
Neural McKean-Vlasov Processes: Inferring Distributional Dependence

Anonymous Author(s)

Affiliation

Address

email

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. These processes differ from standard Itô-SDEs to the extent that MV-SDEs include distributional information in their individual particle parameterization. As such, we study the influence of explicitly including distributional information in the parameterization of the SDE. We first propose a series of semi-parametric methods for representing MV-SDEs, and then propose corresponding estimators for inferring parameters from data based on the underlying properties of the MV-SDE. By analyzing the properties of the different architectures and estimators, we consider their relationship to standard Itô-SDEs and consider their applicability in relevant machine learning problems. We empirically compare the performance of the different architectures on a series of real and synthetic datasets for time series and probabilistic modeling. The results suggest that including the distributional dependence in MV-SDEs is an effective modeling framework for temporal data under an exchangeability assumption while maintaining strong performance for standard Itô-SDE problems 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. Beginning with Itô-SDEs, where individual sample paths are assumed to be independent, neural representations of the drift and diffusion have achieved high performance in many applications, such as time series and generative modeling [Song et al., 2020, Tashiro et al., 2021].

On the other hand, interacting particle systems are also used to model stochastic processes using many of the same characteristics as an Itô-SDE, but they additionally dictate an interaction between the different sample paths [Liggett, 1997]. 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 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

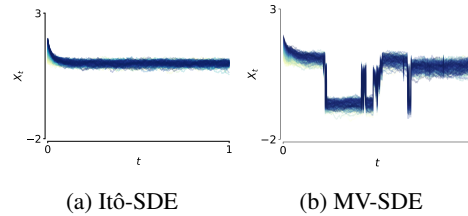


Figure 1: SDE sample paths of a double-well potential, where the particles (a) do not interact and (b) exhibit complex phase transitions as a result only of interaction via weak attraction.

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 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 problem of estimating parameters from observations or their application in machine learning tasks.

This brings us to a motivating question:

(Q1) *Can we develop theoretically justified neural architectures to represent MV-SDEs?*

To answer (Q1), we use the relationship between the approximation capabilities of neural networks and properties of MV-SDEs. We consider two ideas: (i) expressing a layer in a neural network as an expectation with respect to a density and (ii) using generative models to capture distributions and generate samples.

Our second question relates the theoretical generality of MV-SDEs to Itô-SDEs:

(Q2) *Does including explicit distributional dependence empirically affect modeling capabilities?*

We discuss a few theoretical properties that motivate this question and answer the question empirically by comparing different architectures for applications in time series and in probabilistic modeling.

1.1 Related work

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 Zanon [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. 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] considered a machine learning approach for solving certain kinds of mean field games and mean field control problems. 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 existing 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 three 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 k th moment. Let $W_t \in \mathbb{R}^d$ be a d -dimensional Wiener process and let $X_t \in \mathbb{R}^d$ be a

84 solution to the following MV-SDE

$$dX_t = b(X_t, p_t, t)dt + \sqrt{\Sigma(X_t, p_t, t)}dW_t \quad (1)$$

85 where p_t denotes the law of X_t at time t and $\sqrt{\Sigma}$ denotes the Cholesky decomposition of Σ . We
 86 assume that the *drift* vector $b : \mathbb{R}^d \times \mathcal{P}_2(\mathcal{D}) \times \mathbb{R}_+ \rightarrow \mathbb{R}^d$ and the *diffusion* matrix $\Sigma : \mathbb{R}^d \times \mathcal{P}_2(\mathcal{D}) \times$
 87 $\mathbb{R}_+ \rightarrow \text{SPD}(\mathbb{R}^{d \times d})$ are globally Lipschitz for the existence and uniqueness of the solution, with SPD
 88 denoting the space of symmetric, positive definite matrices.

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

$$dX_t = f(X_t, t)dt + \mathbb{E}_{y_t \sim p_t} [\varphi(X_t, y_t)]dt + \sigma dW_t \quad (2)$$

93 where $f : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}^d$ can be seen as the Itô drift, the expectation as the mean-field drift, and
 94 $\varphi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^k$ as the *interaction* function describing the interaction between particles, e.g.
 95 attraction with $\varphi(x, y) = -(x - y)$ in Figure 1b and the left side of Figure 2. We also assume that
 96 all coefficients exhibit sufficient regularity such that the empirical law converges to the true law of
 97 the system (i.e. $\frac{1}{N} \sum_{i=1}^N \delta_{X_t^{(i)}} \rightarrow_{N \rightarrow \infty} p_t(X_t)$), i.e. propagation of chaos holds [Méléard, 1996].
 98 As mentioned, unlike Itô-SDEs which only consider dependence on X_t and t , MV-SDEs also depend
 99 on the marginal time distribution p_t . By introducing a dependence on the marginal law, the transition
 100 density of the process satisfies a richer class of functions.

101 2.2 Non-locality of the transition density

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

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

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

110 2.3 Discontinuous sample paths

111 The richer class of densities modeled by MV-
 112 SDEs has direct influence on individual sample
 113 paths. In a modeling scenario, we may wish to
 114 approximate a process that exhibits jumps. For
 115 example, in finance, a number of related entities
 116 may have common exposure and experience failure
 117 simultaneously [Nadtochiy and Shkolnikov,
 118 2019, Feinstein and Søjmark, 2021]. Similarly,
 119 in neuroscience, a number of neurons spiking
 120 simultaneously results in discontinuities in the
 121 sample paths [Carrillo et al., 2013]. The fact
 122 that the interaction of many particles can cause blowups leads to a remarkable property of MV-SDEs
 123 that allows discontinuous paths. The major benefit of this property is that we do not need to consider
 124 an additional jump noise process – we only need to specify a particular interaction between the
 125 particles to induce the jump behavior. A simple proof for the case of positive feedback is given
 126 in Hambly et al. [2019, Theorem 1.1].

127 Having described the theoretical advantages of MV-SDEs as compared to Itô-SDEs, we will proceed
 128 to discuss the neural architectures for representing these processes.

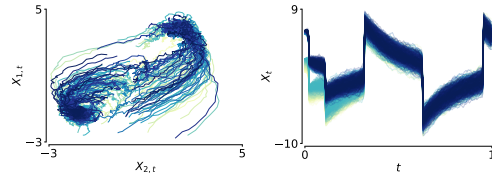


Figure 2: MV-SDE sample paths with non-local dynamics (left) and discontinuities (right).

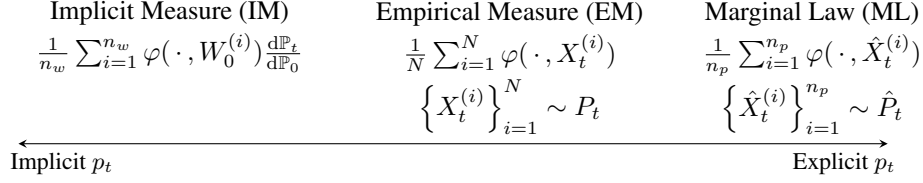


Figure 3: Schematic comparing neural architectures for modeling MV-SDEs. Implicit measure (IM) architecture uses a mean-field layer that represents particles as learned weights; the empirical measure (EM) architecture computes the expectation with the observed particles; the marginal law (ML) estimates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

3 Mean-field Architectures

We now describe methods for representing the mean-field drift of a MV-SDE in (2). We first consider a modification of the cylindrical architecture [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, denoted as the implicit measure (IM) architecture, and a generative architecture based on representing the marginal law of the samples (ML). Figure 3 provides a schematic of the different architectures and mean-field representations. We denote a function f parameterized by parameters θ as $f(\cdot; \theta)$.

