac{\mathrm{d}q}{\mathrm{d}p_t}\right)^2\right)\right]$$

minimizes the energy distance between p_t and q.

The proof follows from a straightforward application of Santambrogio (2017, Section 4). Note that this construction is only possible through distributional



(a) MLP (b) EM (c) IM (d) ML (e) Truth

Figure 4: Top row: sample paths from the different synthetic datasets. Middle row: mean squared error (MSE) of different architectures' performance (average of 10 runs) on drift estimation, under the effect of different levels of observation noise. Bottom row: Example of estimated gradient flow of Kuramoto model at terminal time. The colors correspond to the density of generated samples at terminal time.

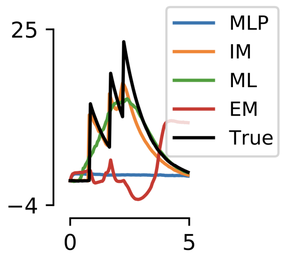
dependence in MV-SDE whereas the standard Itô SDE does not satisfy this drift. This has particular impact on generative modeling which we will later discuss in our experiment section.

6 Numerical experiments

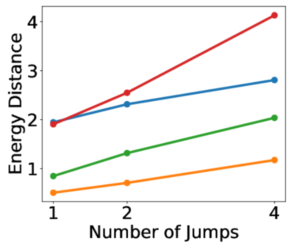
We discussed *Q1* on modeling and inferring distributional dependence. We now wish to answer *Q2* and quantify the effect of explicit distributional dependence in machine learning tasks. We test the methods on synthetic and real data for time series and generative modeling. The main goal is to determine the difference between standard Neural Itô-SDE and the proposed Neural MV-SDEs under different modeling scenarios. In that sense, the baseline we consider is the Itô-SDE parameterized using an MLP. However, we also consider other deep learning based methods for comparison in a broader context. We abbreviate the different architectures as the Neural Itô-SDE (MLP) and Neural MV-SDEs: Empirical Measure (EM), Implicit Measure (IM) and Marginal Law (ML) architectures. These architectures were presented in Section 3 and summarized in Figure 3. Full descriptions of the models, baselines, and datasets are given in the appendix.

Synthetic data experiments Motivated by the application of MV-SDEs in physical, biological, social, and financial settings, we benchmark the proposed methods on 4 canonical MV-SDEs: the Kuramoto model which describes synchronizing oscillators (Sonnenschein and Schimansky-Geier, 2013), the mean-field FitzHugh-Nagumo model which characterizes spikes in neuron activations (Mischler et al., 2016), the opinion dynamic model on the formation of opinion groups (Sharrock et al., 2021), and the mean-field atlas model for pricing equity markets (Jourdain and Reygner, 2015). We additionally benchmark the proposed methods on two Itô-SDEs: an Ornstein–Uhlenbeck (OU) process and a circular motion equation to determine the performance on Itô-SDEs. Finally, to understand the performance on discontinuous paths, we benchmark the proposed methods on an OU process with jumps. We focus on recovering the drift from observations.

Since the true drifts of the synthetic data are known, we directly compare the estimated drifts to the true drifts using mean squared error (MSE). The performance on five different datasets with three different levels of added observational noise is presented in Figure 4. The bottom row of Figure 4 illustrates an example of the density and the gradient flow at the terminal time for the Kuramoto model. The proposed mean-field architectures outperform the standard MLP in modeling MV-SDEs; moreover, incorporating explicit distributional depedence does not diminish the performance in estimating Itô-SDEs. When modeling processes with jump discontinuities, Figure 5b highlights the flexibility of the proposed methods, IM, ML, to match such



(a) Average paths.



(b) Energy distance.

Figure 5: Results for approximating sample paths containing jumps.

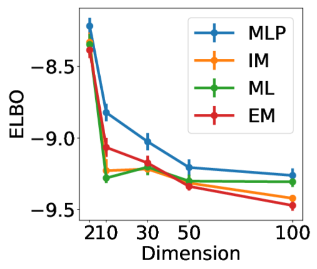


Figure 6: ELBO of generated paths from standard Gaussian to eight Gaussian mixture (in increasing dimension) evaluated against OT mapping.

models. The EM likely does not perform as well due to the high variance of the empirical measure, leading to difficulties in learning. Additionally, the MLP does not have an explicit decomposition between the MV and Itô components, resulting in issues when estimating the feedback between the particles inducing jumps.

Real data experiments We consider two real examples: crowd trajectory in an open interacting environment, which is related to the Cucker-Smale model (Cucker and Smale, 2007; Warren, 2018) and chemically stimulated movement of organisms (chemotaxis), which can be modeled by the Keller-Segel model (Tomašević, 2021; Keller and Segel, 1971).

We evaluate the proposed architectures in these modeling tasks by comparing the goodness-of-fit of generated path samples to the observed path samples, measured in normalized MSE (normalized with sample variance) with respect to the held out data. We also benchmark against the DeepAR probabilistic time series forecaster (Salinas et al., 2020) with RNN, GRU, LSTM, and Transformer (TR) backbones as another baseline model to compare the goodness-of-fit.

The performances of different architectures are presented in Table 1. For the crowd trajectory data, the proposed MV-SDE architectures outperform the MLP, EM and DeepAR architectures for forecasting. Notably, the EM architecture exhibits high variance in the crowd trajectory data, indicating the difficulty of relying on the empirical margins to compute expectations. For chemotaxis data, the MV-SDE based architectures outperform the DeepAR baselines. Additional experiments and results are presented in appendix.

Generative modeling experiments We focus on applying the bridge estimator discussed in Section 4.2 to map between a Gaussian and a target distribution for the purposes of generative modeling. We are interested in two aspects: 1) the properties of the learned mapping, and 2) the generated trajectories. We first study the properties of the learned mapping using a synthetic eight Gaussian mixture with increasing dimensionality. We compare the performance of different architectures by evaluating the ELBO of the sample paths generated by the optimal transport (OT) mapping between the initial distribution and held out target samples. We next evaluate the generated trajectories through the energy distance as motivated in Remark 5.2 between generated and held-out data for 5 real data density estimation experiments. In addition, we compare to common density estimators of variational autoencoder (VAE) (Kingma and Welling, 2013), Wasserstein generative adversarial network (WGAN) (Gulrajani et al., 2017), masked autoregresive flow (MAF) (Papamakarios et al., 2017) and score-based generative modeling through SDEs, which corresponds to a constrained form of the MLP architecture (Song et al., 2020).

The MV-SDE architectures not only outperform the Itô architecture for all dimensions in the eight Gaussian experiment, as shown in Figure 6, but also for the 5 real data density estimation experiments, as shown in Table 2, while outperforming common baselines. Sampling is performed using standard Euler-Maruyama, with full details in the appendix. This again suggests the MV-SDE provides a more amenable probability flow for generative modeling compared with the Itô-SDE.

7 Discussion

In this paper we discuss an alternative viewpoint of diffusion type models beyond the standard Itô-SDE parameterization. In particular, we focus on MV-SDEs and discuss neural representations of a process that depends on the distribution, and ways of making this dependence more explicit. We demonstrated the efficacy of the proposed architectures on a number of synthetic and real benchmarks. The results suggest that the proposed architectures provide an improvement in certain time series and generative modeling applications.

Limitations We studied the implicit regularization of the IM architecture under gradient descent, and the

Table 1: Time series estimation on held out trajectories. Values in **bold** and *italic* are best and second best.

	Crowd Traj	C.Cres	E.Coli
MLP (Itô)	0.068 (0.03)	0.096 (0.002)	*0.080* (0.003)
IM	*0.034* (0.01)	0.094 (0.003)	**0.080** (0.001)
ML	**0.016** (0.01)	**0.093** (0.002)	0.084 (0.002)
EM	0.091 (0.059)	*0.093* (0.004)	0.086 (0.004)
LSTM	1.408 (0.92)	1.159 (0.234)	0.585 (0.350)
RNN	1.05 (0.54)	1.563 (1.070)	0.773 (0.092)
GRU	1.339 (0.61)	0.826 (0.289)	0.568 (0.301)
TR	2.732 (0.88)	1.503 (0.212)	1.204 (0.212)

Table 2: Density estimation: energy distance between observed samples and generated samples. Values in **bold** and *italic* are best and second best.

	POW	MINI	HEP	GAS	CORT
MLP (Itô)	0.34 (0.1)	0.67 (0.05)	0.54 (0.05)	0.41 (0.08)	0.74 (0.06)
IM	0.29 (0.08)	**0.40** (0.0)	0.41 (0.03)	**0.29** (0.08)	**0.53** (0.03)
ML	**0.28** (0.08)	*0.44* (0.03)	*0.37* (0.03)	0.31 (0.06)	0.57 (0.03)
EM	0.33 (0.1)	0.46 (0.04)	0.43 (0.05)	*0.30* (0.03)	0.58 (0.037)
VAE	1.2 (0.02)	2.1 (0.15)	1.8 (0.03)	1.5 (0.02)	2.4 (0.2)
WGAN	1.2 (0.02)	2.1 (0.003)	1.8 (0.01)	1.3 (0.02)	2.2 (0.01)
MAF	*0.29* (0.04)	0.48 (0.01)	**0.31** (0.02)	0.52 (0.03)	*0.53* (0.03)
Score	0.30 (0.05)	0.50 (0.02)	0.32 (0.03)	0.56 (0.04)	0.58 (0.02)

extension of the analysis to the other proposed architectures is important to understand the corresponding regularization. With regard to computing expectations, using a multilevel scheme (Szpruch et al., 2019) could improve accuracy while reducing computational cost.

Future directions The proposed architectures provide a baseline to extend the work to estimation of alternative processes. Heterogeneity amongst the particles is a useful property in many types of systems, e.g. described in Lacker and Soret (2022). Extending the W_0 architectures to the case of heterogeneous agents corresponds to introducing depth into the architecture (i.e. having multiple measures W_0 to take the expectation with respect to). Additionally, solving inverse problems using Wasserstein gradient flows solved using MV-SDEs can be another application of the proposed methods (Crucinio et al., 2022). Interpreting the W_0 architecture through the interpolation lens used in Szpruch et al. (2019) could also provide avenues for improvement of the architecture. Finally, establishing convergence rates for the architectures such as the W_0 or X_t architectures would possibly be a direction for further analysis on the proposed algorithms.

Acknowledgements

This work was supported in part by the Air Force Office of Scientific Research under award number FA9550-20-1-0397. AH was partially supported by an NSF Graduate Research Fellowship.

References

David Alvarez-Melis and Nicolò Fusi. Dataset dynamics via gradient flows in probability space. In *International Conference on Machine Learning*, pages 219–230. PMLR, 2021.

Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. *arXiv preprint arXiv:2012.00780*, 2020.

Mohamed Ali Belabbas. On implicit regularization: Morse functions and applications to matrix factorization. *arXiv preprint arXiv:2001.04264*, 2020.

Rainer Buckdahn, Juan Li, Shige Peng, and Catherine Rainer. Mean-field stochastic differential equations and associated pdes. *The Annals of Probability*, 45 (2):824–878, 2017. ISSN 00911798, 2168894X. URL `http://www.jstor.org/stable/44245559`.

Scott Cameron, Tyron Cameron, Arnu Pretorius, and Stephen Roberts. Robust and scalable sde learning: A functional perspective. *arXiv preprint arXiv:2110.05167*, 2021.

René Carmona, François Delarue, et al. *Probabilistic theory of mean field games with applications I-II*. Springer, 2018.

JA Carrillo, RS Gvalani, GA Pavliotis, and A Schlichting. Long-time behaviour and phase transitions for the mckean–vlasov equation on the torus. *Archive for Rational Mechanics and Analysis*, 235(1):635–690, 2020.

José A Carrillo, María d M González, Maria P Gualdani, and Maria E Schonbek. Classical solutions for a nonlinear fokker-planck equation arising in computational neuroscience. *Communications in Partial Differential Equations*, 38(3):385–409, 2013.

Xiaoli Chen, Liu Yang, Jinqiao Duan, and George Em Karniadakis. Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43 (3):B811–B830, 2021.

Lénaïc Chizat, Stephen Zhang, Matthieu Heitz, and Geoffrey Schiebinger. Trajectory inference via mean-field langevin in path space. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=Mftcm8i4sL`.

