SDE MATCHING: SCALABLE AND SIMULATION-FREE TRAINING OF LATENT STOCHASTIC DIFFERENTIAL EQUATIONS

Grigory Bartosh[1,†] **Dmitry Vetrov**[2] **Christian A. Naesseth**[1]
[1]University of Amsterdam [2]Constructor University, Bremen

ABSTRACT

The Latent Stochastic Differential Equation (SDE) is a powerful tool for time series and sequence modeling. However, training Latent SDEs typically relies on adjoint sensitivity methods, which depend on simulation and backpropagation through approximate SDE solutions, which limit scalability. In this work, we propose SDE Matching, a new simulation-free method for training Latent SDEs. Inspired by modern Score- and Flow Matching algorithms for learning generative dynamics, we extend these ideas to the domain of stochastic dynamics for time series and sequence modeling, eliminating the need for costly numerical simulations. Our results demonstrate that SDE Matching achieves performance comparable to adjoint sensitivity methods while drastically reducing computational complexity.

1 INTRODUCTION

Differential equations are a natural choice for modeling continuous-time dynamical systems and have recently received significant interest in machine learning. Since Chen et al. (2018) proposed the adjoint sensitivity method for learning Ordinary Differential Equations (ODEs) in a memory-efficient manner, ODE-based approaches became popular in deep learning for density estimation Grathwohl et al. (2019) and to model unevenly observed time series Yildiz et al. (2019); Rubanova et al. (2019)].

However, ODEs describe deterministic systems and encode all uncertainty into their initial conditions. This limits the applicability of ODE-based approaches when modeling stochastic and chaotic processes. To address this limitation, Li et al. (2020) extended the adjoint sensitivity method to Latent Stochastic Differential Equations (SDEs).

Table 1: Asymptotic complexity comparison. L and R are number of sequential and parallel evaluations of drift/diffusion terms, respectively, and D is the number of parameters/states.

Method	**Memory**	**Time**
Forward Pathwise (Yang & Kushner, 1991) (Gobet & Munos, 2005)	$\mathcal{O}(1)$	$\mathcal{O}(LD)$
Backprop through Solver (Giles & Glasserman, 2006)	$\mathcal{O}(L)$	$\mathcal{O}(L)$
Stochastic Adjoint (Li et al., 2020)	$\mathcal{O}(1)$	$\mathcal{O}(L \log L)$
Amortized Reparameterization (Course & Nair, 2023)	$\mathcal{O}(R)$	$\mathcal{O}(R)$
SDE Matching (this paper)	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Despite these advancements, training of ODE and SDE models is *simulation-based*, it relies on costly numerical integration of differential equations and backpropagation through the solutions. With training algorithms that are difficult to parallelize on modern hardware, Latent SDEs have resisted truly scaling.

[†]Corresponding to: g.bartosh@uva.nl

In parallel, Score Matching (Ho et al., 2020; Song et al., 2021c) and Flow Matching methods (Lipman et al., 2023; Albergo et al., 2023; Liu et al., 2023) have demonstrated that continuous-time dynamics for generative modeling can be learned efficiently in a simulation-free manner—without requiring numerical integration. These techniques have proven computationally efficient and scalable for high-dimensional problems. Inspired by these developments, we develop a simulation-free approach for learning SDEs.

We introduce SDE Matching—a simulation-free framework for learning Latent SDE models. The key idea we adopt from matching-based approaches is direct access to latent *posterior* samples at any time step. This eliminates the need to integrate SDEs during training. The SDE Matching objective is estimated using the Monte Carlo method, achieving $\mathcal{O}(1)$ memory and time complexity. We summarize the complexity comparisons in Table 1.

We summarize our contributions as follows:

1. We propose an efficient parameterization of the Latent SDE posterior process and use it to develop SDE Matching, a simulation-free training procedure for Latent and Neural SDEs.
2. We demonstrate that SDE Matching achieves comparable performance to the adjoint sensitivity method on both synthetic and real data, while significantly reducing the computational cost of training and improving convergence.

2 Background

Consider a series of observations $\mathbf{X} = x_{t_1}, x_{t_2}, \ldots, x_{t_N}$, where each $x_{t_i} \in \mathrm{X}$ where the space X depends on the application. For simplicity of notation, we assume that all time steps t_i belong to the interval $[0, 1]$. The extension to intervals of arbitrary lengths is straightforward.

The Latent SDE model, also known as Neural SDE, assumes that this series is generated constructively by the following stochastic process, referred to as the prior process. First, sample a latent variable $z_0 \in \mathbb{R}^D$ from the initial prior distribution $p_\theta(z_0)$. Next, infer the latent continuous dynamics $\mathbf{Z} = (z(t))_{t\in[0,1]}$ by integrating the following SDE:

$$dz_t = h_\theta(z_t, t)dt + g_\theta(z_t, t)d\mathbf{w}, \tag{1}$$

where $h_\theta : \mathbb{R}^D \times [0, 1] \mapsto \mathbb{R}^D$ is the drift term, $g_\theta(z_t, t) : \mathbb{R}^D \times [0, 1] \mapsto \mathbb{R}^{D\times D}$ is the diffusion term, and $\mathbf{w}$ is a standard Wiener process. The SDE in Equation (1), together with the prior distribution $p_\theta(z_0)$, defines a sequence of probability distributions $p_\theta(z_t)$ at each time step $t \in [0, 1]$. Once the trajectory of the latent variables $\mathbf{Z}$ is sampled, we independently sample observations x_{t_i} from the conditional distributions $p_\theta(x_{t_i}|z_{t_i})$ for each time step t_i.

The goal of the Latent SDE is to find a set of parameters θ that best fits a dataset of observed time series or sequence. Unfortunately, the posterior distribution of the latent variables, $p_\theta(\mathbf{Z}|\mathbf{X})$, is generally intractable. However, variational inference can be used to train the Latent SDE. Similar to the prior process, we introduce an approximate posterior process, which is conditioned on the observations $\mathbf{X}$. The posterior process consists of two components: the initial posterior distribution $q_\varphi(z_0|\mathbf{X})$ and a conditional SDE:

$$dz_t = f_\varphi(z_t, t, \mathbf{X})dt + g_\theta(z_t, t)d\mathbf{w}, \tag{2}$$

where $f_\varphi(z_t, t, \mathbf{X})$ defines the drift term of the SDE conditionally on the observations $\mathbf{X}$. It is important to note that, despite having a different drift term, the posterior shares the same diffusion term $g_\theta(z_t, t)$ as in Equation (1).