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

$$\mathbb{E}_{y_t \sim p_t} [\varphi(X_t, y_t)] \approx \mathbb{E}_{y_t \sim p_t^\delta} [\varphi(X_t, y_t; \theta)] = \frac{1}{N} \sum_{i=1}^N \varphi(X_t, X_t^{(i)}; \theta) \quad (4)$$

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 true MV-SDE drift in the limit as the number of observations $N \rightarrow \infty$. 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

While the EM architecture in (4) explicitly defines the relationship between the law p_t and the interaction φ , it 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, let us first recall that a single layer in a multilayer perceptron (MLP) can be written in terms of an expectation as

$$\text{MLP}^{W,b}(x) = \int \sigma(Wx + b) d\nu(W, b) \quad (5)$$

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 i th row as $W_0^{(i)}$. The mean-field layer then is defined by the

161 operation

$$\text{MF}_{(n)}(\varphi(X_t)) := \frac{1}{n} \sum_{i=1}^n \varphi(X_t, W_0^{(i)}) \frac{d\mathbb{P}_t}{d\mathbb{P}_0}. \quad (6)$$

162 The mean-field layer (MF) can be thought of as another layer within the network architecture that
 163 approximates the law p_t . Each row $W_0^{(i)}$ is of size R^d , corresponding to the dimensions of $X_t^{(i)} \in \mathbb{R}^d$.
 164 The activation function of the mean-field layer is the average over the augmented dimension over
 165 which MF operates. The change of measure $\frac{d\mathbb{P}_t}{d\mathbb{P}_0}$ can be learned as part of the estimator of the
 166 interaction function, $\varphi(\cdot, \cdot, t; \theta)$. Importantly, the above representation allows modeling mean-field
 167 interactions without the need for a full set of observations at each time point and without the need to
 168 explicitly represent the distribution p_t at each time point. Assuming that φ and MF are well learned,
 169 this architecture can represent the true MV-SDE drift in the limit as the width $n \rightarrow \infty$. We note
 170 empirically that a finite n is sufficient and we provide examples of ablations in the appendix.

171 A similar analysis can be made for the standard MLP architecture. However, the explicit separation
 172 of f and φ is not enforced in this case. This leads us to the following remark:

173 **Remark 3.2.** *[Itô-SDEs with drift represented using MLPs can model MV-SDEs] From the above*
 174 *discussion, the expectation with respect to the law p_t may be implicitly represented by a MLP.*

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

179 3.3 Marginal law architecture

180 A solution to the MV-SDE is the pair (X, p) such that $p_t = \text{Law}(X_t)$. In addition, if p is a solution
 181 to the SDE in (2), it is also a weak solution to the PDE in (3), and the converse holds. For this reason,
 182 p is often itself the main object of study. In the marginal law (ML) architecture, in conjunction with
 183 the drift, we introduce a generative model for representing the time-varying density. In this case, we
 184 approximate the expectation in (2) as

$$\mathbb{E}_{y_t \sim p_t} [\varphi(X_t, y_t)] \approx \mathbb{E}_{y_t \sim \hat{P}_t} [\varphi(X_t, y_t; \theta)] = \frac{1}{n} \sum_{i=1}^n \varphi(X_t, \hat{X}_t^{(i)}; \theta) \quad (7)$$

185 where the expectation is taken with respect to the discrete measure derived from samples $\{\hat{X}_t^{(i)}\}_{i=1}^n$
 186 from the generative model \hat{P}_t . The parameter estimation problem then requires optimizing both the
 187 generative model \hat{P}_t and the networks f and φ representing the drift, while ensuring consistency
 188 between the two. Using knowledge of the PDE in (3), we regularize \hat{P}_t such that it matches the flow
 189 relating to the drift. Additional details regarding the PDE and its relationship to the ML architecture
 190 are in the appendix.

191 4 Parameter Estimation

192 Having presented the relevant architectures, we now describe the procedures for estimating the
 193 parameters of the different architectures. We first describe the likelihood function for use in cases
 194 with regularly sampled data. We then describe a bridge estimator for cases of irregularly sampled
 195 data. Finally, we describe an estimator for the generative architecture based on both the likelihood
 196 function and the transition density. For this section, we assume that we observe multiple paths, i.e.,
 197 $\left\{ \{X_{t_j}\}_{j=1 \dots K}^{(i)} \right\}_{i=1 \dots N}$. Full details of all algorithms are in the appendix.

198 4.1 Maximum likelihood estimation

199 We use an estimator based on the path-wise likelihood derived from Girsanov's theorem and an
 200 Euler-Maruyama discretization for the likelihood, considered in Sharrock et al. [2021]. The likelihood

function is given as

$$\mathcal{L}(\theta; t_1, t_K) := \exp \left(\frac{1}{\sigma^2} \int_{t_1}^{t_K} b(X_s, p_s, s; \theta) dX_s - \frac{1}{2\sigma^2} \int_{t_1}^{t_K} b(X_s, p_s, s; \theta)^2 ds \right). \quad (8)$$

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

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

If the time interval Δt is large, then this likelihood loses accuracy, as is a property of the Euler-Maruyama discretization. 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 following 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

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

The ELBO in this case aims to fit the observed marginal distributions exactly while penalizing deviations in regions without data that deviate from the Brownian bridge paths.

4.3 Estimation with explicit marginal law \hat{P}_t

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 regularity of p_t through its PDE in (3). Let the parameters of the drift be θ and the parameters of the generative model be ϕ , then we solve the optimization problem

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

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

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

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

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.

5.1 Implicit regularization of the implicit measure architecture

Closely related to the IM architecture are neural Itô-SDEs, where we previously remarked can model MV-SDEs. On the other hand, the factorization of the IM architecture into φ and MF leads to a type of implicit regularization when the parameters are estimated using gradient descent.

Proposition 5.1 (Implicit Regularization). *Suppose f , φ 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 i th row $W_0^{(i)}$ under gradient descent satisfies the following optimization problem*

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

Proof. We follow the blueprint in Belabbas [2020] and give full details in the appendix. \square

Proposition 5.1 effectively says that the mean-field system approximated is the one that has the least influence from the other particles under perfectly matched marginals. 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 MV-SDE and Itô-SDE particle flows, we invoke the analysis in Santambrogio [2017, Section 4.6] to describe the functionals that are minimized by each.

Remark 5.2 (Functional Minimizer). *Consider two drifts $B = \nabla f(X)$ and $B_{MF} = B + \mathbb{E}[\nabla \varphi(X - y)]$. Consider a functional $F[p] = \int \log p dp + \int f(X) dp$ for some measure p absolutely continuous with respect to the Lebesgue measure. Then, the gradient flow satisfying the linear Fokker-Planck equation with drift B minimizes F . On the other hand, the nonlinear Fokker-Planck associated with drift B_{MF} minimizes the functional $F_{MF}[p] = F[p] + \int \varphi(X - Y) dp(X) dp(Y)$.*

This has an important implication, for example, if we take $\varphi(\cdot) = 2\|\cdot\| \frac{dq}{dp} - \|\cdot\|^2 - \|\cdot\|^2 \left(\frac{dq}{dp}\right)^2$ then the functional is minimizing the squared energy distance between a target measure q as well as the entropy. We use this example to motivate some of the experiments on probabilistic modeling.