Joachim Crevat, Grégory Faye, and Francis Filbet. Rigorous derivation of the nonlocal reaction-diffusion fitzhugh–nagumo system. *SIAM Journal on Mathematical Analysis*, 51(1):346–373, 2019.

Francesca R Crucinio, Valentin De Bortoli, Arnaud Doucet, and Adam M Johansen. Solving fredholm integral equations of the first kind via wasserstein gradient flows. *arXiv preprint arXiv:2209.09936*, 2022.

Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on automatic control*, 52 (5):852–862, 2007.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Laetitia Della Maestra and Marc Hoffmann. Nonparametric estimation for interacting particle systems: Mckean–vlasov models. *Probability Theory and Related Fields*, 182(1):551–613, 2022.

Laetitia Della Maestra and Marc Hoffmann. The lan property for mckean-vlasov models in a mean-field regime. *Stochastic Processes and their Applications*, 155:109–146, 2023. ISSN 0304-4149. doi: https://doi.org/10.1016/j.spa.2022.10.002. URL `https://www.sciencedirect.com/science/article/pii/S0304414922002113`.

Kai Du, Yifan Jiang, and Jinfeng Li. Empirical approximation to invariant measures for mckean–vlasov processes: mean-field interaction vs self-interaction. *arXiv preprint arXiv:2112.14112*, 2021.

Zachary Feinstein and Andreas Søjmark. Dynamic default contagion in heterogeneous interbank systems. *SIAM Journal on Financial Mathematics*, 12 (4):SC83–SC97, 2021.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL `https://doi.org/10.1198/016214506000001437`.

Susana N Gomes, Andrew M Stuart, and Marie-Therese Wolfram. Parameter estimation for macroscopic pedestrian dynamics models from microscopic data. *SIAM Journal on Applied Mathematics*, 79(4):1475–1500, 2019.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJxgknCcK7`.

Marianne Grognot and Katja M Taute. A multiscale 3d chemotaxis assay reveals bacterial navigation mechanisms. *Communications biology*, 4(1):1–8, 2021.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Ben Hambly, Sean Ledger, and Andreas Søjmark. A mckean–vlasov equation with positive feedback and blow-ups. *The Annals of Applied Probability*, 29(4): 2338–2373, 2019.

Jiequn Han, Ruimeng Hu, and Jihao Long. Learning high-dimensional mckean-vlasov forward-backward stochastic differential equations with general distribution dependence. *arXiv preprint arXiv:2204.11924*, 2022.

Ali Hasan, Joao M Pereira, Sina Farsiu, and Vahid Tarokh. Identifying latent stochastic differential equations. *IEEE Transactions on Signal Processing*, 70: 89–104, 2021.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Kaitong Hu, Zhenjie Ren, David Šiška, and Łukasz Szpruch. Mean-field langevin dynamics and energy landscape of neural networks. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 57, pages 2043–2065. Institut Henri Poincaré, 2021.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34: 22863–22876, 2021.

Benjamin Jourdain and Julien Reygner. Capital distribution and portfolio performance in the mean-field atlas model. *Annals of Finance*, 11(2):151–198, 2015.

Evelyn F. Keller and Lee A. Segel. Model for chemotaxis. *Journal of Theoretical Biology*, 30(2):225–234, 1971.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018. URL `https://doi.org/10.48550/arXiv.1807.03039`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7386–7400, 2021.

Daniel Lacker. Mean field games and interacting particle systems. 2018. URL `http://www.columbia.edu/~dl3133/MFGSpring2018.pdf`.

Daniel Lacker and Agathe Soret. A case study on stochastic games on large graphs in mean field and sparse regimes. *Mathematics of Operations Research*, 47(2):1530–1565, 2022.

Hugo Lavenant, Stephen Zhang, Young-Heon Kim, and Geoffrey Schiebinger. Towards a mathematical theory of trajectory inference. *arXiv preprint arXiv:2102.09204*, 2021.

Thomas M. Liggett. Stochastic models of interacting systems. *The Annals of Probability*, 25(1):1 – 29, 1997. doi: 10.1214/aop/1024404276. URL `https://doi.org/10.1214/aop/1024404276`.

Sylvie Méléard. Asymptotic behaviour of some interacting particle systems; mckean-vlasov and boltzmann models. *Probabilistic models for nonlinear partial differential equations*, pages 42–95, 1996.

Daniel A. Messenger and David M. Bortz. Learning mean-field equations from particle data using wsindy. *Physica D: Nonlinear Phenomena*, 439:133406, 2022. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2022.133406. URL `https://www.sciencedirect.com/science/article/pii/S0167278922001543`.

Stéphane Mischler, Cristóbal Quininao, and Jonathan Touboul. On a kinetic fitzhugh–nagumo model of neuronal network. *Communications in mathematical physics*, 342(3):1001–1042, 2016.

Sergey Nadtochiy and Mykhaylo Shkolnikov. Particle systems with singular interaction through hitting times: application in systemic risk modeling. *The Annals of Applied Probability*, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Grigorios A Pavliotis and Andrea Zanoni. A method of moments estimator for interacting particle systems and their mean field limit. *arXiv preprint arXiv:2212.00403*, 2022.

Huyên Pham and Xavier Warin. Mean-field neural networks: learning mappings on wasserstein space. *arXiv preprint arXiv:2210.15179*, 2022.

Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017.

Louis Sharrock, Nikolas Kantas, Panos Parpas, and Grigorios A Pavliotis. Parameter estimation for the mckean-vlasov stochastic differential equation. *arXiv preprint arXiv:2106.13751*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Bernard Sonnenschein and Lutz Schimansky-Geier. Approximate solution to the stochastic kuramoto model. *Physical Review E*, 88(5), nov 2013.

Lukasz Szpruch, Shuren Tan, and Alvin Tse. Iterative multilevel particle approximation for mckean–vlasov sdes. *The Annals of Applied Probability*, 29(4):2230–2265, 2019.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34: 24804–24816, 2021.

Milica Tomašević. A new mckean–vlasov stochastic interpretation of the parabolic-parabolic keller–segel model: The two-dimensional case. *The Annals of Applied Probability*, 31(1):432–459, 2021.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

William H Warren. Collective motion in human crowds. *Current directions in psychological science*, 27(4):232–240, 2018.

Rentian Yao, Xiaohui Chen, and Yun Yang. Mean-field nonparametric estimation of interacting particle systems. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 2242–2275. PMLR, 02–05 Jul 2022. URL `https://proceedings.mlr.press/v178/yao22a.html`.

Xiao Lei Zhang, Henri Begleiter, Bernice Porjesz, Wenyu Wang, and Ann Litke. Event related potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] See the supplementary materials for all code.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] See the assumptions in the Appendix.
   (b) Complete proofs of all theoretical results. [Yes] See the full proofs in the Appendix.
   (c) Clear explanations of any assumptions. [Yes] Most assumptions just require propagation of chaos to hold, which should hold when considering McKean-Vlasov processes.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] See the supplemental material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] See the Appendix.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] See the Appendix.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] See the Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] All citations available in the main text.
   (b) The license information of the assets, if applicable. [Yes] All datasets are publicly available.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Neural McKean-Vlasov Processes: Supplementary Materials

A Proofs

In the main text, we briefly discussed some theoretical advantages in terms of the flexibility in the time marginal distributions that MV-SDEs can represent versus Itô-SDEs, such as non-local dynamics and jumps in the sample paths. For more background and properties, we refer to the notes by Lacker (2018) and the book by Carmona et al. (2018).

In this section, we begin by considering the theoretical advantages of the proposed architectures and estimators. Specific to the architectures, we develop the implicit measure architecture, with a comparison to attention. In addition, we study the implicit regularization of explicit distributional dependence, with a comparison to optimal transport. Specific to the estimators, we develop the compatibility criterion for the modeled density to be consistent with the flow of the modeled SDE, and discuss a similar interpretation for the interpolation approach of the Brownian bridge estimator.

A.1 Development of implicit measure architecture

The implicit measure (IM) architecture is motivated by the fact that given a drift b that is Lipschitz continuous, by the universal approximation theorem, a two-layer multi-layer perceptron (MLP) can approximate b to arbitrary precision (Hornik, 1991). We first show that the drift of a MV-SDE may be represented by a MLP then describe the IM architecture where the distributional dependence is made more apparent.

Proof. Consider a McKean-Vlasov process where the drift b is factorized into a linear form

$$b(X_t, p_t, t) = f(X_t, t) + \mathbb{E}_{y_t \sim p_t}[\varphi(X_t - y_t)]$$

and assume that $f(\cdot;\theta)$ and $\varphi(\cdot;\theta)$ are well approximated by MLPs following the universal approximation theorem. It remains to show that $\mathbb{E}_{y_t \sim p_t}[\varphi(X_t - y_t)]$ can be well approximated by an MLP. We will begin by presenting the proof for the case where the law is stationary, then perform a change of measure to extend to the case where the law is non-stationary.

Recall that a MLP can be written in terms of an expectation as

$$\begin{aligned}\mathrm{MLP}^{W,b}(x) &= \int \sigma\left(Wx + b\right) \mathrm{d}\nu\left(W, b\right) \\ &= \mathbb{E}[\sigma(Wx + b)]\end{aligned}$$

where the expectation is taken over $\nu(\cdot)$, a measure over the space of parameters W, b, and σ is an activation function. By our original argument that φ is well approximated by a MLP, we can let that represent the activation function. Next, set $\nu(W) = \delta_{I_d}$ and $\nu(b) = \mathrm{Law}(-X_t)$. Since we assumed X_t is stationary, $\mathrm{Law}(X_t) = \mathrm{Law}(X_\star)$ for all t. We now have our approximation as

$$\begin{aligned}\mathrm{MLP}^{W,b}(x) &= \int \varphi\left(x - b\right) p_t(b) \mathrm{d}b \\ &= \mathbb{E}_{y \sim p_t}[\varphi(x - y)].\end{aligned}$$

Non-stationary law Next we consider the case where the law of X_t is not the same for all t. For this argument, we will consider the change of measure that maps p_t to q. Since we are assuming that the diffusion is constant, all measures p_t are absolutely continuous with respect to each other. We additionally assume that Novikov's condition is satisfied. Following Girsanov's theorem, we can write the expectation in terms of this changed measure by introducing the factor $\frac{\mathrm{dp_t}}{\mathrm{dq}}$

$$\mathbb{E}_{y\sim p_t}[\varphi(x-y)] = \mathbb{E}_{y\sim q}\left[\varphi(x-y)\frac{\mathrm{d}p_t}{\mathrm{d}q}\right].$$

Under this formulation q is the learned measure and p_t is the measure at each time point t. Assuming that the function $\varphi(\cdot;\theta)\frac{\mathrm{d}p_t}{\mathrm{d}q}$ can be learned for all t as another MLP $\tilde{\varphi}(\cdot, t; \theta)$, we conclude the proof. A similar idea was explored in Du et al. (2021) where the authors attempt to compute a stationary measure as a change of measure of particle samples.

□

Following a similar notation to the MLP proof, we only change the base measure such that it is given by the mean-field layer. A similar change of measure argument is then applied to complete the development of the IM architecture.

The proposed neural architectures differ from existing methods that consider the empirical measure, since we consider parameters to describe the measure at different time points. The proposed neural architectures also differ from existing methods that describe Ito-SDEs since we consider a more explicit parameterization of distributional dependence and a more general interaction function, $\varphi(x,y)$ instead of $\varphi(x-y)$.

Relationship to attention Recently, works such as Sander et al. (2022) described the relationship between interacting particle systems and the attention structure in the transformer architecture. Here we briefly describe a motivation for using the proposed architectures in the sense that they describe a similar structure to attention.

Recall that the attention module is defined by matrices $W_K, W_Q \in \mathbb{R}^{N_W\times d}, W_V \in \mathbb{R}^{N_V\times d}$ and the normalized attention matrix by

$$\alpha_{i,j} = N\exp(\langle W_K X^{(i)}, W_Q X^{(j)}\rangle)\Big/\sum_{k=1}^{N}\exp(\langle W_K X^{(i)}, W_Q X^{(k)}\rangle).$$

We focus on the attention matrix since it describes the dependence between particles $X^{(i)}$.