The training objective of the Latent SDE is a variational bound on the log-marginal likelihood of the observations:

$$-\log p_\theta(\mathbf{X}) \le \mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{rec}}, \quad \mathcal{L}_{\text{prior}} = \mathrm{D}_{\mathrm{KL}}\big(q_\varphi(z_0|\mathbf{X}) \| p_\theta(z_0)\big) \tag{3}$$

$$\mathcal{L}_{\text{diff}} = \mathop{\mathbb{E}}_{q_\varphi(\mathbf{Z}|\mathbf{X})}\left[\int_0^1 \frac{1}{2}\left\|r_{\theta,\varphi}(z_t, t, \mathbf{X})\right\|_2^2 dt\right], \quad \mathcal{L}_{\text{rec}} = \mathop{\mathbb{E}}_{q_\varphi(\mathbf{Z}|\mathbf{X})}\left[\sum_{i=1}^N -\log p_\theta(x_{t_i}|z_{t_i})\right], \tag{4}$$

where $r_{\theta,\varphi}(z_t, t, \mathbf{X})$ satisfies

$$g_\theta(z_t, t) r_{\theta,\varphi}(z_t, t, \mathbf{X}) = h_\theta(z_t, t) - f_\varphi(z_t, t, \mathbf{X}). \tag{5}$$

Obtaining an estimate of the objective $\mathcal{L}$ for stochastic optimization requires joint numerical integration of the posterior process SDE (Equation (2)) and the integral in Equation (4). While there exist methods to improve the memory and computational efficiency of numerical integration, most are simulation-based and require backpropagation through numerical solutions of SDEs. Besides being computationally expensive, these methods are difficult to parallelize and suffer from numerical instabilities. Altogether, these challenges make training the Latent SDE a difficult task.

3 SDE MATCHING

Diffusion models can be trained in a simulation-free manner because they do not require full simulation of the noising process. Instead, they allow direct sampling of latent variables z_t from the marginal distribution $q(z_t|x)$. In contrast, Latent SDEs are often trained using a posterior process parameterized by a general-form SDE. This renders the marginal posterior distribution $q_\varphi(z_t|\mathbf{X})$ intractable, which prevents simulation-free training.

To address this limitation, we propose SDE Matching – a framework for simulation-free training of Latent SDE models. The key idea behind SDE Matching is to parameterize the posterior process' conditional SDE via a learnable function $F_\varphi(\varepsilon, t, \mathbf{X})$ that directly defines the marginal distributions $q_\varphi(z_t|\mathbf{X})$. This design inherently allows direct sampling of latent variables without numerical integration. Importantly, SDE Matching simplifies only the training procedure while leaving the generative dynamics of the Latent SDE prior process fully flexible.

3.1 GENERATIVE MODEL

In SDE Matching, the prior process is identical to the standard Latent SDE model (see Section 2):

$$dz_t = h_\theta(z_t, t)dt + g_\theta(z_t, t)d\mathbf{w}. \tag{6}$$

In general, the functions h_θ and g_θ may be parameterized using neural networks of arbitrary form. However, the choice of parameterization involves some trade-offs, which we will discuss later.

Observations x_{t_i} are likewise assumed to be sampled conditionally independently from the likelihood distributions $x_{t_i} \sim p_\theta(x_{t_i}|z_{t_i})$ for each time step t_i.

3.2 POSTERIOR PROCESS

Instead of defining the posterior process through a conditional SDE (Equation (2)) and then deriving its analytical solution, we propose an alternative approach. We first define the conditional marginal distribution of latent variables $q_\varphi(z_t|\mathbf{X})$ for each t, and then derive the corresponding conditional SDE with the desired marginal distributions.

We construct the posterior process by following three steps:

1. Define the marginal distribution implicitly through a parameterized function of noise;
2. Construct a conditional ODE that satisfies the target marginals;
3. Transition from the ODE to a SDE while preserving the marginal distributions.

A detailed description of each step with proofs is provided in Appendix A.

Posterior Marginal Distribution. We define the conditional marginal distribution $q_\varphi(z_t|\mathbf{X})$ implicitly. First, we introduce a function that, given time t and observations $\mathbf{X}$, transforms noise ε into a latent variable z_t:

$$z_t = F_\varphi(\varepsilon, t, \mathbf{X}), \tag{7}$$

where $\varepsilon \sim q(\varepsilon) = \mathcal{N}(\varepsilon; 0, I)$. This implicitly defines the conditional distribution of latent variables $q_\varphi(z_t|\mathbf{X})$. Although, in general, we do not have an explicit form for this distribution, the above definition inherently enables efficient sampling. The specific parameterization of F_φ and the dependence of z_t on observations $\mathbf{X}$ are user design choices.

Conditional ODE. We assume that F_φ is differentiable with respect to ε and t, and invertible with respect to ε. Fixing observations $\mathbf{X}$ and noise ε, and varying t from 0 to 1, results in a smooth trajectory from z_0 to z_1. Differentiating these trajectories over time yields a velocity field that generates the conditional distribution $q_\varphi(z_t|\mathbf{X})$, the conditional ODE:

$$dz_t = \bar{f}_\varphi(z_t, t, \mathbf{X})dt, \quad \text{where} \qquad \bar{f}_\varphi(z_t, t, \mathbf{X}) = \left.\frac{\partial F_\varphi(\varepsilon, t, \mathbf{X})}{\partial t}\right|_{\varepsilon = F_\varphi^{-1}(z_t, t, \mathbf{X})}. \tag{8}$$

Thus, if we sample $z_0 \sim q_\varphi(z_0|\mathbf{X})$ and solve Equation (8) up to time t, we obtain a sample $z_t \sim q_\varphi(z_t|\mathbf{X})$.

The time derivative of F_φ can be computed efficiently using forward-mode differentiation in modern frameworks such as PyTorch Paszke et al. (2017) or JAX Bradbury et al. (2018).

Conditional SDE. Finally, we derive the conditional SDE that defines the posterior process.

Given access to both the conditional ODE and the score function $\nabla_{z_t} \log q_\varphi(z_t|\mathbf{X})$, we follow Song et al. (2021c) and define a conditional SDE with marginal distributions $q_\varphi(z_t|\mathbf{X})$ as follows:

$$dz_t = f_{\theta,\varphi}(z_t, t, \mathbf{X})dt + g_\theta(z_t, t)d\mathbf{w}, \quad \text{where} \tag{9}$$

$$f_{\theta,\varphi}(z_t, t, \mathbf{X}) = \bar{f}_\varphi(z_t, t, \mathbf{X}) + \frac{1}{2} g_\theta(z_t, t) g_\theta^\top(z_t, t) \nabla_{z_t} \log q_\varphi(z_t|\mathbf{X}) + \frac{1}{2} \nabla_{z_t} \cdot \left[g_\theta(z_t, t) g_\theta^\top(z_t, t) \right]. \tag{10}$$

Here, the diffusion term g_θ is the matrix-valued function from the prior process. It is crucial that the posterior process has the same diffusion term as the prior process to ensure that the variational bound is finite. Notably, g_θ affects only the distribution of trajectories $z(t)_{t\in[0,1]}$, while the marginal distributions $q_\varphi(z_t|\mathbf{X})$ do not depend on g_θ.