6 Numerical Experiments

We discussed QI on modeling and inferring distributional dependence. We now wish to answer $Q2$ and quantify the effect of distributional dependence in machine learning tasks. To do this, we test the methods on synthetic and real data for time series estimation and sample generation. 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 Empirical Measure (EM) in Section 3.1, Implicit Measure (IM) in Section 3.2, and Marginal Law (ML) in Section 3.3. 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

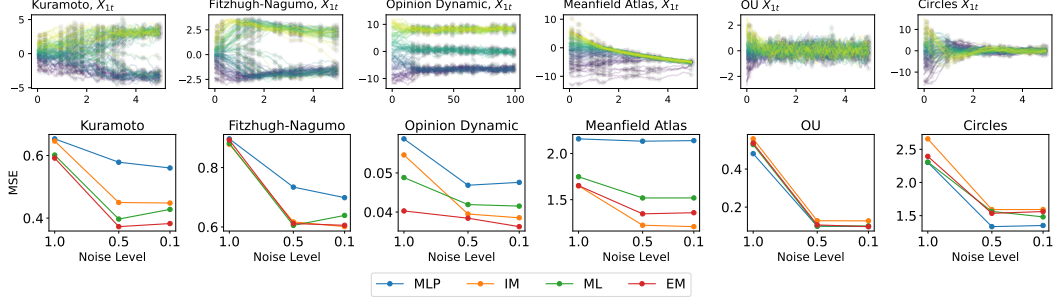


Figure 4: Top row: sample paths from the different synthetic datasets. Bottom row: mean squared error (MSE) of different architectures' performance on drift estimation, under the effect of different levels of observation noise. Reported value is an average of 10 runs.

[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. The performance on five different datasets with three different levels of added observational noise is presented in Figure 4. The proposed mean-field architectures outperform the standard MLP in modeling MV-SDEs; moreover, our experiments on OU and circular process suggest that incorporating explicit distributional dependence does not diminish the performance in estimating non-interacting Itô-SDEs. When modeling processes with jump discontinuities, Figure 5 highlights the flexibility of the proposed methods, IM, ML, to match such 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.

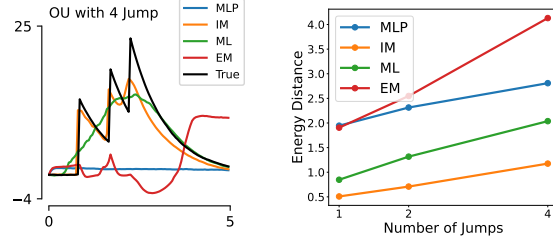


Figure 5: Left: Average paths of true and estimated OU process with 4 jumps. Right: Energy distance between true and generated paths.

Real data experiments Extending from the synthetic examples, we consider two real examples: brain activity recorded by electroencephalograms (EEG), which is closely related to the Kuramoto model [Nguyen et al., 2020]; 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. We compute the Continuous Ranked Probability Score (CRPS) defined in Gneiting and Raftery [2007] (see appendix for details) for the 1-dimensional EEG data, and the normalized MSE (normalized with sample variance) for the 3-dimensional chemotaxis data 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 EEG, the proposed architectures generally perform better than the baselines in generating paths within the training time steps, and on par with the DeepAR architectures for forecasting (full results presented in appendix). For chemotaxis data, the MV-SDE based architectures all outperform the DeepAR baselines.

Table 1: Time series estimation on held out trajectories. NA/A stands for non-alcoholics/alcoholics. **Bolded** values and *italic* values are best and second best respectively.

| | CRPS ↓ | | MSE ↓ | |
|-----------|--------------------|--------------------|----------------------|----------------------|
| | NA-EEG | A-EEG | C.Cres | E.Coli |
| MLP (Itô) | 5.52 (1.40) | 4.33 (1.14) | 0.096 (0.002) | <i>0.080</i> (0.003) |
| IM | 5.23 (1.24) | 4.30 (1.21) | 0.094 (0.003) | 0.080 (0.001) |
| ML | 5.10 (1.22) | 4.05 (1.12) | 0.093 (0.002) | 0.084 (0.002) |
| EM | 5.35 (1.22) | <i>4.09</i> (1.11) | <i>0.093</i> (0.004) | 0.086 (0.004) |
| LSTM | 6.27 (2.02) | 5.68 (2.56) | 1.159 (0.234) | 0.585 (0.350) |
| RNN | 6.22 (2.07) | 4.64 (1.38) | 1.563 (1.070) | 0.773 (0.092) |
| GRU | 6.35 (2.01) | 6.18 (2.73) | 0.826 (0.289) | 0.568 (0.301) |
| TR | 5.95 (1.45) | 4.29 (1.36) | 1.503 (0.212) | 1.204 (0.212) |

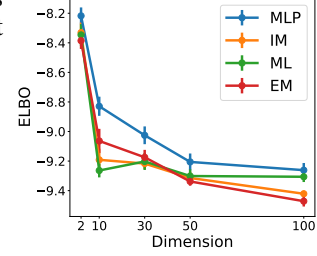


Figure 6: ELBO of generated paths from standard Gaussian to eight Gaussian mixture (in increasing dimension) evaluated against OT mapping.

Generative modeling experiments We focus on applying the bridge estimator discussed in Section 4.2 to map between a Gaussian and a target distribution. 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 through 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 (see appendix for details) 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 (W-GAN) [Gulrajani et al., 2017], masked autoregressive flow (MAF) [Papamakarios et al., 2017] and score-based generative modeling through SDEs, which corresponds to a constrained form of the MLP [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. All sampling is performed using standard Euler-Maruyama, with full details of the sampling and inference algorithms in the appendix. This again suggests the MV-SDE provides a more amenable probability flow for modeling compared with the Itô case.

Table 2: Density estimation: Energy distance between observed samples and generated samples of different methods. **Bolded** values and *italic* values are best and second best correspondingly.

| | POWER | MINIBOONE | HEPMASS | GAS | CORTEX |
|-------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| MLP (Itô) | 0.342 (0.096) | 0.674 (0.048) | 0.537 (0.052) | 0.405 (0.08) | 0.742 (0.062) |
| IM | 0.292 (0.078) | 0.395 (0.045) | 0.405 (0.025) | 0.287 (0.082) | 0.53 (0.026) |
| ML | 0.282 (0.083) | <i>0.443</i> (0.034) | <i>0.366</i> (0.03) | 0.305 (0.063) | 0.568 (0.03) |
| EM | 0.328 (0.116) | 0.455 (0.036) | 0.429 (0.046) | <i>0.298</i> (0.036) | 0.577 (0.037) |
| VAE | 1.19 (0.024) | 2.117 (0.148) | 1.763 (0.031) | 1.516 (0.023) | 2.412 (0.197) |
| W-GAN | 1.248 (0.017) | 2.079 (0.003) | 1.819 (0.013) | 1.3 (0.016) | 2.19 (0.011) |
| MAF | <i>0.288</i> (0.041) | 0.467 (0.009) | 0.308 (0.017) | 0.519 (0.033) | <i>0.532</i> (0.026) |
| Score-Based | 0.302 (0.049) | 0.499 (0.019) | 0.324 (0.028) | 0.562 (0.043) | 0.582 (0.020) |

7 Discussion

In this paper we discuss an alternative viewpoint of the standard Itô-SDE parameterization. In particular, we focus on MV-SDEs and discuss how neural networks can represent a process that depends on the distribution, and we describe 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 over baselines in certain generative modeling and time series applications.

Limitations We only studied the implicit regularization of the IM architecture under gradient descent, but the extension of the analysis to the other proposed architectures is important to understand the corresponding regularization. Additionally, computing expectations incurs additional computational cost. Improving the computational accuracy using a multilevel scheme as proposed in Szpruch et al. [2019] could improve the performance of the methods.

References

- 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, Arnū 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.
- 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.
- 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.
- Tilman 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.

388 Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville.
389 Improved training of wasserstein gans. *Advances in neural information processing systems*, 30,
390 2017.

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

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

396 Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):
397 251–257, 1991.

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

401 Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based
402 generative models and score matching. *Advances in Neural Information Processing Systems*, 34:
403 22863–22876, 2021.