We can rewrite the above equation as an expectation

$$\alpha_{i,j} = \exp(\langle W_K X^{(i)}, W_Q X^{(j)}\rangle)\Big/\mathbb{E}[\exp(\langle W_K X^{(i)}, W_Q y\rangle)],$$

where the expectation is taken with respect to a discrete measure $\nu = \sum_{k=1}^{N}\delta_{X^{(k)}}$, as we do in the IM architecture. We can write the numerator as the expectation with an indicator and the denominator as the full expectation,

$$\alpha_{i,j} = \mathbb{E}[\exp(\langle W_K X^{(i)}, W_Q y\rangle)\mathbb{1}_{y=X^{(j)}}]\Big/\mathbb{E}[\exp(\langle W_K X^{(i)}, W_Q y\rangle)].$$

Finally, since we do not assume a particular structure on φ in the IM architecture, we can let φ be equal to the exponential of the dot product with the transformation by W_K, W_Q. Note that this is applied to particles at each time marginal t rather than for a sequence of particles. A sequence of particles would correspond to the case of non-exchangability, which is a direction of future work.

A.2 Implicit regularization of explicit distributional dependence

Proof. Consider a McKean-Vlasov process governed by

$$\mathrm{d}X_t = \{f(X_t,t) + \mathbb{E}_{y_t\sim p_t}[\varphi(X_t,y_t)]\}\,\mathrm{d}t + \mathrm{d}W_t.$$

Our goal is to understand the implicit regularization of the IM architecture where the expectation is approximated by a discrete measure $\nu = \frac{1}{w}\sum_{k=1}^{w}\delta_{\theta_k}$ and θ_k corresponds to the kth row of a $w\times d$ weight matrix θ. We show that the path preferred by gradient descent is the one that minimizes $\mathbb{E}_{y_t\sim\nu}\left[\varphi(X_t, y_t)\right]$, i.e. the solution with least influence from other particles. In addition, when φ can be decomposed as a norm, this amounts to finding the weights with smallest norm. For ease of notation, we will begin by presenting the proof for one time step and in 1-dimension, i.e. $d = 1$.

Following the blueprint given by Belabbas (2020), we wish to study the implicit bias of the weight matrix θ by understanding the compatibility between two optimization problems, the *training* problem given by the loss:

$$\min_\theta \mathcal{L}(\theta, X) = \sum_{i=1}^N \frac{1}{2\Delta_t^2}\left((X_{t+\Delta_t}^{(i)} - X_t^{(i)}) - \left(f(X_t^{(i)}, t) + \frac{1}{w}\sum_{k=1}^w \varphi(X_t^{(i)}, \theta_k)\right)\Delta_t\right)^2 \tag{13}$$

for observations $\left\{X_t^{(i)}, X_{t+1}^{(i)}\right\}_{i=1\dots N}$ and the *regularization* problem given by

$$\min_\theta K(\theta, X) \quad \text{s.t.} \quad \mathcal{L}(\theta, X) = 0$$

for some function K that satisfies the PDE:

$$\frac{\partial^2 K}{\partial\theta^2} g(\theta, X) + \sum_{i=1}^N \lambda_i \frac{\partial^2 \mathcal{L}}{\partial\theta^2}(\theta, X^{(i)}) g(\theta, X) = 0 \tag{14}$$

where g denotes the dynamics of gradient descent given as

$$g(\theta, X) = \dot{\theta} = \sum_{i=1}^N \frac{\partial\mathcal{L}}{\partial\theta}(\theta, X^{(i)}).$$

Following Belabbas (2020), the PDE(14) has a simple interpretation: the Hessian of K, acting on g, is a linear combination of the Hessians of $\mathcal{L}$ at datapoints $X^{(i)}$, acting on g. The next step is to find the function K.

We compute the first derivative

$$\partial_{\theta_j}\mathcal{L}^{(i)} = \left(\frac{-1}{\Delta_t w}\left((X_{t+\Delta_t}^{(i)} - X_t^{(i)}) - \left(f^{(i)} + \frac{1}{w}\sum_{k=1}^w \varphi(X_t^{(i)}, \theta_k)\right)\Delta_t\right)\partial_{\theta_j}\varphi(X_t^{(i)}, \theta_j)\right).$$

Then the second derivative as

$$\begin{aligned}\partial_{\theta_j,\theta_j}\mathcal{L}^{(i)} = &\Bigg(\frac{1}{w^2}(\partial_{\theta_j}\varphi(X_t^{(i)}, \theta_j))^2 \\ &\quad - \frac{1}{\Delta_t w}\left((X_{t+\Delta_t}^{(i)} - X_t^{(i)}) - \left(f^{(i)} + \frac{1}{w}\sum_{k=1}^w \varphi(X_t^{(i)}, \theta_k)\right)\Delta_t\right)\partial_{\theta_j,\theta_j}\varphi(X_t^{(i)}, \theta_j)\Bigg).\end{aligned}$$

with the off-diagonal second derivative as

$$\partial_{\theta_k,\theta_j}\mathcal{L}^{(i)} = \frac{1}{w^2}\partial_{\theta_i}\varphi(X_t^{(i)}, \theta_k)\partial_{\theta_j}\varphi(X_t^{(i)}, \theta_j).$$

The terms with coefficient $\frac{1}{w^2}$ will have coefficient $\frac{1}{w^3}$ when multiplied by the first partial derivative in g. Taking $w = \mathcal{O}(1/\Delta_t)$, these terms are negligible. With the choice of

$$\lambda_i = \Delta_t\left((X_{t+\Delta_t}^{(i)} - X_t^{(i)}) - \left(f^{(i)} + \frac{1}{w}\sum_{k=1}^w \varphi(X_t^{(i)}, \theta_k)\right)\Delta_t\right)^{-1}$$

we obtain the PDE

$$\frac{\partial^2 K}{\partial\theta^2} - \sum_{i=1}^N \frac{1}{w}\sum_{k=1}^w \partial_{\theta_k,\theta_k}\varphi(X_t^{(i)}, \theta_k) = 0.$$

This suggests that the regularization problem that we are solving, repeating for T time steps, is

$$\min_\theta K(\theta, X) = \sum_{t=1}^T\sum_{i=1}^N \frac{1}{w}\sum_{k=1}^w \varphi(X_t^{(i)}, \theta_k) \quad \text{s.t.} \quad \mathcal{L}(\theta, X) = 0. \tag{15}$$

In the context of the MV-SDE, the mean-field system approximated is the one that has the least influence from the other particles.

***d*-dimensions.** Now consider the case with θ_k as vectors. The notation becomes more complex as the partial derivatives now form tensors. However, since the diffusion is assumed to be constant and diagonal, we can give a brief analysis similar to the 1-dimensional case. The loss function is now

$$\mathcal{L}(\theta, X) = \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{1}{2\Delta_t^2}\sum_{j=1}^{d}\left((X_{t+\Delta_t}^{(i)} - X_t^{(i)}) - \left(f(X_t^{(i)}, t) + \frac{1}{w}\sum_{k=1}^{w}\varphi(X_t^{(i)}, \theta_k)\right)\Delta_t\right)_j^2.$$

The regularized problem has a similar form of

$$\min_\theta K(\theta, X) = \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{1}{w}\sum_{k=1}^{w}\sum_{j=1}^{d}\varphi(X_t^{(i)}, \theta_k)_j \quad \text{s.t.} \quad \mathcal{L}(\theta, X) = 0.$$

□

A.2.1 Comparison to optimal transport

Now consider the case $f = 0$ and recall that the transition density satisfies the PDE

$$\partial_t p(x,t) = -\nabla \cdot \left(\int_\Omega \varphi(x,y)p(y,t)\mathrm{d}y p(x,t)\right) + \frac{\sigma^2}{2}\nabla^2 p(x,t) \tag{16}$$

such that $p(x, 0) = p_0(x)$ and $p(x, T) = p_T(x)$. Suppose that φ can be represented as a norm $\|g(x, y)\|^2$ and replace the drift with the one given by the implicit bias, the PDE then becomes

$$\begin{aligned}\partial_t p(x,t) &= -\nabla \cdot \left(\min_\nu \left(\int_\Omega \|g(x,y)\|^2 \nu(y,t)\mathrm{d}y\right) p(x,t)\right) + \frac{\sigma^2}{2}\nabla^2 p(x,t)\\ &= -\nabla \cdot \left(\min_\nu \mathbb{E}_{y\sim\nu}\left[\|g(x,y)\|^2\right] p(x,t)\right) + \frac{\sigma^2}{2}\nabla^2 p(x,t)\\ &= -\nabla \cdot \left(\min_g g(x,t)p(x,t)\right) + \frac{\sigma^2}{2}\nabla^2 p(x,t)\end{aligned}$$

where the last step can be seen as a parameterization of the function g by the measure ν.

We see some similarities to the Benamou-Brenier form of the Wasserstein-2 distance, where the optimization problem is given by

$$W_2(\rho, \mu) = \min_g \int_0^T \mathbb{E}_{X_t\sim p(x,t)}\left[\|g(X_t, t)\|^2\right]\mathrm{d}t \tag{17}$$

subject to

$$\partial_t p = -\nabla \cdot (g(x,t)p(x,t)), \quad p_0(x) = \rho,\ p_T(x) = \mu. \tag{18}$$

Compare (15) to (17) where we have the same objective. In addition, note that the probability flow (16) satisfies the transport equation (18) in the limit as $\sigma \to 0$. This lends to an interpretation that, under certain choices of φ, the problem relates to the entropy regularized optimal transport problem under the W_2 cost. Notably, this comes as a result of the implicit bias introduced by the neural network gradient optimization scheme and is not a separate term that needs to be added.

A.3 Compatibility criterion in inferring explicit distributional dependence

A.3.1 Feynman-Kac for the Kolmogorov backward equation

The Kolmogorov backward and forward equations are PDEs that describe the time evolution of the marginal density of the associated SDE. The Kolmogorov backward equation describes the evolution of the density when given a known terminal condition. Its adjoint, the Kolmogorov forward equation, establishes an initial condition and provides the density at some future time. In this section, we focus on regularizing the modeled density to be consistent with the flow of the modeled SDE using the Kolmogorov backward equation. In Section C.4.5, we derive a likelihood and perform additional generative modeling experiments based on a linearization of the Kolmogorov forward equation, also known as the Fokker-Planck equation.

For the modeled density to be consistent with the flow of the modeled SDE, it has to satisfy the Kolmogorov backward equation defined as

$$-\partial_t p_t = b(\cdot)\nabla p_t + \frac{\sigma^2}{2}\nabla^2 p_t. \tag{19}$$

A solution to the above equation is given by the Feynman-Kac formula as an expectation of trajectories at terminal time, i.e.

$$p_t(x) = \mathbb{E}\left[p_T(X_T) \mid X_t = x\right] \tag{20}$$

where $p_T(\cdot)$, $t < T$ is the given terminal condition and X_s satisfies the SDE $\mathrm{d}X_s = b(\cdot)\mathrm{d}s + \sigma \mathrm{d}W_s$.

Following (20), we evolve X_s from $X_t = x$ to X_T, then penalize the difference between $p_t(x)$ and $\mathbb{E}\left[p_T(X_T) \mid X_t = x\right]$. The estimation algorithm with this compatibility criterion on the marginal density is detailed in Algorithm 3.

A.3.2 Feynman-Kac analysis of the Brownian bridge estimator

Consider the bridge estimator

$$\mathcal{L}_{BB} = P(\{Z_{t_{j+1}} = X_{t_{j+1}}\} \mid Z_{t_j} = X_{t_j}) = \mathbb{E}_{\mathbb{Q}}\left[\mathbb{1}\{Z_{t_{j+1}} = X_{t_{j+1}}\} \mid Z_{t_j} = X_{t_j}\right]$$

where the expectation is taken over Brownian paths Z_t under the Wiener measure $\mathbb{Q}$. This computes the probability that a Brownian motion Z_t, conditioned to be equal to X_{t_j} at t_j, is equal to $X_{t_{j+1}}$ at t_{j+1}. This can be thought of using the Kolmogorov backward equation and Feynman-Kac formula from the previous section. Applying a change of measure using Girsanov's theorem to a drifted Brownian motion, we arrive at the estimator described in the main text

$$\mathcal{L}_{BB}(\theta) = \mathbb{E}_{\mathbb{Q}}\left[\mathbb{1}\{Z_{t_{j+1}} = X_{t_{j+1}}\} \exp\left(\int_{t_j}^{t_{j+1}} b(\,\cdot\,;\theta)\mathrm{d}Z_t - \frac{1}{2}\int_{t_j}^{t_{j+1}} b(\,\cdot\,;\theta)^2 \mathrm{d}t\right) \mid Z_{t_j} = X_{t_j}\right].$$

The indicator function, which acts as the boundary condition for the Kolmogorov backward equation, restricts the paths of $\mathbb{Q}$ to those that are Brownian bridges between X_{t_j} and $X_{t_{j+1}}$. The change of measure via Girsanov's provides the mechanism for inferring the optimal drift for the observed data.