Evaluating the drift term in Equation (9) presents several challenges. First, it requires access to the conditional score function $\nabla_{z_t} \log q_\varphi(z_t|\mathbf{X})$, which can be computationally expensive. However, for F_φ functions that enable efficient log-determinant computation of the Jacobian matrix, the score function can be computed efficiently (e.g., functions linear in ε or RealNVP architectures Dinh et al. (2017); Kingma & Dhariwal (2018)). The calculation of this score function is further discussed in Appendix A.4.

The second challenge involves computing the last term in Equation (10), which can be expensive in high-dimensional settings. However, it can be computed efficiently if for example the diffusion term g_θ is of the form $\sigma_\theta(z_t, t)\Gamma_\theta(t)$ for scalar-valued σ_θ and matrix-valued Γ_θ. Another option is if g_θ is a diagonal matrix, where each element $g_\theta(z_t, t)_{k,k}$ depends only on the k-th coordinate of z_t. Furthermore, if g_θ does not depend on the state z_t, this term vanishes. Notably, while this structure limits flexibility, it is identical to constraints in memory-efficient implementations of standard Latent SDE training Li et al. (2020), which means that SDE Matching does not introduce additional restrictions.

3.3 OPTIMIZATION

Like standard Latent SDE training, SDE Matching optimizes the parameters θ and φ end-to-end by optimizing the same variational objective. However, the training procedure of SDE Matching can be made simulation-free by leveraging the above posterior process and rewriting the objective from Equation (3) as follows:

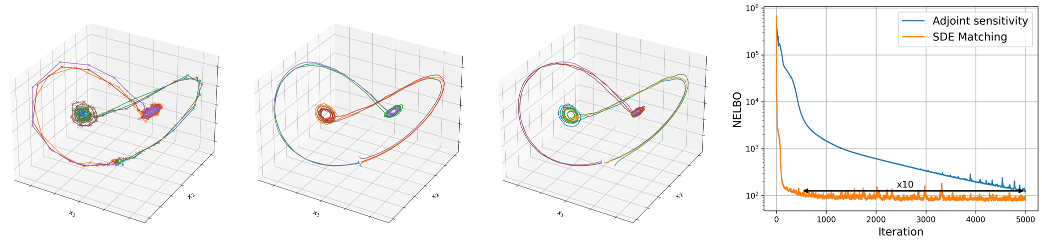
$$-\log p_\theta(\mathbf{X}) \le \mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{rec}}, \quad \mathcal{L}_{\text{prior}} = \mathrm{D}_{\mathrm{KL}}\big(q_\varphi(z_0|\mathbf{X}) \| p_\theta(z_0)\big) \tag{11}$$

$$\mathcal{L}_{\text{diff}} = \mathop{\mathbb{E}}_{u(t) q_\varphi(z_t|\mathbf{X})} \left[\frac{1}{2} \left\| r_{\theta,\varphi}(z_t, t, \mathbf{X}) \right\|_2^2 \right], \quad \mathcal{L}_{\text{rec}} = N \mathop{\mathbb{E}}_{u(i) q_\varphi(z_{t_i}|\mathbf{X})} \left[-\log p_\theta(x_{t_i}|z_{t_i}) \right], \tag{12}$$

where $u(t)$ represents a uniform distribution over the interval $[0, 1]$, $u(i)$ represents a uniform distribution over the set $1, \ldots, N$, and $r_{\theta,\varphi}(z_t, t, \mathbf{X})$ satisfies:

$$g_\theta(z_t, t) r_{\theta,\varphi}(z_t, t, \mathbf{X}) = h_\theta(z_t, t) - f_{\theta,\varphi}(z_t, t, \mathbf{X}). \tag{13}$$

The objective in Equation (11) consists of three terms: the prior loss $\mathcal{L}_{\text{prior}}$, the diffusion loss $\mathcal{L}_{\text{diff}}$, and the reconstruction loss $\mathcal{L}_{\text{rec}}$. The optimization of $\mathcal{L}_{\text{prior}}$ with respect to parameters θ can be omitted if we are not interested in unconditional sampling or in learning the initial prior $p_\theta(z_0)$—for



(a) Training data (b) Adjoint sensitivity (c) SDE Matching (d) Convergence

Figure 1: (a) Training data distribution and learned dynamics from a 3D stochastic Lorenz attractor. Results for Latent SDE trained with (b) adjoint sensitivity method and (c) SDE Matching. (d) Negative Evidence Lower Bound (NELBO) (Equation (11)) objective with respect to iteration step on 3D stochastic Lorenz attractor data. SDE Matching demonstrates approximatly 10 times faster convergence in terms of iterations.

example, if we are solely interested in forecasting. Otherwise, this term can still be optimized efficiently.

The other two terms, $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{rec}}$, which previously required the numerical integration of the conditional SDE, can now be estimated more efficiently. Specifically, for each term, we can first sample a timestep t, then sample $z_t \sim q_\varphi(z_t|\mathbf{X})$, and finally evaluate the corresponding function. Importantly, SDE Matching allows the estimation of $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{rec}}$ with an arbitrary number of samples evaluated in parallel. It also enables inference of the posterior process (Equation (9)) as a special case. However, we found that using a single sample was sufficient for our experiments. We summarize the training procedure in Algorithm 1.

Note that while we use the functions $\bar{f}_\varphi$ (Equation (8)) and $f_{\theta,\varphi}$ (Equation (9)) to describe the conditional SDE in the posterior process, these functions are ultimately defined by the reparameterization function F_φ (Equation (7)) and g_θ, which are the functions we actually parameterize.

3.4 Sampling

Unconditional sampling at inference- or test-time for SDE Matching follows the same procedure as sampling from a Latent SDE. First, we sample an initial latent variable $z_0 \sim p_\theta(z_0)$ and then integrate the unconditional SDE in Equation (6) using any off-the-shelf SDE solver. Then, at each desired time step t, we project the latent states z_t to the data space by sampling x_t from the distribution $p_\theta(x_t|z_t)$.