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

406 Evelyn F. Keller and Lee A. Segel. Model for chemotaxis. *Journal of Theoretical Biology*, 30(2):
407 225–234, 1971.

408 Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions.
409 In *Advances in Neural Information Processing Systems*, 2018. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.1807.03039)
410 [48550/arXiv.1807.03039](https://doi.org/10.48550/arXiv.1807.03039).

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

413 Daniel Lacker. Mean field games and interacting particle systems. 2018. URL [http://www.](http://www.columbia.edu/~dl3133/MFGSpring2018.pdf)
414 [columbia.edu/~dl3133/MFGSpring2018.pdf](http://www.columbia.edu/~dl3133/MFGSpring2018.pdf).

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

417 Thomas M. Liggett. Stochastic models of interacting systems. *The Annals of Probability*, 25(1):1 –
418 29, 1997. doi: 10.1214/aop/1024404276. URL <https://doi.org/10.1214/aop/1024404276>.

419 Sylvie Méléard. Asymptotic behaviour of some interacting particle systems; mckean-vlasov and
420 boltzmann models. *Probabilistic models for nonlinear partial differential equations*, pages 42–95,
421 1996.

422 Daniel A. Messenger and David M. Bortz. Learning mean-field equations from particle data using
423 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/](https://www.sciencedirect.com/science/article/pii/S0167278922001543)
424 [article/pii/S0167278922001543](https://www.sciencedirect.com/science/article/pii/S0167278922001543).
425

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

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

430 Phuong Thi Mai Nguyen, Yoshikatsu Hayashi, Murilo Da Silva Baptista, and Toshiyuki Kondo.
431 Collective almost synchronization-based model to extract and predict features of EEG signals. *Sci-*
432 *entific Reports*, 10(1):16342, 2020. URL <https://doi.org/10.1038/s41598-020-73346-z>.

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

- 435 Grigorios A Pavliotis and Andrea Zanoni. A method of moments estimator for interacting particle
436 systems and their mean field limit. *arXiv preprint arXiv:2212.00403*, 2022.
- 437 Huy  n Pham and Xavier Warin. Mean-field neural networks: learning mappings on wasserstein
438 space. *arXiv preprint arXiv:2210.15179*, 2022.
- 439 Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine
440 learning framework for solving high-dimensional mean field game and mean field control problems.
441 *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.
- 442 David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic
443 forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):
444 1181–1191, 2020.
- 445 Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyr  . Sinkformers: Transformers with
446 doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*,
447 pages 3515–3530. PMLR, 2022.
- 448 Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of*
449 *Mathematical Sciences*, 7(1):87–154, 2017.
- 450 Louis Sharrock, Nikolas Kantas, Panos Parpas, and Grigorios A Pavliotis. Parameter estimation for
451 the mckean-vlasov stochastic differential equation. *arXiv preprint arXiv:2106.13751*, 2021.
- 452 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben
453 Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint*
454 *arXiv:2011.13456*, 2020.
- 455 Bernard Sonnenschein and Lutz Schimansky-Geier. Approximate solution to the stochastic kuramoto
456 model. *Physical Review E*, 88(5), nov 2013.
- 457 Lukas Szpruch, Shuren Tan, and Alvin Tse. Iterative multilevel particle approximation for mckean-
458 vlasov sdes. *The Annals of Applied Probability*, 29(4):2230–2265, 2019.
- 459 Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffu-
460 sion models for probabilistic time series imputation. *Advances in Neural Information Processing*
461 *Systems*, 34:24804–24816, 2021.
- 462 Milica Toma  evi  . A new mckean-vlasov stochastic interpretation of the parabolic-parabolic keller-
463 segel model: The two-dimensional case. *The Annals of Applied Probability*, 31(1):432–459,
464 2021.
- 465 Rentian Yao, Xiaohui Chen, and Yun Yang. Mean-field nonparametric estimation of interacting
466 particle systems. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth*
467 *Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages
468 2242–2275. PMLR, 02–05 Jul 2022. URL <https://proceedings.mlr.press/v178/yao22a.html>.
469
- 470 Xiao Lei Zhang, Henri Begleiter, Bernice Porjesz, Wenyu Wang, and Ann Litke. Event related
471 potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.

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 consider the theoretical advantages of the proposed architectures and estimators. Specific to the architectures, we develop the implicit measure architecture, and 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 can 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

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

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 $\nu(b) = \text{Law}(-X_t)$. Since we assumed X_t is stationary, $\text{Law}(X_t) = \text{Law}(X_*)$ for all t . We now have our approximation as

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

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 \mathbb{P}_t to \mathbb{P}_{t+1} . Since we are assuming that the diffusion is constant, all measures \mathbb{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 time variable

$$\mathbb{E}_{y \sim \mathbb{P}_t}[\varphi(x - y)] = \mathbb{E}_{y \sim \mathbb{Q}} \left[\varphi(x - y) \frac{d\mathbb{P}_t}{d\mathbb{Q}} \right].$$

Under this formulation \mathbb{Q} is the learned measure and \mathbb{P}_t is the measure at each time point t . Assuming that the function $\varphi(\cdot; \theta) \frac{d\mathbb{P}_t}{d\mathbb{Q}}$ can be learned for all t as another MLP $\hat{\varphi}(\cdot, t; \theta)$, 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.

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 $W_K, W_V, W_Q \in \mathbb{R}^{N_w \times d}$ and the normalized attention matrix by

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

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

We can rewrite the above equation as an expectation

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

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,

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

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

$$dX_t = \{f(X_t, t) + \mathbb{E}_{y_t \sim p_t}[\varphi(X_t, y_t)]\} dt + dW_t.$$

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 k th row of a $w \times d$ weight matrix θ . We show that the path preferred by gradient descent is the one that minimizes $\mathbb{E}_{y_t \sim \nu}[\varphi(X_t, y_t)]$, 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:

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

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

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

529 for some function K that satisfies the PDE:

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

where g denotes the dynamics of gradient descent given as

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

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

We compute the first derivative

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

533 Then the second derivative as

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

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

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

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

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

535 In the context of the MV-SDE, the mean-field system approximated is the one that has the least
 536 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

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

537 The regularized problem has a similar form of

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

538

□

539 A.2.1 Comparison to Optimal Transport

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

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

541 such that $p(x, 0) = p_0(x)$ and $p(x, T) = p_T(x)$. Suppose that φ can be represented as a norm
542 $\|g(x, y)\|^2$ and replace the drift with the one given by the implicit bias, the PDE then becomes

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

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

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

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

546 subject to

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

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

552 A.3 Compatibility Criterion in Inferring Explicit Distributional Dependence

553 A.3.1 Feynman-Kac for the Kolmogorov Backward Equation

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

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

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

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

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

566 where $p_T(\cdot)$, $t < T$ is the given terminal condition and X_s satisfies the SDE $dX_s = b(\cdot)ds + \sigma dW_s$.

567 Following (20), we evolve X_s from $X_t = x$ to X_T , then penalize the difference between $p_t(x)$ and
568 $\mathbb{E}[p_T(X_T) \mid X_t = x]$. The estimation algorithm with this compatibility criterion on the marginal
569 density is detailed in Algorithm 3.