The experiments then provide a way of evaluating whether including distributional properties in the drift (i.e. *nonlinear* Kolmogorov backward equation with $b(X_t, p_t, t; \theta)$) results in better probabilities than without (i.e. linear Kolmogorov backward equation with $b(X_t, t; \theta)$).

A.4 Gradient flow minimizing the energy distance

The energy distance is defined as

$$E(p, q) = 2\int\int \|X - Y\|\mathrm{d}q(X)\mathrm{d}p(Y) - \int\int \|X - X'\|\mathrm{d}q(X)\mathrm{d}q(X') - \|Y - Y'\|\mathrm{d}p(Y)\mathrm{d}p(Y')$$

and by linearity of expectations, we can rewrite as

$$E(p, q) = 2\int\int \left(\underbrace{\|X - X'\|\frac{\mathrm{d}p(Y)}{\mathrm{d}q(X')} - \|X - X'\| - \|X - X'\|\left(\frac{\mathrm{d}p(Y)}{\mathrm{d}q(X')}\right)^2}_{\varphi}\right)\mathrm{d}q(X)\mathrm{d}q(X').$$

Our goal is to find a gradient flow that minimizes this distance. Following Santambrogio (2017), we can use the gradient of this function as the drift to promote attraction to the target density. We will use the form within the parenthesis as φ for representing the aggregation potential, $\mathcal{W}(p) = \int\int \varphi(X, X')\mathrm{d}p(X)\mathrm{d}p(X')$. Additionally, we include the Radon-Nikodym derivative for this between the target density and the current time density to define the energy distance. Applying the gradient to φ, we achieve the desired result. While we do not explicitly impose this, we use this idea to motivate our results on generative modeling.

B Algorithms

To supplement the algorithmic contributions in the main paper we detail the inference procedure for the regular time observations in Algorithm 1 and the irregular time observations with Brownian bridges in Algorithm 2. We then detail the inference procedure with regularization of the marginal law using a compatibility criterion of the PDE with the associated SDE in Algorithm 3. Finally, we describe a sampling procedure in Algorithm 4. The code is attached in the supplementary material and will be posted online.

Algorithm 1 Maximum Likelihood Estimation (MLE) with Girsanov's Theorem

```
Input: observed trajectories $\left\{\{X_{t_j}\}^{(i)}_{j=1...K}\right\}_{i=1...N}$.
Initialize: neural drift $b(\cdot;\theta)$.
for $i$ in mini-batch do
  for $j$ in $1...K-1$ do
    Compute $\Delta X^{(i)}_{t_j} = X^{(i)}_{t_{j+1}} - X^{(i)}_{t_j}$.
    Compute discretized approximation to log of exponential martingale:
    $\mathcal{L}(\theta) := b(X^{(i)}_{t_j}, p_{t_j}, t_j;\theta)\Delta X^{(i)}_{t_j} - \frac{1}{2} b(X^{(i)}_{t_j}, p_{t_j}, t_j;\theta)^2(t_{j+1}-t_j)$.
    Maximize $\mathcal{L}(\theta)$ using gradient based optimizer.
  end for
end for
```

In the computation of the mean-field component of $b(X^{(i)}_{t_j}, p_{t_j}, t_j;\theta)$, we do the following:

- EM architecture: $\frac{1}{n}\sum_{k=1}^n \varphi(X^{(i)}_{t_j}, X^{(k)}_{t_j};\theta)$.
- IM architecture: $\frac{1}{n}\sum_{k=1}^n \varphi(X^{(i)}_{t_j}, W^{(k)}_0, t_j;\theta)$
  as a neural network with an additional layer and additional conditioning on t_j.
- ML architecture: $\frac{1}{n}\sum_{k=1}^n \varphi(X^{(i)}_{t_j}, \hat{X}^{(k)}_{t_j};\theta)$
  where we compute the expectation with samples $\{\hat{X}^{(k)}_{t_j}\}_{k=1}^n$ from $\hat{p}(\cdot, t_j;\phi)$,
  a generative network with additional conditioning on t_j.

Additional details on the parameterization of the neural architectures are in Section C.3.

In the case of irregular time observations, for each trajectory, we first sample Brownian bridges between observations, then use the sampled Brownian bridges as regular time observations. In this case, the estimation procedure aims to fit the observations while penalizing deviations from the Brownian bridge paths in regions without observations. The Brownian bridge approach also has the interpretation of the shortest distance interpolator that exactly fits the margins. Using a Brownian bridge path construction reduces the variance of the estimator.

We next detail the estimation procedure with regularization of the marginal law using the correspondence between the PDE and its associated SDE via the nonlinear Kolmogorov backwards equation (Buckdahn et al., 2017).

Algorithm 2 MLE with Girsanov and Brownian Bridges

```
Input: observed trajectories $\left\{\{X_{t_k}\}_{k=1...K}^{(i)}\right\}_{i=1...N}$.
Initialize: neural drift $b(\cdot;\theta)$.
for $i$ in mini-batch do
  for $k$ in $1...K-1$ do
    Sample Brownian bridge $\{Z_{t_j}\}_{j=1...J}^{(i)}$ between $X_{t_k}^{(i)}$ and $X_{t_{k+1}}^{(i)}$.
    for $j$ in $1...J-1$ do
      Compute $\Delta Z_{t_j}^{(i)} = Z_{t_{j+1}}^{(i)} - Z_{t_j}^{(i)}$.
      Compute discretized approximation to log of exponential martingale:
      $\mathcal{L}(\theta) := b(Z_{t_j}^{(i)}, p_{t_j}, t_j; \theta)\Delta Z_{t_j}^{(i)} - \frac{1}{2} b(Z_{t_j}^{(i)}, p_{t_j}, t_j; \theta)^2 (t_{j+1} - t_j)$.
      Maximize $\mathcal{L}(\theta)$ using gradient based optimizer.
    end for
  end for
end for
```

Algorithm 3 MLE with Girsanov and Regularization of Explicit Marginal Law $\hat{p}_t$

```
Input: observed trajectories $\left\{\{X_{t_j}\}_{j=1...J}^{(i)}\right\}_{i=1...N}$.
Initialize: neural drift $b(\cdot;\theta,\phi)$, including explicit marginal law $\hat{p}(\cdot;\phi)$.
for $i$ in mini-batch do
  for $j$ in $1...J-1$ do
    Compute $\Delta X_{t_j}^{(i)} = X_{t_{j+1}}^{(i)} - X_{t_j}^{(i)}$.
    Compute discretized approximation to log of exponential martingale:
    ELBO $:= b(X_{t_j}^{(i)}, p_{t_j}, t_j; \theta, \phi)\Delta X_{t_j}^{(i)} - \frac{1}{2} b(X_{t_j}^{(i)}, p_{t_j}, t_j; \theta, \phi)^2 (t_{j+1} - t_j)$.
    Sample $\{\{Z_{t_{j+1}} | z = X_{t_j}^{(i)}\}^{(k)}\}_{k=1...K}$ following the dynamics of the ML architecture.
    Compute the expected log-likelihood $\mathbb{E}[\log \hat{p}_{t_{j+1}}(Z_{t_{j+1}};\phi)] = \frac{1}{K} \log \hat{p}_{t_{j+1}}(Z_{t_{j+1}}^{(k)};\phi)$.
    Compute compatibility criterion CC $:= (\log \hat{p}_{t_j}(X_{t_j}^{(i)};\phi) - \mathbb{E}[\log \hat{p}_{t_{j+1}}(Z_{t_{j+1}};\phi)])^2$
    Compute total loss $\mathcal{L}(\theta) :=$ ELBO + CC.
    Maximize $\mathcal{L}(\theta)$ using gradient based optimizer.
  end for
end for
```

Algorithm 4 Sampling Trajectories with Euler-Maruyama Scheme

```
Initialize: time grid $\{t_j\}_{j=1...K}$.
Initialize: initial observations $\{X_0^{(i)}\}_{i=1...N} \sim p_0$.
for $j$ in $1...K-1$ do
  Compute $\Delta t_j = t_{j+1} - t_j$.
  for $i$ in $1...N$ do
    Sample $\Delta W_{t_j}^{(i)} \sim_{iid} \mathcal{N}(0, \Delta t_j)$
    Compute $X_{t_{j+1}}^{(i)} = b(X_{t_j}^{(i)}, p_{t_j}, t_j; \theta)\Delta t_j + \sigma \mathrm{d}W_{t_j}^{(i)}$.
  end for
end for
```

C Experimental details

In this section, we detail the evaluation metrics, datasets, hyperparameter settings, and provide additional experiments to supplement the results in the main paper.

C.1 Evaluation metrics

C.1.1 Continuous ranked probability score

Following Gneiting and Raftery (2007), the continuous ranked probability score (CRPS) is given by

$$\mathrm{CRPS}(F, x) = \int_{-\infty}^{\infty} \left[F(y) - \mathbb{1}(y \geq x)\right]^2 \mathrm{d}y.$$

The CRPS evaluates the modeled distribution against a single observation by comparing the cumulative distribution function (CDF) of the modeled distribution F to a step function placed at the observation x.

C.1.2 Energy distance

The squared energy distance between two distributions P_0 and P is defined as

$$d^2(P_0, P) := 2\,\mathbb{E}_{X\sim P_0, Y\sim P}[\|X - Y\|] - \mathbb{E}_{X\sim P_0, X'\sim P_0}[\|X - X'\|] - \mathbb{E}_{Y\sim P, Y'\sim P}[\|Y - Y'\|]$$

where we compute the expectations empirically.

C.1.3 KS-distance

The KS-distance between two 1-d empirical cumulative distribution functions (ECDF) is defined as:

$$\text{KS-distance} = \sup_x |F_1(x) - F_2(x)|$$

where F_1, F_2 are two ECDFs. This metric is only used in the appendix for additional 1-dimensional experiment results.

C.2 Datasets

Here we describe the datasets in more detail and provide exact statements on the simulation parameters.

C.2.1 Synthetic time series data

Kuramoto model. The Kuramoto model which describes synchronizing oscillators takes the form

$$\mathrm{d}X_t^{(i)} = \left[h^{(i)} + \frac{K}{N}\sum_{j=1}^{N} \sin\left(y_t^{(j)} - X_t^{(i)}\right)\right]\mathrm{d}t + \sigma \mathrm{d}W_t^{(i)},$$

where movements of N particles are governed by a linearly factored drift that includes some function $h^{(i)}$ and a mean-field term that couples the particles. We simulate 2-d trajectories with $X_t^{(i)} = [X_{1t}^{(i)}, X_{2t}^{(i)}] \in \mathbb{R}^2$, $h^{(i)} = \left[\sin(X_{1t}^{(i)}), \sin(X_{2t}^{(i)})\right]$, $K = 2$, $N = 20$, and $\sigma = 1$.

Fitzhugh-Nagumo model. The FitzHugh-Nagumo model is a set of equations that models spikes in neuron activations as membrane voltage spikes X_{1t}, driven by external stimulus I_{ext}, and diminishing over time X_{2t}. It takes the form

$$\begin{aligned}
\mathrm{d}X_{1t} &= \left(aX_{1t}\left(X_{1t} - \lambda\right)\left(1 - X_{1t}\right) - X_{2t} + I_{\text{ext}}\right)\mathrm{d}t + \mathbb{E}\left[X_{1t} - y_{1t}\right]\mathrm{d}t + \sigma \mathrm{d}W_t, \\
\mathrm{d}X_{2t} &= \left(-bX_{2t} + cX_{1t} + d\right)\mathrm{d}t,
\end{aligned}$$

We chose $a = 0.2, b = 0.8, c = 1, d = 0.7, \lambda = 0.4, I_{\text{ext}} = 0.1\sin(10t)$, and $\sigma = 0.3$. The expectation is approximated with $N = 20$ particles.