However, if we have access to partial observations $\mathbf{X} = x_{t_1}, x_{t_2}, \dots, x_{t_N}$ and are interested in forecasting for $t > t_N$, we can instead first sample the latent state z_{t_N} from $q_\varphi(z_{t_N}|\mathbf{X})$ and then integrate the prior process dynamics using this sample as initial condition. This follows because the Latent SDE is Markovian with observations that are conditionally independent given the latent state. Importantly, because SDE Matching provides direct access to an approximation to the posterior (smoothing) marginals it can be used for simulation-free sampling (inference) of the latent state, whereas conventional parameterization of the posterior process would require simulating the conditional SDE up to time step t_N first.

Similarly, interpolation can be performed by inferring only the posterior process dynamics. Alternatively, for more general conditioning events and steering one could leverage Sequential Monte Carlo methods (Naesseth et al., 2019; Chopin et al., 2020; Wu et al., 2023) for conditional sampling, but a detailed investigation of these approaches is beyond the scope of this work.

Due to the strong connection between SDE Matching and diffusion models—and given that diffusion models have demonstrated exceptional performance in deterministic sampling—it is natural to ask whether it is possible to sample deterministically from SDE Matching. Indeed, the prior process and the initial distribution $p_\theta(z_0)$ together define a sequence of marginal probability distributions $p_\theta(z_t)$. While it is possible to learn an ODE corresponding to $p_\theta(z_t)$, it is important to clarify that solving this ODE does not necessarily produce meaningful time-series samples.

4 EXPERIMENTS

SDE Matching is a simulation-free framework for training Latent SDE models. Therefore, the goal of this section is to demonstrate that SDE Matching indeed enables computationally efficient training of Latent SDEs. Our objective is not to outperform all other Latent SDE-based approaches for time series modeling. Instead, SDE Matching is compatible with existing extensions and can be combined with them for further improvements.

Table 2: Test MSE on motion capture dataset. We report an average performance based on 10 models trained with deferent random seeds and 95% confidence interval based on t-statistic. The first section of the table contains simulation-based approaches and the second part contains simulation-free approaches. †results from Yildiz et al. (2019), *from Li et al. (2020) and ‡from Course & Nair (2023).

Model	Test MSE ↓
DTSBN-S Gan et al. (2015)	$34.86 \pm 0.02^{\dagger}$
npODE Heinonen et al. (2018)	$22.96^{\dagger}$
NeuralODE Chen et al. (2018)	$22.49 \pm 0.88^{\dagger}$
ODE^{2}VAE Yildiz et al. (2019)	$10.06 \pm 1.4^{\dagger}$
ODE^{2}VAE-KL Yildiz et al. (2019)	$8.09 \pm 1.95^{\dagger}$
Latent ODE Rubanova et al. (2019)	$5.98 \pm 0.28^{*}$
Latent SDE Li et al. (2020)	$\mathbf{4.03 \pm 0.2}^{*}$
ARCTA Course & Nair (2023)	$7.62 \pm 0.93^{\ddagger}$
SDE Matching (ours)	$\mathbf{4.50 \pm 0.32}$

To this end, we provide two sets of experiments for both synthetic data and real motion capture data. In all experiments, we use the same hyperparameters for both the conventional parameterization of Latent SDE and the parameterization described in Section 3. Additional discussions on parameterization and training are provided in Appendix C. When using the SDE Matching training procedure, we jointly optimize the parameters of the generative model θ in the prior $p_\theta(z_0)$, drift $h_\theta(z_t, t)$, diffusion $g_\theta(z_t, t)$, and observation model $p_\theta(x_t|z_t)$, as well as the parameters φ of the posterior reparameterization function F_φ. All parameters are optimized jointly by minimizing the variational bound in Equation (11). We do not apply annealing or additional regularization techniques during training.

The experiments demonstrate that SDE Matching achieves comparable accuracy to simulation-based Latent SDE training, while significantly reducing computational complexity. Not only does SDE Matching reduce the computational cost of each training iteration, the parameterization of the posterior process also leads to faster convergence in terms of the number of iterations for further gains compared to alternatives.

4.1 SYNTHETIC DATASETS

For the synthetic data experiment, we consider the 3D stochastic Lorenz attractor process from Li et al. (2020) with identical observation generation procedure.

As shown in Figure 1, both models—one trained using the adjoint sensitivity method and the other with SDE Matching—successfully learned the underlying dynamics. However, SDE Matching required only a single evaluation of the drift term in the posterior process for each iteration, whereas the adjoint sensitivity method required 100 simulation steps for this experiment. In terms of absolute runtime, a single training iteration with SDE Matching was approximately five times faster.

Moreover, as demonstrated in Figure 1d, SDE Matching leads to faster convergence of the model. We attribute this to the parameterization of the posterior process through the reparameterization function F_φ (Equation (7)). We hypothesize that since this function directly models the marginal distributions of the posterior process, the model can more efficiently learn compared to integrating conditional SDEs.

The combined, per-iteration and convergence, runtime speed-up is over $50\times$ compared to the adjoint sensitivity approach for this example.

4.2 MOTION CAPTURE DATASET

To validate SDE Matching on real-world data, we follow Li et al. (2020); Course & Nair (2023) and evaluate it on the motion capture dataset from Gan et al. (2015). This dataset consists of motion recordings from 35 walking subjects, represented as 50-dimensional time series with 300 observations each. The dataset is split into 16 training sequences, 3 validation sequences, and 4 test sequences, following the preprocessing method from Wang et al. (2007).

We use the same hyperparameters as Li et al. (2020), including setting the latent state dimensionality to 6 and keeping the number of training iterations unchanged. In Table 2, we report the average performance on the training set over 10 models trained from different random seeds. SDE Matching achieves similar performance to the adjoint sensitivity method while being significantly more computationally efficient.

SDE Matching also outperforms the other simulation-free approach, ARCTA Course & Nair (2023), which can be seen as a special case of SDE Matching. Notably, ARCTA requires drawing around 100 latent samples and evaluations of SDEs per batch, whereas SDE Matching requires only one. We attribute this result to the fact that ARCTA employs a less flexible posterior parameterization, where the distribution of latent states $q_\varphi(z_t|\mathbf{X})$ depends heavily on observations close to time t. This dependence makes it challenging to learn long-term dependencies and limits the ability to propagate learning signals effectively.