570 A.3.2 Feynman-Kac Analysis of the Brownian Bridge Estimator

Consider the bridge estimator

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

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

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

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

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

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

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

- EM architecture: $\frac{1}{N} \sum_{k=1}^N \varphi(X_{t_j}^{(i)}, X_{t_j}^{(k)}; \theta)$.
- IM architecture: $\frac{1}{n_w} \sum_{k=1}^{n_w} \varphi(X_{t_j}^{(i)}, W_0^{(k)}, t_j; \theta)$
as a neural network with an additional layer and additional conditioning on t_j .
- ML architecture: $\frac{1}{n_p} \sum_{k=1}^{n_p} \varphi(X_{t_j}^{(i)}, \hat{X}_{t_j}^{(k)}; \theta)$
where we compute the expectation with samples $\{\hat{X}_{t_j}^{(k)}\}_{k=1}^{n_p}$ from $\hat{P}(\cdot, t_j; \theta)$,
a generative network with additional conditioning on t_j .

Additional details on the parameterization of the neural architectures are in Section C.2.3.

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].

We finally describe a sampling algorithm.

Algorithm 2 MLE with Girsanov and Brownian Bridges

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

Algorithm 4 Sampling Trajectories with Euler-Maruyama Scheme

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

Following Gneiting and Raftery [2007], the Continuous Ranked Probability Score (CRPS) is given by

$$\text{CRPS}(F, x) = \int_{-\infty}^{\infty} [F(y) - \mathbb{1}(y \geq x)]^2 dy.$$

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.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

$$dX_t^{(i)} = \left[h^{(i)} + \frac{K}{N} \sum_{j=1}^N \sin(y_t^{(j)} - X_t^{(i)}) \right] dt + \sigma dW_t^{(i)},$$

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-dimensional trajectories with $X_t^{(i)} = [X_{1t}^{(i)}, X_{2t}^{(i)}] \in \mathbb{R}^2$, $h^{(i)} = [\sin(X_{1t}^{(i)}), \sin(X_{2t}^{(i)})]$, $K = 2$, $N=20$, 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

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

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

$$dX_t = \mathbb{E}[\psi_\theta(\|X_t - y_t\|)(X_t - y_t)] + \sigma dW_t,$$

where $\psi_\theta(r) = \theta_1 \exp(-\frac{0.01}{1-(r-\theta_2)^2})$. We simulate 2-dimensional 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

$$dX_t = \gamma \left(\int \mathbb{1}_{\{X_t - y_t > 0\}} dp_t(y) \right) dt + \sigma dW_t,$$

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

Itô Diffusion - Ornstein-Uhlenbeck. We simulated a 2-dimensional Ornstein-Uhlenbeck (OU) process with drifts $[-3X_{1t}, -2X_{2t}]$.

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

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

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 | 20 | 20 |
| Fitzhugh-Nagumo | 0.3 | 5 | 0.05 | | |
| Opinion Dynamic | 0.5 | 100 | 1.0 | | |
| Mean-field Atlas | 1 | 5 | 0.05 | | |
| Ornstein-Uhlenbeck | 1 | 5 | 0.05 | | |
| Circles | 1 | 5 | 0.05 | 100 | Not Applicable |
| OU with Jumps | 1 | 5 | 0.05 | | |

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.

Chemotaxi Data. We used the 3-dimensional Chemotaxi data provided by Grognot and Taute [2021]. We used V_{0208} for *C.Crescentus* and V_{MeAspI_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, 6 and 7 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, 10 and 200 for the synthetic time series, EEG, Chemotaxis and generative modeling experiments respectively. The models were trained for 500, 500, 2000 and 500 epochs for the synthetic time series, EEG, Chemotaxis and generative modeling experiments.

| Architecture | Modules: Hidden Layers | Layer Size | Activation | # of Parameters |
|--------------|----------------------------------|---------------|------------|-----------------|
| MLP (Itô) | 8 | 128 | | 132740 |
| EM | $\varphi: 4, f: 4$ | 128, 128 | LeakyReLU | 133638 |
| IM | $\varphi: 4, f: 4, W_0: 1$ | 128, 128, 128 | | 134022 |
| ML | $\varphi: 4, f: 4, \hat{P}_t: 1$ | 128, 128, 32 | | 136152 |
| MLP (Itô) | 4 | 128 | | 66820 |
| EM | $\varphi: 2, f: 2$ | 128, 128 | LeakyReLU | 67718 |
| IM | $\varphi: 2, f: 2, W_0: 1$ | 128, 128, 128 | | 67846 |
| ML | $\varphi: 2, f: 2, \hat{P}_t: 1$ | 128, 128, 32 | | 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 Jump Diffusions, LeakyReLU activation on the EM architecture led to diverging behavior, while tanh did not, we thus changed the activation to tanh for a more stable behaviour.

| Architecture | Modules: Hidden Layers | Layer Size | Activation | # of Parameters |
|--------------|----------------------------------|-------------|------------------|-----------------|
| MLP (Itô) | 10 | 64 | | 141858 |
| EM | $\varphi: 4, f: 4$ | 64, 64 | LeakyReLU | 133795 |
| IM | $\varphi: 4, f: 4, W_0: 1$ | 64, 64, 512 | | 134371 |
| ML | $\varphi: 4, f: 4, \hat{P}_t: 3$ | 64, 64, 32 | s: tanh, t: ReLU | 135960 |
| DeepAR-LSTM | | 64 | | 117894 |
| DeepAR-RNN | 3 | 130 | LeakyReLU | 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 | | 133126 |
| EM | $\varphi: 4, f: 4$ | 128, 123 | LeakyReLU | 134409 |
| IM | $\varphi: 4, f: 4, W_0: 1$ | 128, 128, 128 | | 134921 |
| ML | $\varphi: 4, f: 4, \hat{P}_t: 3$ | 128, 128, 128 | s: tanh, t: ReLU | 239724 |
| DeepAR-LSTM | | 64 | | 117123 |
| DeepAR-RNN | 3 | 130 | LeakyReLU | 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 10.

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

Table 7: 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.

683 C.4 Additional Figures and Tables

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

686 C.4.1 Ablation on IM Architecture Width

687 We conduct a series of ablations on the width of the IM architecture. These ablations are performed
 688 on the synthetic datasets of the Kuramoto model and the Fitzhugh-Nagumo model. The results are
 689 presented in Figure 7 and 8.

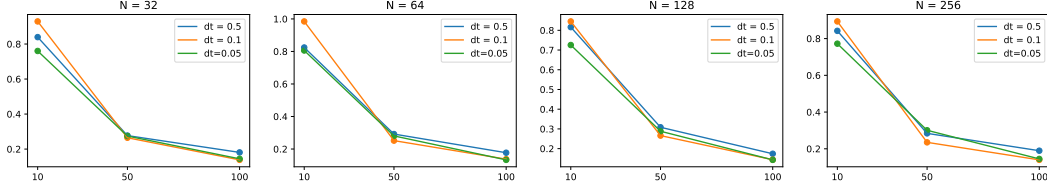


Figure 7: Ablation on different IM architecture widths $N = 32, 64, 128, 256$ for the Kuramoto model, with different time grid size dt and different number of training particles.

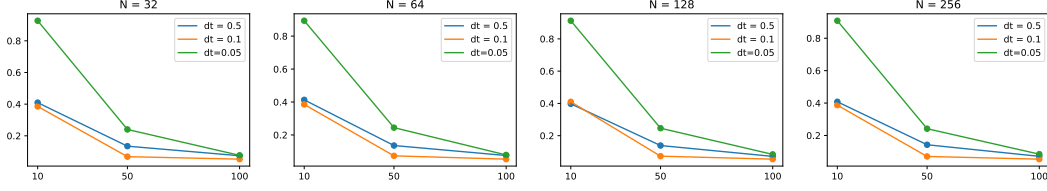


Figure 8: Ablation on different IM architecture widths $N = 32, 64, 128, 256$ for the Fitzhugh-Nagumo model, with different time grid size dt and different number of training particles.