Opinion dynamic model. The opinion dynamic model simulates the opinion formation process through an equation with the form

$$dX_t = \mathbb{E}\left[\psi_\theta(||X_t - y_t||)(X_t - y_t)\right] + \sigma \mathrm{d}W_t,$$

where $\psi_\theta(r) = \theta_1 \exp\left(-\frac{0.01}{1-(r-\theta_2)^2}\right)$. We simulate 2-$d$ trajectories with $\theta_1 = 1, \theta_2 = 2.5$.

Mean-field atlas model. The mean-field atlas model for pricing equity markets takes the form

$$\mathrm{d}X_t = \gamma\left(\int \mathbb{1}_{\{X_t - y_t > 0\}} \mathrm{d}p_t(y)\right)\mathrm{d}t + \sigma \mathrm{d}W_t,$$

where the drift $\gamma(\cdot)$ depends on the rank of the particle at each time. Let $u = \int \mathbb{1}_{\{X_t - y_t > 0\}} \mathrm{d}p_t(y)\mathrm{d}t$, we define $\gamma = 1 - u\exp(2u)$.

Itô diffusion - Ornstein-Uhlenbeck. We simulated a 2-d Ornstein-Uhlenbeck (OU) process with drifts $\left[-3X_{1t}, -2X_{2t}\right]$.

Itô diffusion - circle. We simulated a 2-d SDE with circular evolution given by drifts $[-X_{1t} - 2X_{2t}, -X_{2t} + 2X_{1t}]$.

Jump diffusions. We simulated a 2-d OU process with drifts $[-X_{1t}, -X_{2t}]$ and additional 1, 2, or 4 jumps sampled uniformly in time with jump size distributed as exp(Uniform$(2, 3)$).

All models are two-dimensional except the mean-field atlas model that is one-dimensional.

We first simulated samples using the Euler-Maruyama method on a fine grid Δt, i.e. $X_{t+\Delta t} = X_t + b(X_t, p_t, t)\Delta t + \sigma \Delta W$ with $\Delta W \sim \mathcal{N}(0, \Delta_t)$ and $t \in [0, T]$. For irregular time samples, a batch of observation times are then sampled according to an exponential distribution with rate $\lambda = T/N'$, where N' is the number of irregular time samples. The sampled timestamps are then matched to the closest times in the discretized time sequence used in sample generation. Only the matched timestamps t', the initial condition X_0, and the terminal condition X_T are used in training. For evaluation, we consider the full trajectories. Specific choices of $\sigma, T, \Delta t, N$, and N' are provided in Table 3. To realistically simulate real-world parameter estimation, "observation noise" in the form of Gaussian with standard deviations $\in \{0.1, 0.5, 1\}$ is added to the sampled data.

Dataset	σ	Terminal Time T	Δt	# Particles N	# Irregular Observation N'
Kuramoto	1	5	0.05		
Fitzhugh-Nagumo	0.3	5	0.05		
Opinion Dynamic	0.5	100	1.0		
Mean-field Atlas	1	5	0.05	20	20
Ornstein-Uhlenbeck	1	5	0.05		
Circles	1	5	0.05		
OU with Jumps	1	5	0.05	100	Not Applicable

Table 3: Synthetic time series parameters

C.2.2 Real time series data

EEG data. We used the 1-dimensional EEG data provided by Zhang et al. (1995). Specifically, the EEGs recorded with stimulus 1. Each subject has 64 time series, and each time series has 256 timesteps. We used the following subject-run combinations for Non-Alcoholics EEG (NA-EEG): co2c0000362-076, co2c0000367-052, co2c0000338-016, co2c0000394-044, co2c0000348-016; and these subject-run combinations for Alcoholics EEG (A-EEG): co2a0000364-000, co2a0000372-014, co2a0000396-112, co2a0000411-064, co2a0000390-030. We did not perform any further preprocessing on this dataset.

Crowd trajectory data. We used the 2-dimensional Crowd trajectory data provided by Kothari et al. (2021). We used the crowd_students001.ndjson dataset and extracted every trajectory that starts at the 10th frame with a trajectory length longer than 50 frames. We truncate the data by taking only up to the 50th frame. This leaves 23 pedestrian trajectories to train with. Trajectories that did not start at the 10th frame, or shorter than 50 frames are discarded.

Chemotaxi data. We used the 3-d Chemotaxi data provided by Grognot and Taute (2021). We used *V_0208* for C. Crescentus and *V_MeAsp1_0511* for E. Coli. The time series are truncated to the first 100 timesteps. Particles with less than 100 timesteps recorded are discarded.

All time series data are split into 0.8 training, 0.1 validation, and 0.1 testing particles.

C.2.3 Generative data

We are interested in estimating a flow between a Gaussian and a target distribution described by the nonlinear Fokker-Planck equation. We thus sample a batch of $N = 100$ particles from the initial condition $\mathcal{N}(0, I_{d\times d})$ where d is the dimensions of the process. We then sample the same number of particles from different terminal conditions corresponding to the different datasets, i.e. Gaussian mixture and UCI datasets: Power, Miniboone, Hepmass, Gas and Cortex. To create the training dataset, we randomly match the particles from the initial condition to the particles from the terminal condition, then sample $N_{BB} = 30$ Brownian bridges between each initial-terminal condition pair for $t \in [0, T], T = 0.1, \Delta t = 0.002$.

Eight Gaussians. In the case of two dimensions, the terminal condition is an eight Gaussian mixture with means $\mu \in \{[0,2],[0,-2],[2,0],[0,-2],[\sqrt{2},\sqrt{2}],[\sqrt{2},-\sqrt{2}],[-\sqrt{2},\sqrt{2}],[-\sqrt{2},-\sqrt{2}]\}$ and variance $I_{d\times d}$. For dimensions 10, 30, 50, 100, μ is repeated 5, 15, 25, and 50 times.

Real data. For Power, Miniboone, Hepmass, and Gas, we follow the preprocessing of Grathwohl et al. (2019). For the Cortex data, we normalize the data by subtracting the mean and dividing by the standard deviation.

C.3 Hyperparameter settings

Since our goal is to determine the effect of different architectures, we try to control such that all architectures have similar number of parameters. The details of the hyperparameters are specified in Tables 4, 5, 7, 6 and 8 for the different datasets. The learned measure W_0 in the IM architecture was modeled as an additional fully connected layer. The marginal law $\hat{p}_t$ in the ML architecture was modeled with GLOW (Kingma and Dhariwal, 2018) with an additional conditioning on time.

For the MV-SDE models, we used the AdamW optimizer with a learning rate of 1×10^{-4}, $\epsilon = 1 \times 10^{-4}$ and exponential decay $\gamma = 0.9998$ for all experiments, except EEG where the learning rate was 1×10^{-3}. For the DeepAR models, the learning rate was 1×10^{-3}. The batch sizes used were $10, 5, 5, 10$ and 200 for the synthetic time series, EEG, Crowd Trajectory, Chemotaxis and generative modeling experiments respectively. The models were trained for $500, 500, 200, 2000$, and 500 epochs for the synthetic time series, EEG, Crowd Trajectory, Chemotaxis and generative modeling experiments.

Architecture	Hidden Layers	Layer Size	Activation	# of Parameters
MLP (Itô)	8	128		132740
EM	φ: 4, f: 4	128, 128	LeakyReLU	133638
IM	φ: 4, f: 4, W_0: 1	128, 128, 128		134022
ML	φ: 4, f: 4, $\hat{p}_t$: 1	128, 128, 32	s: tanh, t: ReLU	136152
MLP (Itô)	4	128		66820
EM	φ: 2, f: 2	128, 128	LeakyReLU	67718
IM	φ: 2, f: 2, W_0: 1	128, 128, 128		67846
ML	φ: 2, f: 2, $\hat{p}_t$: 1	128, 128, 32	s: tanh, t: ReLU	70232

Table 4: Hyperparameter specification for synthetic time series data experiments and synthetic generative modeling experiments. The first set of hyperparameter settings are for: Kuramoto, Opinion Dynamic, Mean-field Atlas, Jump Diffusions, and Eight Gaussians. The second set of hyperparameter settings are for: Fitzhugh-Nagumo, Itô-OU, Itô-Circles. For the jump diffusion experiments, the LeakyReLU activation on the EM architecture led to diverging behavior, while tanh did not. We then changed the activation to tanh for more stable behaviour.

Architecture	Modules: Hidden Layers	Layer Size	Activation	# of Parameters
MLP (Itô)	10	64	LeakyReLU	141858
EM	φ: 4, f: 4	64, 64		133795
IM	φ: 4, f: 4, W_0: 1	64, 64, 512		134371
ML	φ: 4, f: 4, $\hat{p}_t$: 3	64, 64, 32	s: tanh, t: ReLU	135960
DeepAR-LSTM	3	64	LeakyReLU	117894
DeepAR-RNN		130		120516
DeepAR-GRU		80		137526
DeepAR-TR	Enc: 8, Dec: 8	512	ReLU	298082

Table 5: Hyperparameter specification for real time series data - EEG experiments. For the DeepAR models, we used a window size of 20.

Architecture	Modules: Hidden Layers	Layer Size	Activation	# of Parameters
MLP (Itô)	8	128	LeakyReLU	133126
EM	φ: 4, f: 4	128, 128		134409
IM	φ: 4, f: 4, W_0: 1	128, 128, 128		134921
ML	φ: 4, f: 4, $\hat{p}_t$: 3	128, 128, 128	s: tanh, t: ReLU	239724
DeepAR-LSTM	3	64	LeakyReLU	117123
DeepAR-RNN		130		119733
DeepAR-GRU		80		136723
DeepAR-TR	Enc: 4, Dec: 4	256	ReLU	81638

Table 6: Hyperparameter specification for real time series data - Chemotaxi experiments. For the DeepAR models, we used a window size of 20.

Architecture	Modules: Hidden Layers	Layer Size	Activation	# of Parameters
MLP (Itô)	2	64	LeakyReLU	8644
EM	φ: 1, f: 1	64, 64		9094
IM	φ: 1, f: 1, W_0: 1	64,64,64		9286
ML	φ: 1, f: 1, $\hat{p}_t$: 1	64, 64, 128	s: tanh, t: ReLU	43672
DeepAR-LSTM	3	64	LeakyReLU	117510
DeepAR-RNN		130		120126
DeepAR-GRU		80		137126
DeepAR-TR	Enc: 8, Dec: 8	256	ReLU	298148

Table 7: Hyperparameter specification for real time series data - Crowd Trajectory experiments. For the DeepAR models using a window size of 10.

Architecture	Modules: Hidden Layers	Layer Size	Activation	# of Parameters
MLP (Itô)	8	128	tanh	133900 $\sim$ 151960
EM	φ: 4, f: 4	64, 128		86098 $\sim$ 122148
IM	φ: 4, f: 4, W_0: 1	64, 128, 128		86930 $\sim$ 131940
ML	φ: 4, f: 4, $\hat{p}_t$: 1	64, 128, 32	s: tanh, t: ReLU	89208 $\sim$ 146048
MAF	4	128	ReLU	75872 $\sim$ 184512
WGAN	Gen: 4, Dis: 3	Gen: [64, 128, 256], Dis: 256	LeakyReLU	114759 $\sim$ 150669
VAE	Enc: 4, Dec: 4	128, 256, latent dim: 50	LeakyReLU	88682 $\sim$ 124592
Score-Based	8	128	SiLU	117318 $\sim$ 135308

Table 8: Hyperparameter specification for generative modeling experiments: Power, Miniboone, Hepmass, Gas and Cortex. The number of parameters depends on the dimension of the data. The hyperparameter specification for the generative modeling experiments with Eight Gaussians follow that of Table 4.

C.4 Additional figures and tables

We provide a series of additional figures to qualitatively illustrate the differences between the proposed architectures and baselines.

C.4.1 Ablation on IM architecture number of training particles

In Table 9 we show that as the number of training particles grows, the KS distance will approach 0. Note that since KS-statistic only applies to 1-d ECDFs. We consider a 1-d Kuramoto simulation with the same parameters and no additional observation noise.