5 Related Work

Differential equations are a widely recognized technique for modeling continuous dynamics. Recently they have seen a surge of interest in machine learning after the introduction of Neural ODEs by Chen et al. (2018) and Rubanova et al. (2019). Neural ODEs demonstrated strong performance in time-series modeling compared to recurrent neural networks, particularly for irregularly sampled time series. However, Neural ODEs have several limitations.

First, they rely on adjoint sensitivity methods for training, which requires numerical integration of gradients and backpropagation through the solutions. Aside from being computationally expensive and difficult to parallelize on modern hardware, ODEs can sometimes suffer from numerical instabilities Lea et al. (2000). Second, ODE-based models inherently encode all uncertainty into the initial conditions as the dynamics is completely deterministic. This approach is inadequate for modeling fundamentally stochastic processes, especially when observations are sparse or uninformative.

Li et al. (2020) extended the adjoint sensitivity method from ODEs to SDEs. Unlike Neural ODEs, Latent (or Neural) SDEs are better suited for modeling inherently stochastic and chaotic processes.

Numerous studies have proposed modifications to objective functions, introduced regularized dynamics, and improved computational efficiency and numerical stability for both Neural ODEs Kelly et al. (2020); Finlay et al. (2020); Kidger et al. (2021a) and Latent SDEs Kidger et al. (2021b). However, most of these methods still rely on backpropagation through the numerical solutions of differential equations limiting their scalability.

Another line of work Archambeau et al. (2007a;b) has proposed less computationally demanding techniques for inferring the Latent SDEs's training objective. Closest to our work is Course & Nair (2023) that develops a method for training Latent SDEs in a simulation-free manner. This approach is restricted to posterior processes with Gaussian marginals and does not support state-dependent diffusion terms in the prior process. Additionally, this method assumes that the latent state in the posterior process depends only on a few temporally closest observations. As a result, for effective optimization it requires drawing approximately 100 latent samples, rather than 1, per batch during training. This method can be seen as a special case of SDE Matching with the corresponding restrictions on the parameterization of the prior process and posterior process.

More recently, Zhang et al. (2024) proposed a simulation-free technique for time-series modeling. However, this approach learns dynamics directly in the original data space rather than in a latent space. Furthermore, to model non-Markovian processes the authors condition the generative dynamics on a fixed set of past observations, making simulations more computationally expensive and limiting the model's ability to capture long-range dependencies. This can be viewed as a special case of SDE Matching, where the latent space is constructed by concatenation of the most recent observations.

Another line of research that explores learning stochastic dynamics in a simulation-free manner is diffusion models Ho et al. (2020); Song et al. (2021c). As we demonstrated in Appendix B, conventional diffusion models can be seen as a special case of Latent SDEs with only a single observation. The key property that enables efficient training in conventional diffusion models is their simple and fixed noising process, i.e., the posterior process.

Recently Singhal et al. (2023); Bartosh et al. (2023); Nielsen et al. (2024); Sahoo et al. (2024); Bartosh et al. (2024); Du et al. (2024) have shown that the noising process in diffusion models can be more complex, and even learnable, while still preserving the simulation-free property. These approaches can also be viewed as special cases of SDE Matching by the re-interpretation of diffusion models as Latent SDEs.

6 Limitations and Future Work

The simulation-free properties of SDE Matching come with certain trade-offs. First, SDE Matching parameterizes the posterior process through the function $F_\varphi(\varepsilon, t, \mathbf{X})$ (Equation (7)). This function must not only be smooth and invertible with respect to ε, it must also provide access to the conditional score function $\nabla_{z_t} \log q_\varphi(z_t|\mathbf{X})$ (Equation (10)). These requirements restrict the flexibility of F_φ and, consequently, the posterior distribution approximation. Nevertheless, to the best of our knowledge, SDE Matching offers the most flexible parameterization that enables simulation-free access to latent samples $z_t \sim q_\varphi(z_t|\mathbf{X})$ of the conditional SDE posterior process.

Another limitation is the computational cost of evaluating the posterior process' SDE drift term for high-dimensional $g_\theta(z_t, t)$ (Equation (6)) of general form, as discussed in Section 3.2. However, it is worth reiterating that other memory-efficient training of Latent SDEs faces this same limitation.

SDE Matching does not constrain the flexibility of the prior process, which defines the generative model. Nevertheless, developing methods that allow even more flexible parameterization of F_φ and diffusion terms g_θ remains a promising direction for future research. Additionally, understanding the impact of this flexibility in modeling latent dynamics versus dynamics in the original data space is an interesting open question.

We believe that thanks to its simulation-free properties, SDE Matching has the potential to scale Latent SDE-based modeling to high-dimensional applications like audio and video generation and many other applications where they were previously infeasible. It could potentially also contribute to the development of simulation-free methods for training Latent ODE models and learning policies in reinforcement learning. However, we leave these directions for future work.

7 Conclusion

We introduced SDE Matching, a simulation-free framework for training Latent SDEs. By leveraging insights from score-based generative models, we formulated a method that eliminates the need for costly numerical simulations while maintaining the expressiveness of Latent SDEs. Our approach directly parameterizes the marginal posterior distributions, enabling efficient training and conditional inference.

Through both synthetic and real-world experiments, we demonstrated that SDE Matching achieves performance comparable to existing adjoint sensitivity-based methods while significantly reducing computational complexity. The ability to estimate latent trajectories without solving high-dimensional SDEs makes our method particularly suitable for large-scale time-series and sequence modeling.

Despite its advantages, SDE Matching introduces trade-offs in terms of posterior parameterization flexibility. Nevertheless, to the best of our knowledge, SDE Matching proposes the most flexible parameterization of the posterior process that allows simulation-free sampling of latent variables. Exploring more expressive function classes for posterior distributions and further extending SDE Matching are promising directions for future work.

Furthermore, our approach provides a foundation for scaling Latent SDEs to previously infeasible sequential domains such as audio and video. We believe that the scalability and efficiency of SDE Matching will enable broader applications in scientific modeling, finance, and healthcare, where structured uncertainty modeling is critical. Future research may also explore extensions to simulation-free training of Latent ODEs and connections to reinforcement learning and stochastic optimal control.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL https://arxiv.org/abs/2303.08797.

Cedric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pp. 1–16. PMLR, 2007a.

Cédric Archambeau, Manfred Opper, Yuan Shen, Dan Cornford, and John Shawe-Taylor. Variational inference for diffusion processes. *Advances in neural information processing systems*, 20, 2007b.

Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural diffusion models. *arXiv preprint arXiv:2310.08337*, 2023.

Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *arXiv preprint arXiv:2404.12940*, 2024.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. In *Advances in neural information processing systems*, volume 34, pp. 21325–21337, 2021.