690 **C.4.2 Synthetic Data Experiments**

Table 8: Synthetic dataset results with noise level standard deviation 0.1

| | Kuramoto | Fitzhugh | OD | MA | OU | Circle |
|-----------|---------------|---------------|---------------|---------------|---------------|---------------|
| MLP (Itô) | 0.56 (0.081) | 0.699 (0.426) | 0.048 (0.013) | 2.14 (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 (0.01) | 1.359 (0.2) | 0.097 (0.038) | 1.562 (2.476) |

Table 9: 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) | 0.1 (0.042) | 1.334 (1.948) |
| IM | 0.45 (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) | 0.1 (0.035) | 1.564 (2.543) |
| EM | 0.373 (0.075) | 0.612 (0.383) | 0.038 (0.01) | 1.347 (0.165) | 0.106 (0.036) | 1.535 (2.535) |

Table 10: 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 (0.2) | 0.481 (0.065) | 2.303 (2.039) |
| IM | 0.646 (0.065) | 0.878 (0.522) | 0.055 (0.012) | 1.65 (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) | 0.04 (0.009) | 1.652 (0.272) | 0.536 (0.055) | 2.394 (2.25) |

691 For a better sense of the different synthetic datasets and each model’s ability in recovering the drift,
692 we provide a figure that qualitatively compares the architectures’ performances in Figure 9. We
693 additionally show the learnt gradient flow for the Kuramoto model in Figure 10.

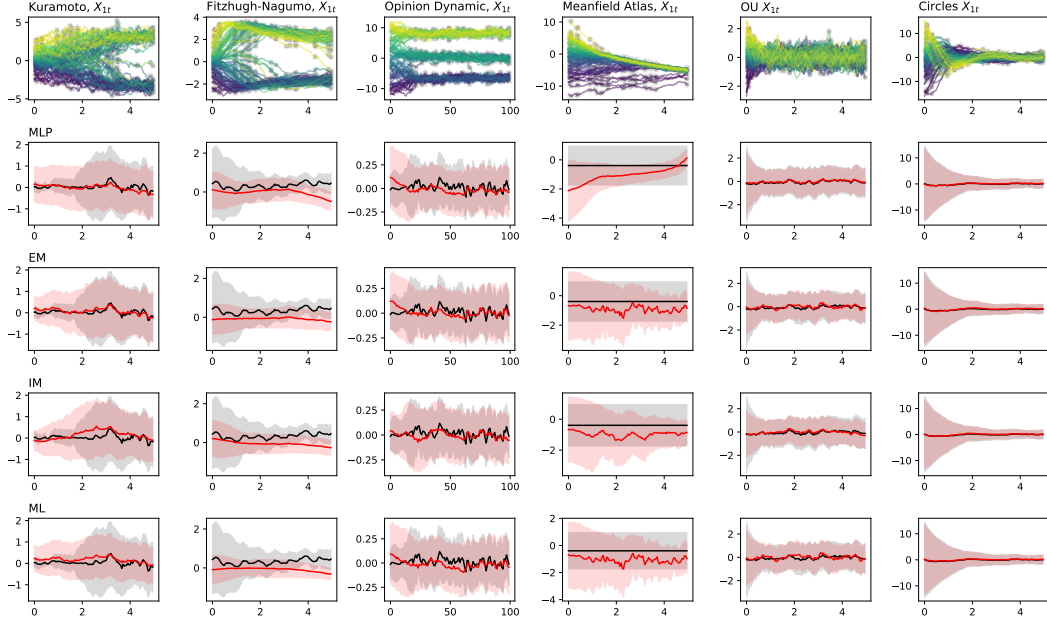


Figure 9: Synthetic data experiments and estimated drifts, only the first dimension is shown. First row: sampled trajectories, grey scattered circles indicate irregular time observations. Rows 2-5: estimated drifts by the MLP (Itô), EM, IM, ML architectures. Black is truth, red is estimated. The models are trained with additional Gaussian observation noise of $SD = 0.1$.

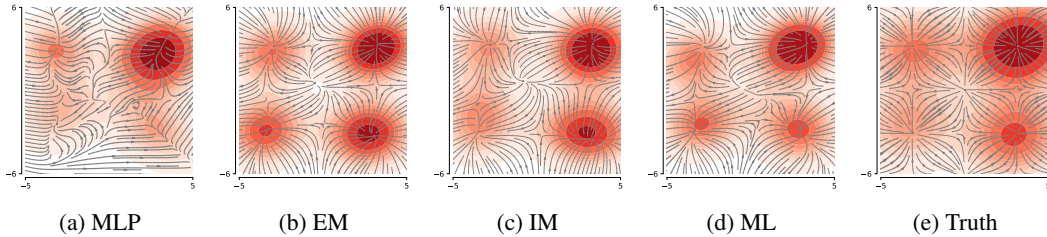


Figure 10: Estimated gradient flow of Kuramoto Model at terminal time. The colors correspond to the density of generated samples at terminal time. The models are trained with additional Gaussian observation noise of $SD = 0.1$.

694 C.4.3 Real Data Experiments

695 We extend the time series experiments in the main paper to forecasting. There are two ways to
696 perform forecasting: i) given the initial condition at T_0 , generate trajectories up to T_{forecast} ; ii) given
697 the training terminal condition at T , generate trajectories for $t \in [T, T_{\text{forecast}}]$. In both cases, the
698 dataset time steps are partitioned into 0.8 training, 0.2 forecasting. We present the numerical results
699 of both types of forecasting in Tables 11 and 12. We note that our methods perform on par with
700 various deepAR methods under both types of forecasting. We also present qualitative results with the
701 learnt drifts in Figures 11 and 12 for the EEG and Chemotaxis data.

Table 11: Time series forecasting Type I. NA/A stands for Non-alcoholics/ Alcoholics.
Bolded values indicate best performance.