N	mean	75%	90%	KS
10	0.0481 (0.023)	0.0794 (0.041)	0.1021 (0.042)	0.1222 (0.041)
50	0.0272 (0.018)	0.0434 (0.033)	0.0609 (0.038)	0.0792 (0.038)
100	0.0177 (0.006)	0.0278 (0.009)	0.0383 (0.009)	0.0544 (0.008)
500	0.0162 (0.005)	0.0249 (0.01)	0.0352 (0.01)	0.0519 (0.009)

Table 9: ECDF distance of 1-d Kuramoto under $N = 10, 50, 100$ and 500 training particles and 500 test particles. The values represent the distance between the ECDFs averaged over all time marginals at the mean, 75th, 90th, and 100 percentiles.

C.4.2 Synthetic data experiments

We provide tables with variance for the synthetic dataset result showcased in the main text. We also provide a KS-distance table for the 1-dimensiaonl mean-field atlas experiment. KS-distance is only applicable in 1-dimensional cases.

Table 10: Synthetic dataset results with noise level standard deviation 0.1

	Kuramoto	Fitzhugh	OD	MA	OU	Circle
MLP (Itô)	0.560 (0.081)	0.699 (0.426)	0.048 (0.013)	2.140 (0.142)	0.098 (0.043)	1.351 (1.979)
IM	0.448 (0.075)	0.601 (0.422)	0.039 (0.011)	1.208 (0.147)	0.128 (0.047)	1.592 (2.38)
ML	0.428 (0.095)	0.639 (0.395)	0.042 (0.012)	1.519 (0.236)	0.101 (0.038)	1.481 (2.237)
EM	0.383 (0.085)	0.606 (0.389)	0.036 (0.010)	1.359 (0.200)	0.097 (0.038)	1.562 (2.476)

Table 11: Synthetic dataset results with noise level standard deviation 0.5

	Kuramoto	Fitzhugh	OD	MA	OU	Circle
MLP (Itô)	0.578 (0.124)	0.734 (0.489)	0.047 (0.012)	2.133 (0.156)	0.100 (0.042)	1.334 (1.948)
IM	0.450 (0.074)	0.617 (0.415)	0.039 (0.012)	1.223 (0.125)	0.128 (0.046)	1.592 (2.368)
ML	0.397 (0.075)	0.605 (0.408)	0.042 (0.011)	1.518 (0.248)	0.100 (0.035)	1.564 (2.543)
EM	0.373 (0.075)	0.612 (0.383)	0.038 (0.010)	1.347 (0.165)	0.106 (0.036)	1.535 (2.535)

Table 12: Synthetic dataset results with noise level standard deviation 1.0

	Kuramoto	Fitzhugh	OD	MA	OU	Circle
MLP (Itô)	0.653 (0.067)	0.897 (0.503)	0.059 (0.009)	2.159 (0.200)	0.481 (0.065)	2.303 (2.039)
IM	0.646 (0.065)	0.878 (0.522)	0.055 (0.012)	1.650 (0.232)	0.559 (0.062)	2.658 (2.142)
ML	0.601 (0.112)	0.882 (0.527)	0.049 (0.009)	1.748 (0.224)	0.529 (0.055)	2.308 (2.252)
EM	0.592 (0.077)	0.893 (0.523)	0.040 (0.009)	1.652 (0.272)	0.536 (0.055)	2.394 (2.250)

Table 13: ECDF distance of mean-field atlas process under 20 training particles and 20 irregular time samples with different noises. The result is tested with 100 test particles and averaged across 10 runs.

	mean ECDF	75% ECDF	90% ECDF	KS
Noise = 1.0				
MLP	0.1 (0.01)	0.16 (0.03)	0.27 (0.03)	0.35 (0.03)
IM	0.05 (0.01)	0.07 (0.02)	0.13 (0.03)	0.21 (0.03)
ML	0.05 (0.02)	0.07 (0.03)	0.14 (0.05)	0.24 (0.03)
EM	0.05 (0.01)	0.07 (0.03)	0.15 (0.04)	0.24 (0.03)
Noise = 0.5				
MLP	0.09 (0.01)	0.16 (0.02)	0.26 (0.02)	0.34 (0.03)
IM	0.03 (0.01)	0.04 (0.01)	0.08 (0.02)	0.15 (0.02)
ML	0.04 (0.01)	0.06 (0.02)	0.12 (0.03)	0.21 (0.03)
EM	0.03 (0.01)	0.04 (0.01)	0.07 (0.02)	0.14 (0.02)
Noise = 0.1				
MLP	0.1 (0.01)	0.16 (0.03)	0.26 (0.03)	0.34 (0.02)
IM	0.03 (0.01)	0.04 (0.01)	0.08 (0.02)	0.14 (0.02)
ML	0.04 (0.01)	0.06 (0.02)	0.13 (0.03)	0.21 (0.02)
EM	0.03 (0.01)	0.04 (0.01)	0.08 (0.02)	0.14 (0.02)

C.4.3 Real data experiments

We provide an additional experiment on brain EEG data recorded for both alcoholic and non-alcoholic subject. We also extend the time series experiments in the main paper to different types of generation and forecasting. For generation, we begin with an initial condition at T_0 and generate a trajectory up to T_{forecast}. There are two ways to perform forecasting: i) given the initial condition at T_0, generate trajectories up to T_{forecast}; ii) given the training terminal condition at T, generate trajectories for $t \in [T, T_{\text{forecast}}]$. In both cases, the dataset time steps are partitioned into 0.8 training, 0.2 forecasting. We present the numerical results of generation and both types of forecasting in Tables 14, 15 and 16. We note that our methods perform on par with various DeepAR methods under both types of forecasting.

Table 14: Time series generation on held out trajectories. NA/A stands for non-alcoholics/alcoholics. Values in **bold** and *italic* are best and second best.

	CRPS ↓		MSE ↓
	NA-EEG	A-EEG	Crowd Traj
MLP (Itô)	5.795 (1.53)	4.404 1.593	0.939 0.477
IM	5.179 (1.24)	*4.163* 1.219	*0.811* 0.888
ML	**5.103** (1.092)	**4.052** 1.006	**0.51** 0.569
EM	*5.174* (1.397)	4.252 1.348	0.877 0.442
LSTM	6.129 (2.237)	5.679 2.557	1.803 0.373
RNN	6.052 (2.286)	4.643 1.379	1.367 1.066
GRU	6.201 (2.219)	6.176 2.726	1.133 0.318
TR	5.986 (1.614)	4.295 1.357	3.278 1.078

Table 15: Time series forecasting Type I. NA/A stands for Non-alcoholics/ Alcoholics. **Bolded** values indicate best performance.

	CRPS ↓		MSE ↓		
	NA-EEG	A-EEG	Crowd Traj	C.Cres	E.Coli
MLP (Itô)	30.087 (32.29)	7.837 (3.018)	3.502 2.017	**0.296** (0.007)	**0.225** (0.007)
IM	8.346 (4.646)	5.438 (1.814)	3.458 3.609	0.307 (0.010)	0.23 (0.006)
ML	7.967 (4.542)	5.652 (1.515)	4.006 6.418	0.312 (0.015)	0.245 (0.006)
EM	8.963 (4.309)	5.82 (1.818)	3.214 1.215	0.312 (0.019)	0.26 (0.013)
LSTM	7.231 (3.051)	6.66 (3.948)	2.876 0.804	1.526 (0.324)	0.786 (0.386)
RNN	**6.993** (2.369)	5.292 (2.317)	2.691 2.604	1.689 (1.107)	0.859 (0.115)
GRU	7.234 (2.75)	7.407 (4.494)	2.292 0.515	1.115 (0.406)	0.813 (0.337)
TR	7.354 (1.998)	**5.122** (2.457)	5.516 2.207	1.489 (0.362)	1.489 (0.362)

Table 16: Time series forecasting Type II. NA/A stands for Non-alcoholics/ Alcoholics. **Bolded** values indicate best performance.

	CRPS ↓		MSE ↓	
	NA-EEG	A-EEG	C.Cres	E.Coli
MLP (Itô)	31.47 (35.659)	6.95 (2.640)	**0.013** (0.0003)	**0.015 (0.0003)**
IM	8.675 (5.638)	**4.884** (1.687)	0.014 (0.0007)	0.016 (0.0003)
ML	8.747 (5.677)	4.907 (1.490)	0.015 (0.0007)	0.015 (0.0005)
EM	8.938 (4.975)	5.403 (2.205)	0.015 (0.0013)	0.015 (0.0005)
LSTM	8.288 (3.142)	6.317 (4.207)	0.291 (0.0704)	0.163 (0.0353)
RNN	**7.002** (2.591)	5.296 (2.262)	1.455 (0.9367)	0.534 (0.191)
GRU	7.019 (2.686)	6.044 (3.457)	0.397 (0.2134)	0.17 (0.054)
TR	7.087 (2.208)	4.971 (2.643)	1.65 (0.1666)	1.65 (0.1666)

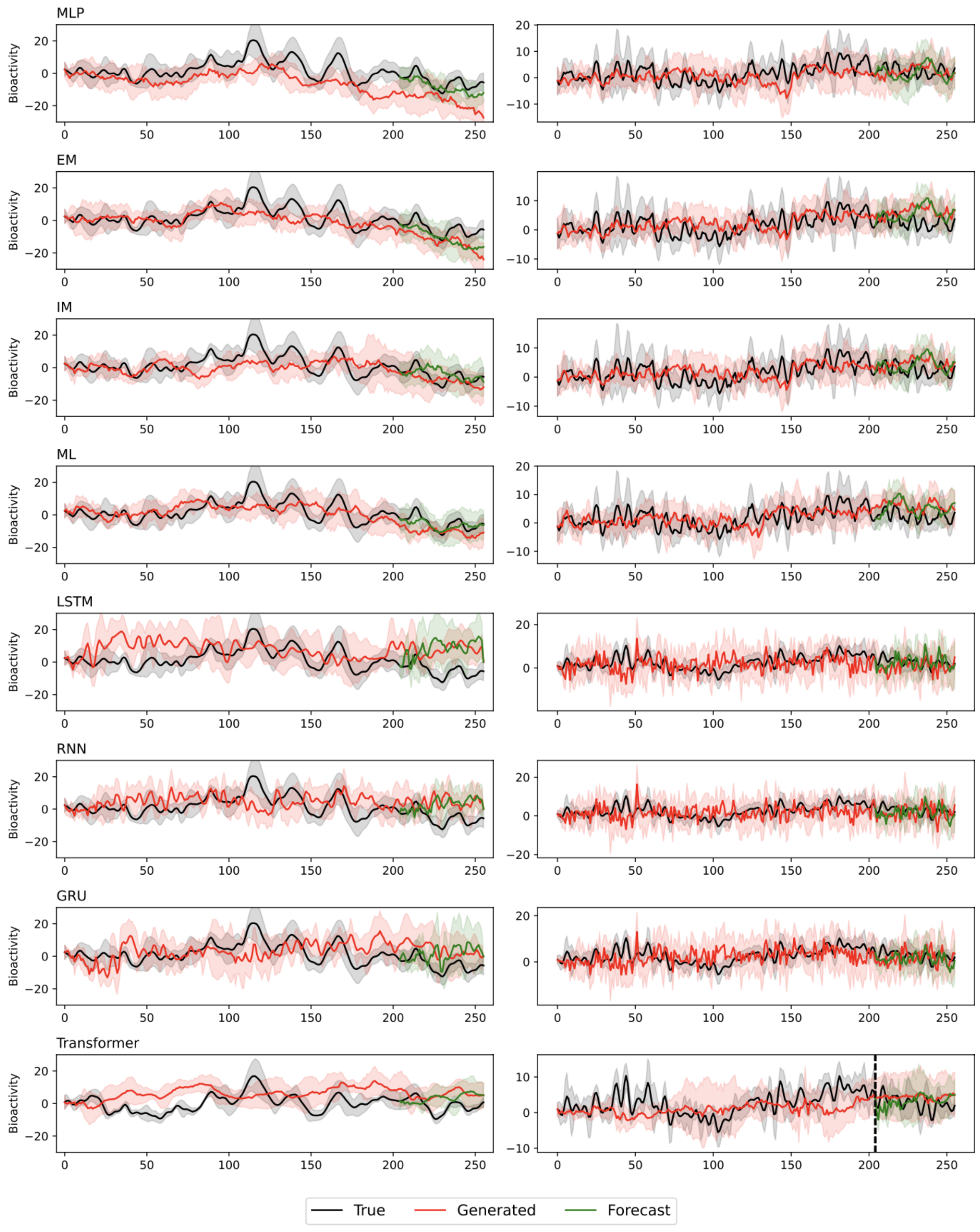


Figure 7: True, generated and forecasted trajectories on EEG dataset. Left: Non-alcoholics; Right: Alcoholics. The dashed vertical line at $t = 205$ indicates the start of the forecast.The shaded region indicates $\pm$ one standard deviation of samples at each time step.