Vladimir Igorevich Bogachev. *Measure theory*. Springer, 2007.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. 2018. URL http://github.com/google/jax.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

Kevin Course and Prasanth Nair. Amortized reparametrization: efficient and scalable variational inference for latent sdes. *Advances in Neural Information Processing Systems*, 36, 2023.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noé, Carla P Gomes, Alán Aspuru-Guzik, and Kirill Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. *arXiv preprint arXiv:2410.07974*, 2024.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pp. 3154–3164. PMLR, 2020.

Zhe Gan, Chunyuan Li, Ricardo Henao, David E Carlson, and Lawrence Carin. Deep temporal sigmoid belief networks for sequence modeling. *Advances in Neural Information Processing Systems*, 28, 2015.

Mike Giles and Paul Glasserman. Smoking adjoints: Fast Monte Carlo greeks. *Risk*, 19(1):88–92, 2006.

Emmanuel Gobet and Rémi Munos. Sensitivity analysis using Itô–Malliavin calculus and martingales, and application to stochastic optimal control. *SIAM Journal on control and optimization*, 43(5): 1676–1713, 2005.

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`.

Markus Heinonen, Cagatay Yildiz, Henrik Mannerström, Jukka Intosalmi, and Harri Lähdesmäki. Learning unknown ode models with gaussian processes. In *International conference on machine learning*, pp. 1959–1968. PMLR, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33:4370–4380, 2020.

Patrick Kidger, Ricky TQ Chen, and Terry J Lyons. " hey, that's not an ode": Faster ode adjoints via seminorms. In *ICML*, pp. 5443–5452, 2021a.

Patrick Kidger, James Foster, Xuechen Chen Li, and Terry Lyons. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021b.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Daniel J Lea, Myles R Allen, and Thomas WN Haine. Sensitivity analysis of the climate of a chaotic system. *Tellus A: Dynamic Meteorology and Oceanography*, 52(5):523–532, 2000.

John M. Lee. *Introduction to Smooth Manifolds*. Springer, 2012.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=XVjTT1nw5z`.

Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

Beatrix Miranda Ginn Nielsen, Anders Christensen, Andrea Dittadi, and Ole Winther. Diffenc: Variational diffusion with a learned encoder. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=8nxy1bQWTG`.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Walter Rudin. *Differential equations, dynamical systems, and linear algebra*. Tata McGraw-Hill Education, 2006.

Subham Sekhar Sahoo, Aaron Gokaslan, Christopher De Sa, and Volodymyr Kuleshov. Diffusion models with learned adaptive noise processes, 2024. URL https://openreview.net/forum?id=8gZtt8nrpI.

Raghav Singhal, Mark Goldstein, and Rajesh Ranganath. Where to diffuse, how to diffuse, and how to get back: Automated learning for multivariate diffusions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=osei3IzUia.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021c. URL https://openreview.net/forum?id=PxTIG12RRHS.

Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.

Luhuan Wu, Brian Trippe, Christian A Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 31372–31403, 2023.

Jichuan Yang and Harold J Kushner. A Monte Carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems. *SIAM journal on control and optimization*, 29(5): 1216–1249, 1991.

Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. Ode2vae: Deep generative second order odes with Bayesian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Xi Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis L Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modeling. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Detailed Description of the Posterior Process

A.1 Marginal Distribution

The posterior marginal distribution is defined by a transformation F_φ from noise ε for each t and $\mathbf{X}$:

$$z_t = F_\varphi(\varepsilon, t, \mathbf{X}), \quad \varepsilon \sim q(\varepsilon) = \mathcal{N}(\varepsilon; 0, I). \tag{14}$$

Assuming that F_φ is invertible and differentiable in ε for each t and $\mathbf{X}$. Then, we can use the standard change of variables result for distributions (Rudin, 2006; Bogachev, 2007) to derive the density of z_t

$$q_\varphi(z_t|\mathbf{X}) = q(\varepsilon)\Big|_{\varepsilon = F_\varphi^{-1}(z_t, t, \mathbf{X})} \cdot \left| \frac{\partial F_\varphi^{-1}(z_t, t, \mathbf{X})}{\partial z_t} \right|, \tag{15}$$

where F_φ^{-1} is the inverse of F_φ and $|\cdot|$ denotes the absolute value of the determinant. This is a well-known result of normalizing flows (Papamakarios et al., 2021) applied to our setting.

A.2 Conditional ODE

We leverage results based on *flows*, solutions to ODEs, inspired by Biloš et al. (2021); Lee (2012); Bartosh et al. (2024) to turn a flow F_φ into its corresponding generating ODE $\bar{f}_\varphi$.

Proposition A.1. *Let $F_\varphi : \mathbb{R}^D \times \mathbb{R} \times \mathrm{X} \mapsto \mathbb{R}^D$ be a smooth function such that $F_\varphi(\cdot, t, \mathbf{X})$ is invertible $\forall t, \mathbf{X}$. Then, $F_\varphi(\cdot, t, \mathbf{X})$ is a* flow *with the infinitesimal generator*

$$\frac{dF_\varphi(\varepsilon, t, \mathbf{X})}{dt}, \tag{16}$$

which is a smooth vector field f such that

$$\frac{\partial F_\varphi(\varepsilon, t, \mathbf{X})}{\partial t} = f(F_\varphi(\varepsilon, t, \mathbf{X}), t, \mathbf{X}). \tag{17}$$

Proof. The result is a consequence of the Fundamental Theorem on Flows (Lee, 2012, Theorem 9.12) for the flow defined by $F_\varphi(\cdot, t, \mathbf{X})$. ☐

Using Proposition A.1 with $\varepsilon = F_\varphi^{-1}(z_t, t, \mathbf{X})$, identifying that $F_\varphi(F_\varphi^{-1}(z_t, t, \mathbf{X}), t, \mathbf{X}) = z_t$, lets us construct the conditional ODE whose solution matches those of the flow $z_t = F_\varphi(\varepsilon, t, \mathbf{X})$:

$$dz_t = \bar{f}_\varphi(z_t, t, \mathbf{X})dt, \quad \text{where} \tag{18}$$

$$\bar{f}_\varphi(z_t, t, \mathbf{X}) = \left. \frac{\partial F_\varphi(\varepsilon, t, \mathbf{X})}{\partial t} \right|_{\varepsilon = F_\varphi^{-1}(z_t, t, \mathbf{X})}. \tag{19}$$

Initializing using $z_0 \sim q_\phi(z_0|\mathbf{X})$ and solving the above conditional ODE in Equation (18) ensures $z_t \overset{d}{=} F_\varphi(\varepsilon, t, \mathbf{X})$ for $\varepsilon \sim q(\varepsilon)$, where $\overset{d}{=}$ denotes equal in distribution.