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

Table 12: 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) |

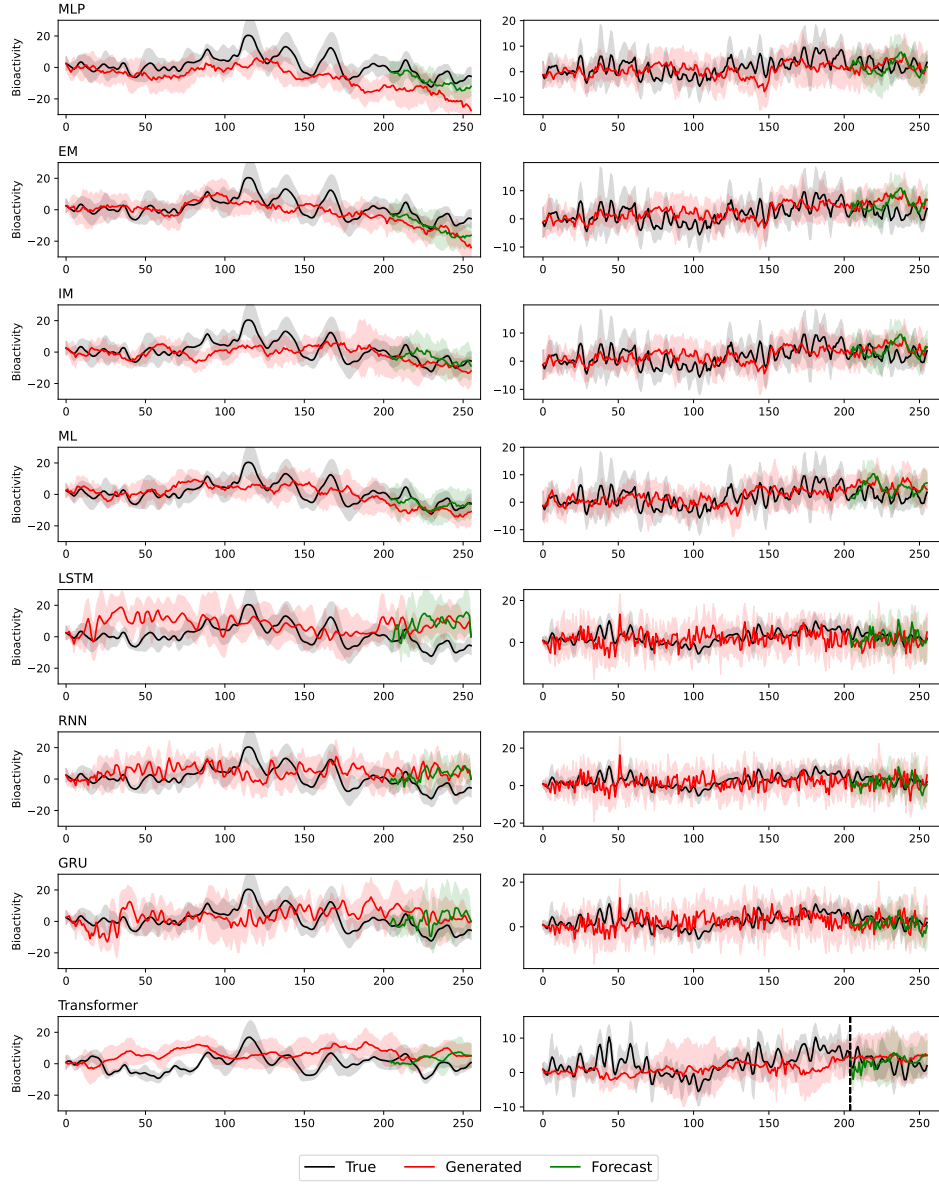


Figure 11: True, generated and forecast trajectories on EEG dataset. Left:Non-Alcoholics; Right:Alcoholics. The dashed vertical line at $t = 205$ indicates the start of the forecast.The shaded region indicates \pm one SD of samples at each time step.

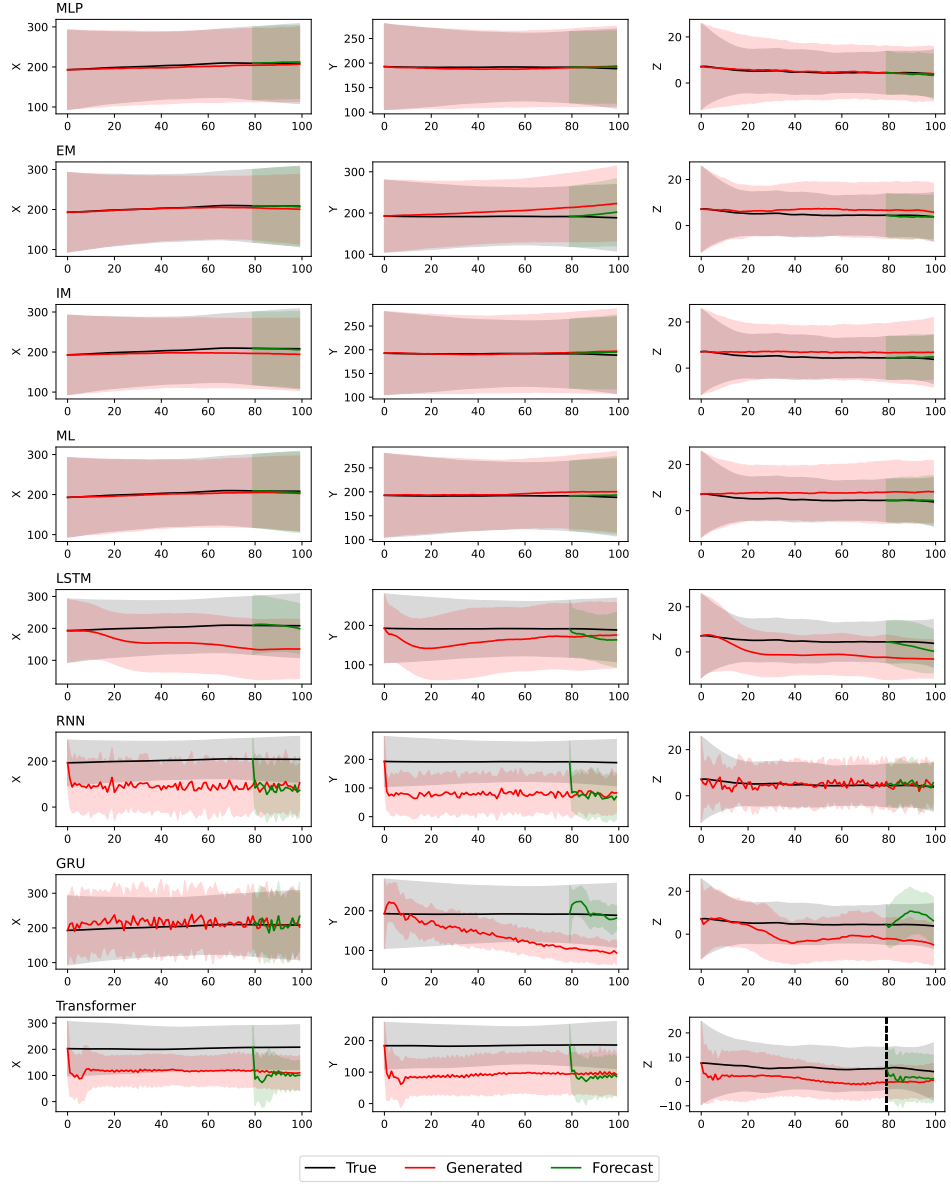


Figure 12: True, generated and forecast trajectories on chemotaxis, *C. Crescentus* dataset. The dashed vertical line at $t = 80$ indicates the start of the forecast. The shaded region indicates \pm one SD of samples at each time step. From left to right, the columns are movements in x, y and z directions.