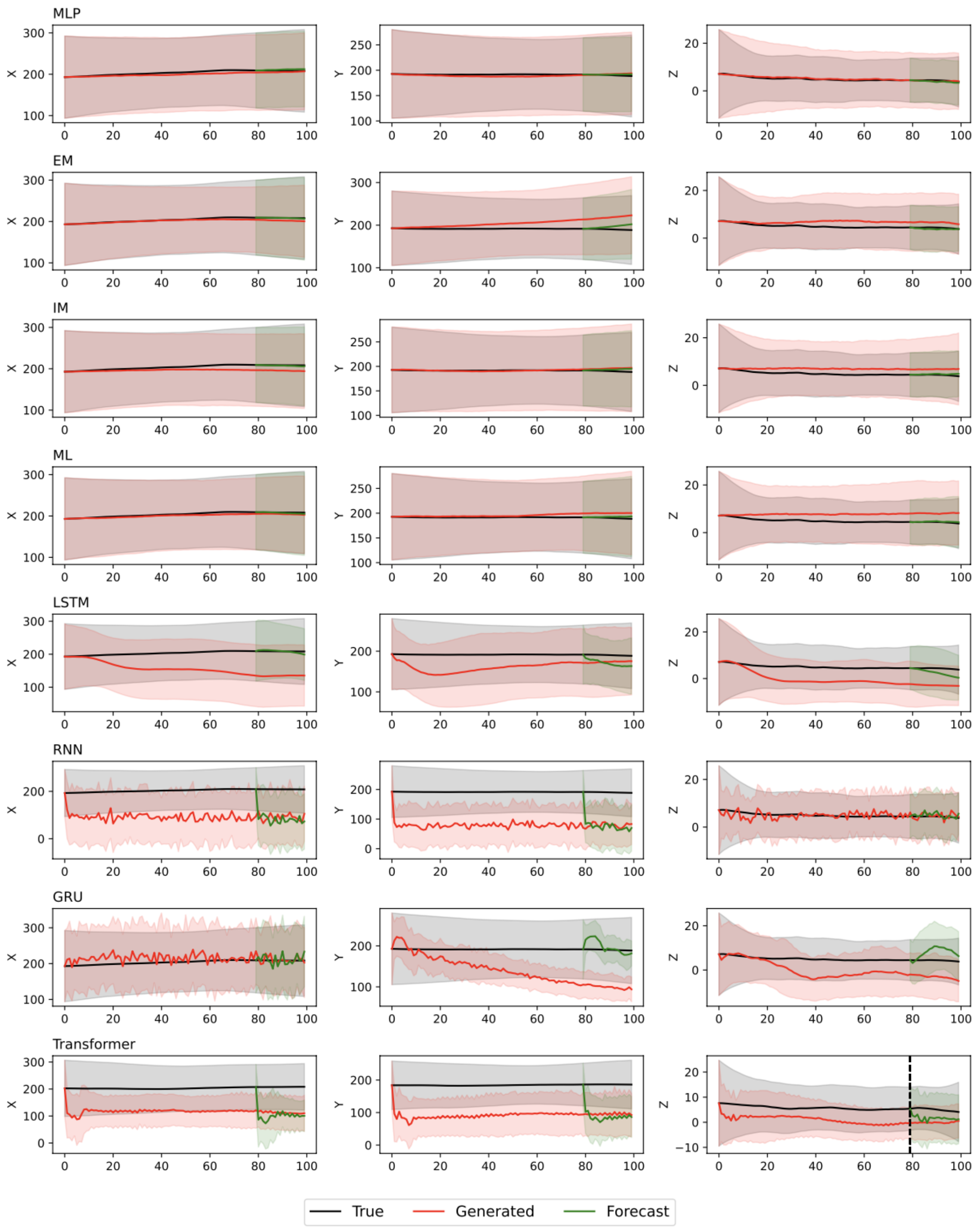


Figure 8: True, generated, and forecasted trajectories for C. Crescentus chemotaxis dataset. The dashed vertical line at $t = 80$ indicates the start of the forecast. The shaded region indicates ± one standard deviation of samples at each time step. From left to right, the columns are movements in x, y and z directions.

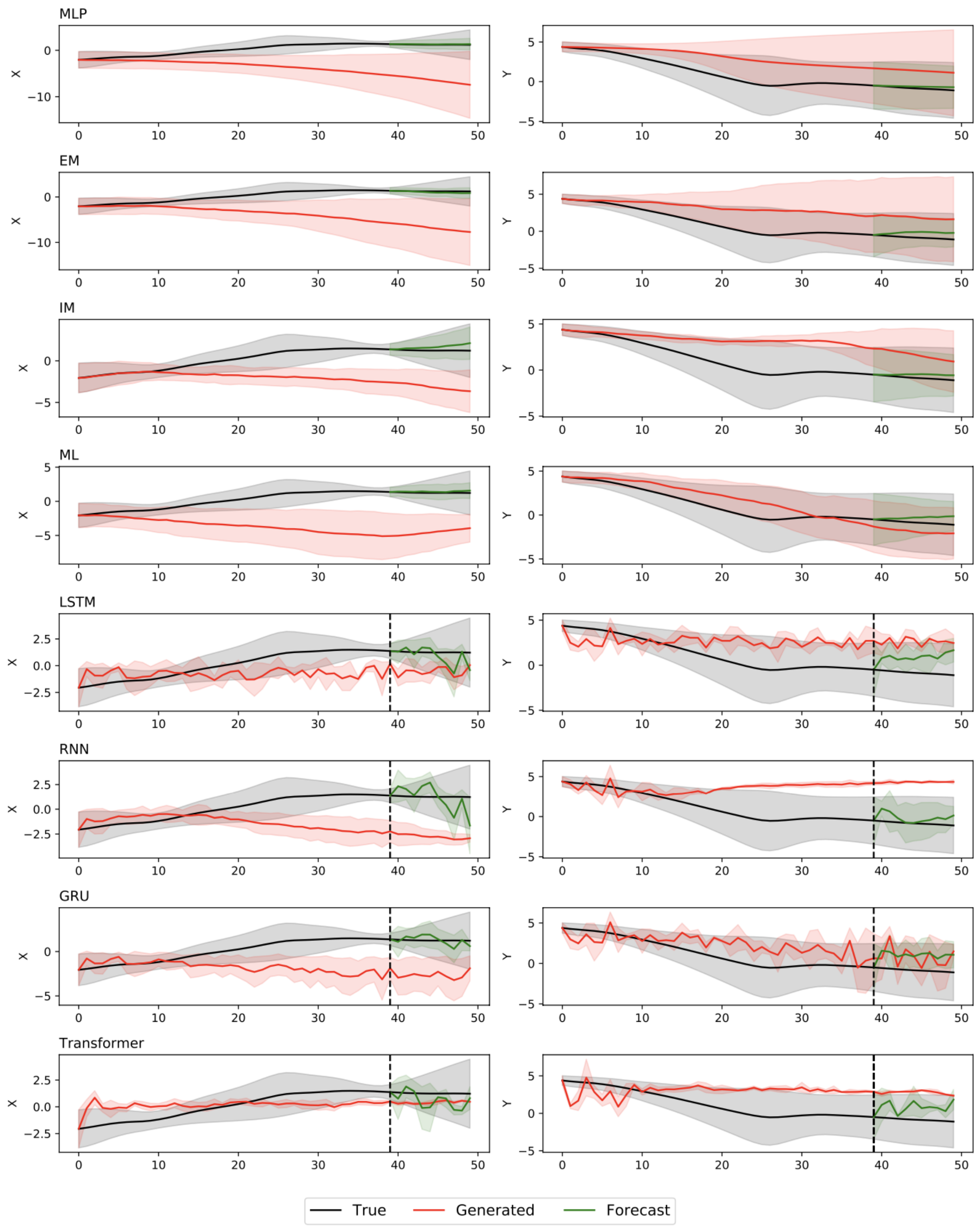


Figure 9: True, generated and forecast trajectories on crowd trajectory data. The dashed vertical line at $t = 39$ indicates the start of the forecast. The shaded region indicates $\pm$ one standard deviation of samples at each time step. From left to right, the columns are movements in x and y directions.

C.4.4 Generative modeling experiments

Figure 10 shows 5 randomly selected 2-d projections of the 100-d mixture of Gaussians.

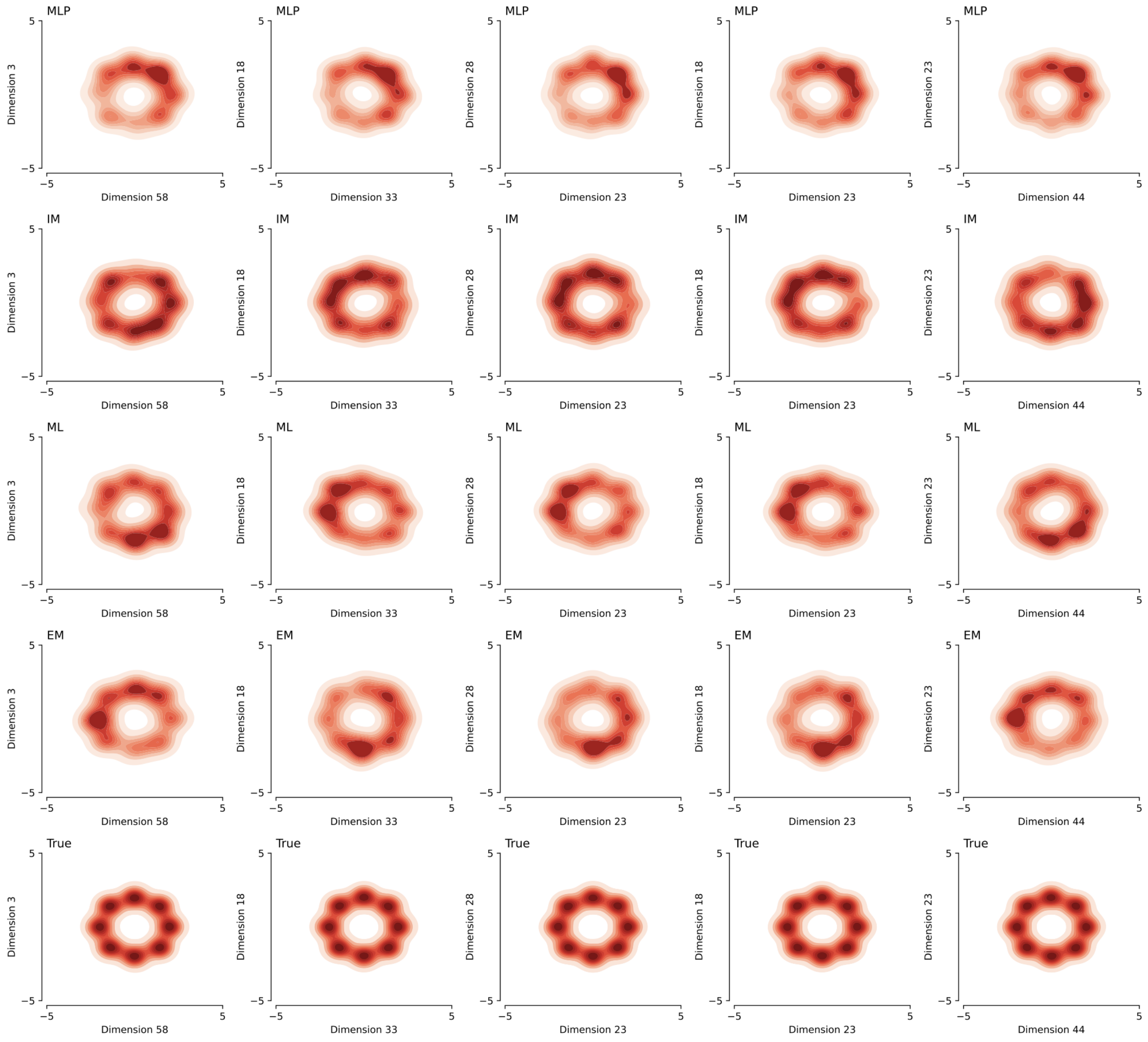
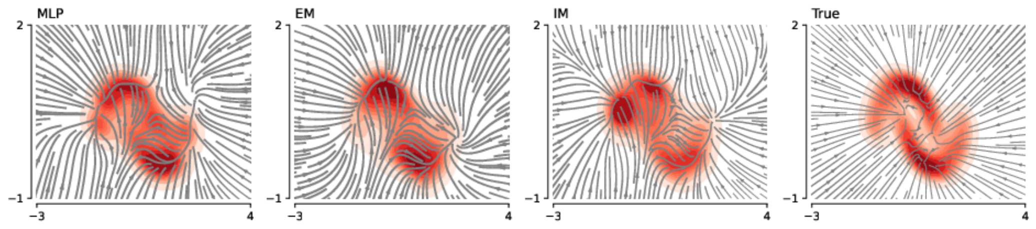
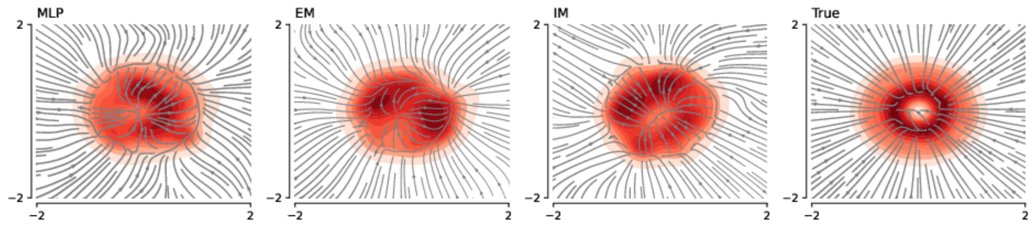