A.3 Conditional SDE

To derive the conditional SDE we leverage a result by Song et al. (2021a) that relates the marginal distributions of an SDE with its corresponding (probability flow) ODE, restated in Proposition A.2 for convenience.

Proposition A.2. *The marginal distributions $p_t(z_t)$ of the SDE*

$$dz_t = h(z_t, t)dt + g(z_t, t)d\mathbf{w}, \tag{20}$$

are, under suitable conditions on the drift and diffusion terms h and g, identical to the distributions induced by the ODE

$$dz_t = \left(h(z_t, t) - \frac{1}{2} g(z_t, t) g^\top(z_t, t) \nabla_{z_t} \log p_t(z_t) - \frac{1}{2} \nabla_{z_t} \cdot \left[g(z_t, t) g^\top(z_t, t) \right] \right) dt. \tag{21}$$

Proof. See Song et al. (2021a, Appendix D.1). □

Note that this equivalence lets us easily find the corresponding (probability flow) ODE. However, we can equally well use it in the reverse order with a known ODE to construct the corresponding SDE by selecting a diffusion term g and using Equation (21) to compute the drift term h.

For the prior process diffusion term $g_\theta(z_t, t)$ with velocity field Equation (18) we apply Proposition A.2 to find the conditional SDE with marginals $q_\varphi(z_t|\mathbf{X})$

$$dz_t = f_{\theta,\varphi}(z_t, t, \mathbf{X})dt + g_\theta(z_t, t)d\mathbf{w}, \quad \text{where} \tag{22}$$

$$f_{\theta,\varphi}(z_t, t, \mathbf{X}) = \bar{f}_\varphi(z_t, t, \mathbf{X}) + \frac{1}{2}g_\theta(z_t, t)g_\theta^\top(z_t, t)\nabla_{z_t}\log q_\varphi(z_t|\mathbf{X}) + \frac{1}{2}\nabla_{z_t}\cdot\left[g_\theta(z_t, t)g_\theta^\top(z_t, t)\right].$$

This result follows by simply identifying terms in Proposition A.2 with our specific case

$$h(z_t, t) \equiv f_{\theta,\varphi}(z_t, t, \mathbf{X}),$$
$$g(z_t, t) \equiv g_\theta(z_t, t).$$

A.4 PARAMETERIZATION

As discussed in Section 3.2, evaluating the posterior SDE (Equation (9)) requires access to the conditional score function $\nabla_{z_t}\log q_\varphi(z_t|\mathbf{X})$. Since we parameterize the posterior marginals $q_\varphi(z_t|\mathbf{X})$ implicitly through an invertible transformation F_φ (Equation (7)) of a random variable $\varepsilon \sim q(\varepsilon)$ into z_t, the score function can in general be computed using Equation (15):

$$\nabla_{z_t}\log q_\varphi(z_t|\mathbf{X}) = \nabla_{z_t}\left[\log q(\varepsilon)\Big|_{\varepsilon=F_\varphi^{-1}(z_t,t,\mathbf{X})} + \log\left|\frac{\partial F_\varphi^{-1}(z_t, t, \mathbf{X})}{\partial z_t}\right|\right]. \tag{23}$$

The first term in Equation (23) represents the log-density of the noise distribution, which is straightforward to compute. The second term is the log-determinant of the Jacobian matrix of the inverse transformation. If this Jacobian is available, the score function can be efficiently computed using automatic differentiation tools.

In this work, we parameterize the function F_φ as follows:

$$F_\varphi(\varepsilon, t, \mathbf{X}) = \mu_\varphi(\mathbf{X}, t) + \sigma_\varphi(\mathbf{X}, t)\varepsilon, \quad \text{and} \quad F_\varphi^{-1}(z_t, t, \mathbf{X}) = \sigma_\varphi^{-1}(\mathbf{X}, t)\big(z_t - \mu_\varphi(\mathbf{X}, t)\big) \tag{24}$$

where σ_φ is a matrix valued function.

Substituting this parameterization into Equation (23), the score function simplifies to:

$$\nabla_{z_t}\log q_\varphi(z_t|\mathbf{X}) = \left[\sigma_\varphi(\mathbf{X}, t)\sigma_\varphi^\top(\mathbf{X}, t)\right]^{-1}\left(\mu_\varphi(\mathbf{X}, t) - z_t\right) = -\sigma_\varphi^{-\top}(\mathbf{X}, t)\varepsilon\big|_{\varepsilon=F_\varphi^{-1}(z_t,t,\mathbf{X})}. \tag{25}$$

The SDE Matching framework allows for other, potentially more flexible, parameterizations of the function F_φ. However, we leave the exploration of such alternatives for future research.

B DIFFUSION MODELS AS LATENT SDES

In contrast to Latent SDEs, recent advancements in diffusion and flow-based modeling demonstrate that continuous dynamics can be learned efficiently in a simulation-free manner, without requiring numerical integration. At first glance, these generative models seem quite different from Latent SDEs. Instead of generating data autoregressively, they produce a single data point through an iterative refinement process, gradually reconstructing corrupted data. However, similar to Latent SDEs, continuous diffusion models can be thought of as learning an SDE in a latent space. This connection is useful for understanding and developing SDE Matching.

Conventional diffusion models are defined through two processes: the forward (or noising) process and the reverse (or generative) process. The forward process takes a data point $x \in \mathbb{R}^D$ and perturbs it over time by injecting noise. This dynamic can be expressed as an SDE with a linear drift term and state-independent diffusion term:

$$dz_t = f(t)z_t dt + g(t)d\mathbf{w}, \tag{26}$$

where $f : [0, 1] \mapsto \mathbb{R}$, $g : [0, 1] \mapsto \mathbb{R}_+$, and $q(z_0|x) \approx \delta(z_0 - x)$. Due to the linearity of Equation (26), the conditional marginal distribution is available in closed form $q(z_t|x) = \mathcal{N}(z_t; \alpha_t x, \sigma_t^2 I)$, where α_t, σ_t are determined by $f(t), g(t)$. The functions $f(t)$ and $g(t)$ are typically chosen to ensure that $q(z_1|x) \approx \mathcal{N}(z_1; 0, I)$. The generative process then reverses this transformation, starting from the prior $p(z_1) = \mathcal{N}(z_1; 0, I)$ and following the (marginal) reverse SDE:

$$dz_t = [f(t)z_t - g^2(t)\nabla_{z_t} \log q(z_t)]dt + g(t)d\bar{\mathbf{w}}. \tag{27}$$

Here, $\bar{\mathbf{w}}$ denotes a standard Wiener process, where time flows backward. Diffusion models approximate this reverse process by learning the score function $\nabla_{z_t} \log q(z_t)$ using a denoising score matching loss:

$$\mathop{\mathbb{E}}_{u(t)q(z_t|x)} \left[\left\| s_\theta(z_t, t) - \nabla_{z_t} \log q(z_t|x) \right\|_2^2 \right], \tag{28}$$

where $u(t)$ represents a uniform distribution over the interval $[0, 1]$, and $s_\theta : \mathbb{R}^d \times [0, 1] \mapsto \mathbb{R}^d$ is a learnable approximation of the score function.