702 C.4.4 Generative Modeling Experiments

703 Figure 13 shows 5 randomly selected 2-d projections of the 100-d mixture of Gaussians.

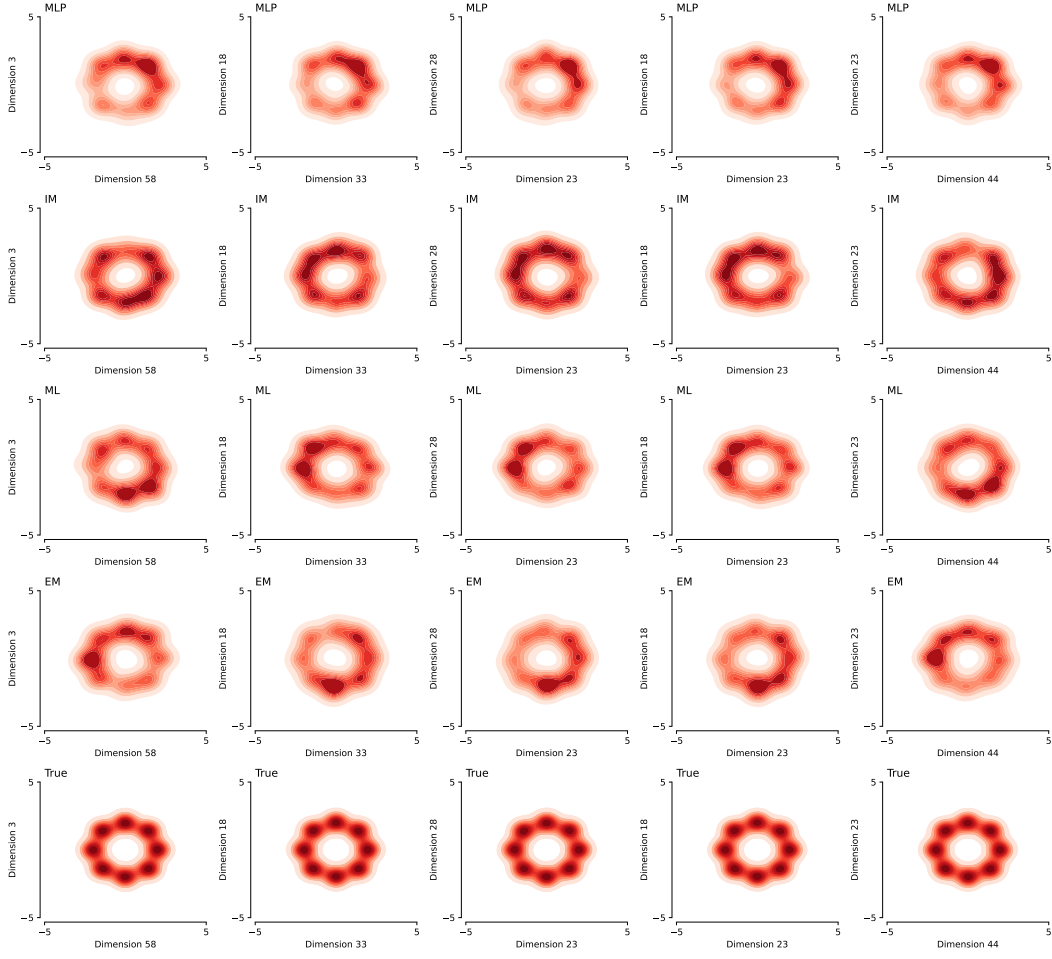
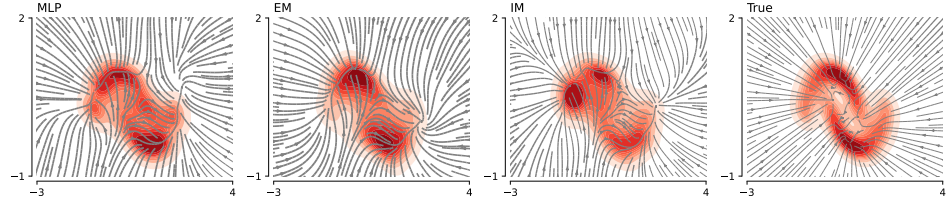
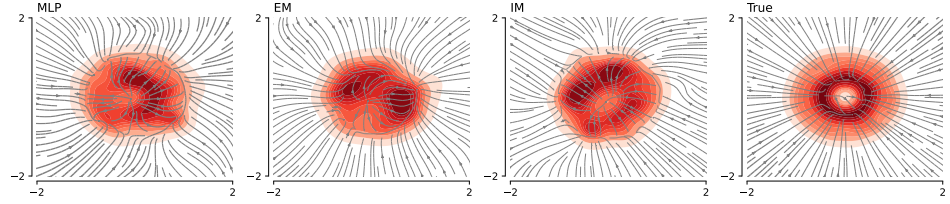


Figure 13: Five randomly selected 2-d projections of 100-d mixture of Gaussians.

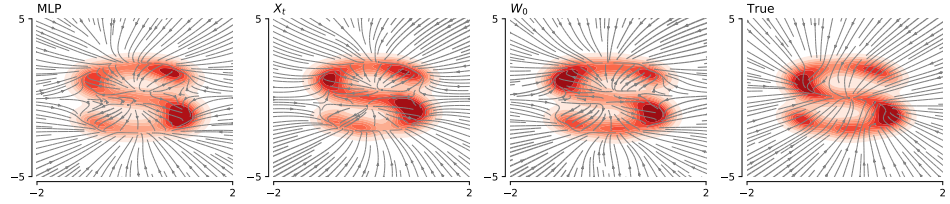
704 In addition to the eight Gaussian mixture and real data presented in the main paper, we present a few
 705 toy generative modeling experiments to better understand the different architectures in Figure 14.



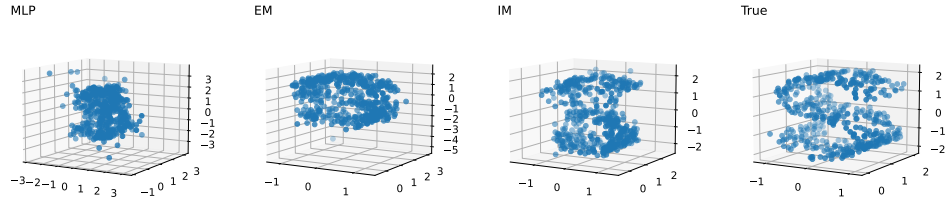
(a) Two Moons



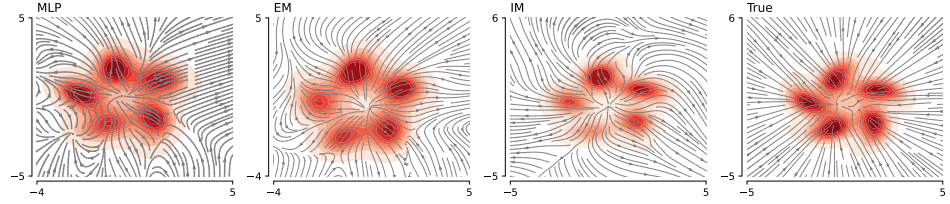
(b) Two Circles



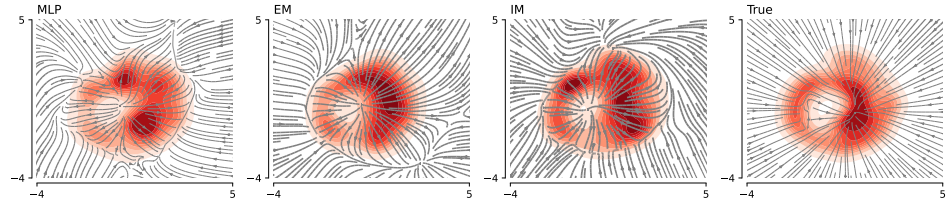
(c) S Curve, 2-d



(d) S Curve, 3-d



(e) Pinwheel



(f) Swissroll

Figure 14: 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.

706 C.4.5 Additional Experiments: Linear Fokker–Planck

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

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

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

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 $dX_t = b(\cdot)dt + \sigma dW_t$ such that

$$p_T(x) = \mathbb{E} \left[\exp \left(\int_0^T -\text{div} b(\cdot) dt \right) p_0(X_T) \mid X_0 = x \right].$$

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 $dX_t = \sigma dW_t$ with no drift and

$$p_T(x) = \mathbb{E} \left[\exp \left(\int_0^T -\text{div} b(\cdot) dt \right) p_0(X_T) \exp \left(\int_0^T b(\cdot) dX_t - \frac{1}{2} \int_0^T b^2(\cdot) dt \right) \mid X_0 = x \right].$$

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

$$\log p_T(x) \geq \mathbb{E} \left[\int_0^T -\text{div} b(\cdot) dt + \log(p_0(X_T)) + \int_0^T b(\cdot) dX_t - \frac{1}{2} \int_0^T b^2(\cdot) dt \mid X_0 = x \right].$$

715 The integrals are approximated using the forward Euler method and the parameters of b are opti-
 716 mized for the set of observations. The results are given in Table 13. The results suggest that the
 717 proposed architectures do not decrease performance in the linear setting and sometimes provide slight
 718 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) | <i>72.045</i> (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) | <i>71.318</i> (0.985) | 69.000 0.550 |
| EM | 37.793 (0.307) | <i>33.319</i> (0.264) | <i>54.089</i> (0.607) | 72.636 (0.600) | 70.692 (3.596) | 69.230 0.546 |

Table 13: 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.