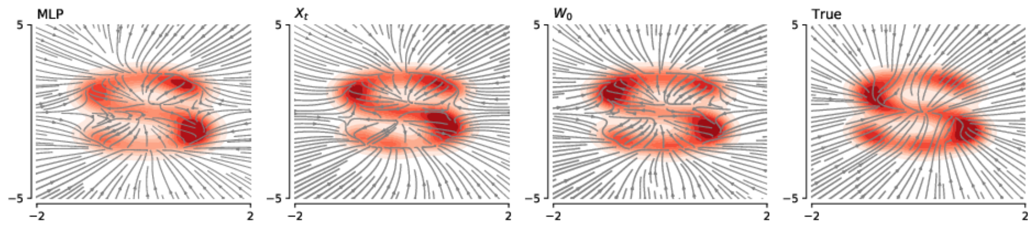
Figure 10: Five randomly selected 2-d projections of 100-d mixture of Gaussians.



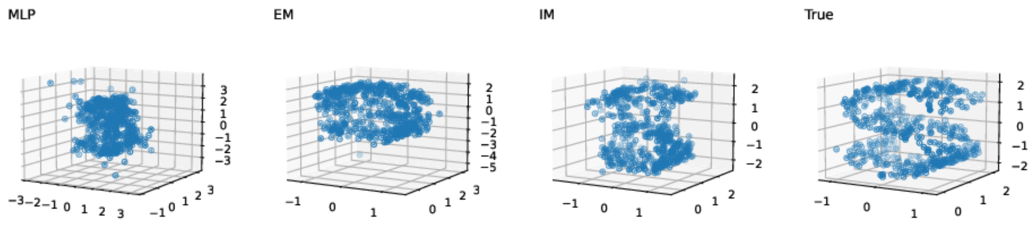
(a) Two Moons



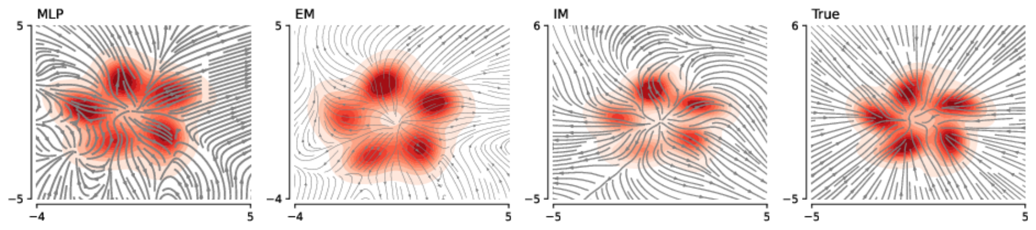
(b) Two Circles



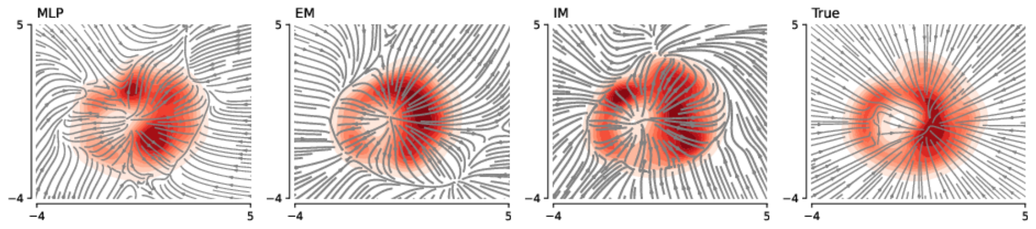
(c) S Curve, 2-d



(d) S Curve, 3-d



(e) Pinwheel



(f) Swissroll

Figure 11: Estimated gradient flow at terminal time. The colors correspond to the density of generated samples at terminal time. From left to right: MLP, EM, IM, true. From top to bottom: two moons, two circles, S-curve 2-d, S-curve 3-d, pinwheel, swissroll. The architectures were trained with the Brownian bridge estimator.

C.4.5 Additional experiments: linear Fokker–Planck

For this set of experiments, the set up is similar to the generative modeling experiments detailed in Section C.2.3 where we map between a Gaussian distribution and target distributions of two moons, two circles, s-curves, and 3-dimensional s-curves. However, we consider a linearization of the PDE that governs the density and derive a likelihood for the target distribution based on the linearized PDE. We are mainly interested in the performance differences due to differences in architectures between MLP (Itô), IM, and ML. A similar framework was considered in Huang et al. (2021) with respect to score-based generative models. We first derive the estimation procedure then show the results.

It is known that the flow satisfies the Fokker-Planck equation given by

$$\partial_t p_t = -\mathrm{div}(b(x, p_t, t)p_t(x)) + \frac{\sigma^2}{2}\nabla^2 p_t(x). \tag{21}$$

Falling back on the Itô-SDE, where b does not depend on p_t, the PDE is linear. We can then consider using the Feynman-Kac formula where the solution to (21) with b independent of p_t can be computed according to an expectation over sample paths X_t that satisfy $\mathrm{d}X_t = b(\cdot)\mathrm{d}t + \sigma \mathrm{d}W_t$ such that

$$p_T(x) = \mathbb{E}\left[\exp\left(\int_0^T -\mathrm{div} b(\cdot)\,\mathrm{d}t\right) p_0(X_T) \,\middle|\, X_0 = x\right].$$

We use Girsanov's theorem to transform the expectation over sample paths with drift to an expectation under Brownian motion, i.e. over sample paths X_t that satisfy $\mathrm{d}X_t = \sigma \mathrm{d}W_t$ with no drift and

$$p_T(x) = \mathbb{E}\left[\exp\left(\int_0^T -\mathrm{div} b(\cdot)\,\mathrm{d}t\right) p_0(X_T) \exp\left(\int_0^T b(\cdot)\,\mathrm{d}X_t - \frac{1}{2}\int_0^T b^2(\cdot)\,\mathrm{d}t\right) \,\middle|\, X_0 = x\right].$$

leading to an efficient Monte Carlo method for computing the probability. To maximize this likelihood, we can use Jensen's inequality to derive an ELBO which we optimize as

$$\log p_T(x) \geq \mathbb{E}\left[\int_0^T -\mathrm{div} b(\cdot)\,\mathrm{d}t + \log(p_0(X_T)) + \int_0^T b(\cdot)\,\mathrm{d}X_t - \frac{1}{2}\int_0^T b^2(\cdot)\,\mathrm{d}t \,\middle|\, X_0 = x\right].$$

The integrals are approximated using the forward Euler method and the parameters of b are optimized for the set of observations. The results are given in Table 17. The results suggest that the proposed architectures do not decrease performance in the linear setting and sometimes provide slight improvements.

	Two Moons	Two Circles	S Curve 2d	S Curve 3d	Pinwheels	Swissroll
MLP (Itô)	38.122 (0.517)	33.738 (0.150)	**54.083** (0.600)	*72.045* (0.688)	72.333 (1.018)	69.345 (0.717)
IM	**37.356** (0.323)	**33.160** (0.371)	54.098 (0.645)	**72.013** (0.645)	*71.318* (0.985)	**69.000** 0.550
EM	*37.793* (0.307)	*33.319 (0.264)*	*54.089* (0.607)	72.636 (0.600)	**70.692 (3.596)**	*69.230* 0.546

Table 17: Density estimation through linear Fokker-Planck training: ELBO between true samples and generated samples. **Bolded** values and *italic* values are best and second best correspondingly.

D Intuition behind the proposed architectures

Here we describe a few details regarding the architectures and how they can be interpreted. The main idea is each representation has its own set of implicit biases that impart different properties on the learned drift.

D.1 Intuition behind the IM architecture

The IM architecture has two key components: the mean field layer, which acts as a weight sharing mechanism; and, the change of measure, which reweighs the shared weight. The main idea is that by using the weight sharing component, learning the correct drift becomes "easier", leading to better estimates of the drift. We represent the shared weight as a matrix W_0 which is of dimensions $\mathbb{R}^{K\times d}$ where K is the width of the mean-field layer. We represent the interaction function φ as a neural network φ_θ. To get an understanding how how this looks like in the stationary case (that is, $p_t = p_\star$ for all t), take the following illustration:

$$X_t^{(i)} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^d, \quad W_0 = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,d} \\ \vdots & \ddots & \ddots & \vdots \\ w_{K,1} & w_{K,d} & \cdots & w_{K,d} \end{bmatrix}$$

$$\mathrm{MF}_{(K)}(\varphi_\theta(X_t^{(i)})) = \frac{1}{K}\sum_{i=1}^{K} \varphi\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}, \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,d} \end{bmatrix}\right)$$

Note that as $K \to \infty$ then this expectation can correspond to the true expectation. We then represent the change of measure by another neural network, $\lambda_\theta(\cdot)$. $\lambda_\theta(\cdot)$ takes as input the weights W_0 and time t to output the re-weighted measure.

$$\mathrm{MF}_{(K)}(\varphi(X_t^{(i)}) = \frac{1}{K}\sum_{i=1}^{K} \varphi_\theta\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}, \lambda_\theta\left(\begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,d} \end{bmatrix}, t\right)\begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,d} \end{bmatrix}\right)$$

To provide more intuition on how the mean-field layer is implemented in PyTorch, it requires the `repeat_interleave` operation to implement the weight sharing. The following code snippet illustrates the module in more detail:

```python
class MeanFieldModule(nn.Module):
    def __init__(self, d):
        self.varphi = MLP(d * 2, WIDTH, DEPTH, d)  # initialize the interaction kernel network
        self.w0     = torch.randn(WIDTH_W, d)       # initialize the shared weight

    def forward(self, x):
        x_prime   = x.repeat_interleave(WIDTH_W, dim=0) # mean-field weight sharing
        in_varphi = torch.cat((x_prime, self.w0),dim=-1)
        B_x = self.varphi(in_varphi)                    # change of measure in self.varphi
```

Note that the change of measure is implicit in the parameterization of `varphi`. We can adjust this as necessary, for example using a separate network that applies the change of measure to the `w0` parameter. Having introduced this parameterization, we note that the interesting behavior comes from the implicit regularization.

D.2 Intuition behind the ML architecture

The main idea behind the ML architecture is that if we explicitly represent the marginal distributions according to some density estimator, we can leverage the regularity of the density over different time steps to improve the estimation. We know that the flow should satisfy a nonlinear Fokker-Planck equation, so by constraining the class of densities such that they satisfy this equation, we can possibly do better than just using the empirical measure. Taking an expectation with respect to the empirical measure — particularly in cases where it may be corrupted by noise — can lead to an incorrect drift. This is particularly important when one is dealing with scientific data which may be noisy.

Instead, we parameterize a time dependent density and ensure there is an agreement between the estimated drift and the estimated density.