A key property of the denoising score matching objective is that it can be learned in a simulation-free manner. Thanks to the simplicity of the forward process, instead of integrating the SDE in Equation (26), we can directly sample from $q(z_t|x)$ and estimate the expectation in Equation (28) using the Monte Carlo method. This property distinguishes diffusion models from Latent SDEs. However, as we will see diffusion models are in fact a special case of Latent SDEs.

To demonstrate this connection we: (1) derive the reverse SDE for the conditional forward process (Equation (26)), and (2) invert the time direction for the entire model, mapping the time interval $[0, 1]$ into $[1, 0]$.

First, the reverse SDE for the forward process can be obtained from the Fokker–Planck equation, similar to the derivation of the marginal reverse SDE in Equation (27).

Second, after inverting the time direction, we obtain:

$$dz_t = [f(t)z_t + g^2(t)\nabla_{z_t} \log q(z_t|x)]dt + g(t)d\mathbf{w}. \tag{29}$$

Except for the change in time direction, the generative process remains unchanged:

$$dz_t = [f(t)z_t + g^2(t)s_\theta(z_t, t)]dt + g(t)d\mathbf{w}. \tag{30}$$

From this perspective, diffusion models can be seen as a special case of Latent SDEs with a specific form of the posterior process and only a single observation x at time step $t = 1$. Moreover, substituting this parameterization of the prior and reverse processes into the Latent SDE objective in Equation (3), we find that it corresponds to a reweighted denoising score matching objective:

$$\begin{aligned} &\mathop{\mathbb{E}}_{q(\mathbf{Z}|x)} \left[\int_0^1 \frac{1}{2g^2(t)} \left\| \cancel{f(t)z_t} + g^2(t)s_\theta(z_t, t) - \cancel{f(t)z_t} - g^2(t)\nabla_{z_t} \log q(z_t|x) \right\|_2^2 dt \right] + C \\ &= \mathop{\mathbb{E}}_{u(t)q(z_t|x)} \left[\frac{g^2(t)}{2} \left\| s_\theta(z_t, t) - \nabla_{z_t} \log q(z_t|x) \right\|_2^2 \right] + C. \end{aligned} \tag{31}$$

This result is particularly meaningful since it is well known that denoising score matching (Equation (28)) corresponds to a reweighted variational bound on the likelihood of diffusion models Song et al. (2021b).

This connection between diffusion models and Latent SDEs, along with the fact that diffusion models can be trained in a simulation-free manner, motivates us to extend and develop a simulation-free framework for training Latent SDEs.

Algorithm 1 Training of SDE Matching

Require: $\mathbf{X}$, $p_\theta(z_0)$, h_θ, g_θ, $p_\theta(x_t|z_t)$, F_φ
for learning iterations **do**
  # Calculation of prior loss $\mathcal{L}_{\text{prior}}$
  $\varepsilon_0 \sim \mathcal{N}(0, I)$
  $z_0 = F_\varphi(\varepsilon_0, 0, \mathbf{X})$
  $\mathcal{L}_{\text{prior}} = \mathrm{D}_{\text{KL}}\big(q_\varphi(z_0|\mathbf{X}) \| p_\theta(z_0)\big)$
  # Calculation of diffusion loss $\mathcal{L}_{\text{diff}}$
  $t \sim u(t)$
  $\varepsilon_t \sim \mathcal{N}(0, I)$
  $z_t = F_\varphi(\varepsilon_t, t, \mathbf{X})$
  $r_{\theta,\varphi}(z_t, t, \mathbf{X}) = g_\theta^{-1}(z_t, t)\,(h_\theta(z_t, t) - f_{\theta,\varphi}(z_t, t, \mathbf{X}))$
  $\mathcal{L}_{\text{diff}} = \frac{1}{2}\left\| r_{\theta,\varphi}(z_t, t, \mathbf{X})\right\|_2^2$
  # Calculation of reconstruction loss $\mathcal{L}_{\text{rec}}$
  $i \sim u(i)$
  $\varepsilon_{t_i} \sim \mathcal{N}(0, I)$
  $z_{t_i} = F_\varphi(\varepsilon_{t_i}, t_i, \mathbf{X})$
  $\mathcal{L}_{\text{rec}} = -N \log p_\theta(x_{t_i}|z_{t_i})$
  # Optimization
  $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{prior}}$
  Gradient step on θ and φ w.r.t. $\mathcal{L}$
end for

C Implementation Details

For the experiments presented in Section 4, we follow the setup from Li et al. (2020) for both the 3D stochastic Lorenz attractor and the motion capture dataset. We use the same parameterizations and hyperparameters for the prior process, including the observation models $p_\theta(x_t|z_t)$ and the coordinate-wise diagonal diffusion term $g_\theta(z_t, t)$.

We make a slight modification to the parameterization of the posterior process. First, we do not use an encoder, i.e., a separate network that defines the initial conditions of the posterior process. Second, Li et al. (2020) propose using a time-reversal GRU layer Cho et al. (2014) to aggregate information from observations into what they refer to as a context. This context is then used to predict the drift term of the posterior SDE via an MLP. To maintain consistency with Cho et al. (2014), we use the same hyperparameters for the GRU and MLP. However, instead of predicting the drift term, we use the MLP to predict the functions μ_φ and $\log \sigma_\varphi$ (Equation (24)) based on the context computed from all observations and the time step t. We use diaganal parameterization σ_φ.

For optimization, we use the same training hyperparameters, including the Adam optimizer Kingma (2014) and the same number of training iterations. However, we do not apply any additional regularization, reweighting, or annealing techniques.

We believe that there exists more efficient ways to parameterize the posterior process for SDE Matching. However, in this work, our primary goal was to provide a fair comparison with the standard Latent SDE training approach rather than focusing on specific design choices.

We summarize the training procedure in Algorithm 1.