Probabilistic Forecasting with Stochastic Interpolants and Föllmer Processes

Yifan Chen [*1] **Mark Goldstein** [*1] **Mengjian Hua** [*1]
Michael S. Albergo [1] **Nicholas M. Boffi** [1]
Eric Vanden-Eijnden [1]

Abstract

We propose a framework for probabilistic forecasting of dynamical systems based on generative modeling. Given observations of the system state over time, we formulate the forecasting problem as sampling from the conditional distribution of the future system state given its current state. To this end, we leverage the framework of stochastic interpolants, which facilitates the construction of a generative model between an arbitrary base distribution and the target. We design a fictitious, non-physical stochastic dynamics that takes as initial condition the current system state and produces as output a sample from the target conditional distribution in finite time and without bias. This process therefore maps a point mass centered at the current state onto a probabilistic ensemble of forecasts. We prove that the drift coefficient entering the stochastic differential equation (SDE) achieving this task is non-singular, and that it can be learned efficiently by square loss regression over the time-series data. We show that the drift and the diffusion coefficients of this SDE can be adjusted after training, and that a specific choice that minimizes the impact of the estimation error gives a Föllmer process. We highlight the utility of our approach on several complex, high-dimensional forecasting problems, including stochastically forced Navier-Stokes and video prediction on the KTH and CLEVRER datasets. The code is available at https://github.com/interpolants/forecasting.

[*]Equal contribution [1]Courant Institute of Mathematical Sciences, New York University, New York, NY, USA. Correspondence to: Eric Vanden-Eijnden <eve2@nyu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Forecasting the future state of a dynamical system given complete or partial information about the current state is a ubiquitous problem across science and engineering, with diverse applications in climate modeling (Smagorinsky, 1963; Palmer et al., 1993; Gneiting et al., 2005; Pathak et al., 2022), fluid dynamics (Buaria & Sreenivasan, 2023), video prediction (Oprea et al., 2020; Finn et al., 2016; Lee et al., 2018), and extrapolation of time series data (Lim & Zohren, 2021; Masini et al., 2023). Forecasting can be performed *deterministically*, whereby the goal is to compute a single prediction for the future state (Giannakis et al., 2023), or *probabilistically*, whereby the goal is to predict a distribution over future states consistent with the current information (Gneiting & Katzfuss, 2014). Probabilistic forecasting is the natural formulation when the underlying system dynamics are stochastic, when the full system state cannot be measured, or when measurements are corrupted by noise, as is the case for most real-world systems of interest. Moreover, while deterministic forecasting appears to be a simpler problem, recent works have uncovered difficulties with deterministic forecasting methods when the underlying dynamics are chaotic (Jiang et al., 2023).

Motivated by the recent success of generative models built upon dynamical transport of measure, such as score-based diffusion (Ho et al., 2020; Song et al., 2020), flow matching (Lipman et al., 2022), and stochastic interpolants (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023), here we introduce a generative modeling approach for probabilistic forecasting. This approach maps the current state of the system onto the ensemble of possible future outcomes after a fixed time lag. From a transport perspective, this requires pushing a point mass measure onto a probability distribution with larger support. In what follows, we show that the interpolant framework enables us to design an artificial dynamics that performs this task using a stochastic differential equation. In practice, the drift fields entering the SDEs we introduce can be learned via square loss regression. We also show that the diffusion coefficient in these SDEs can be tuned *a-posteriori* (i.e. without having to retrain a drift). We show that a specific choice that minimizes the impact of the estimation error recovers a Föllmer process (Föllmer,

1986), a specific instantiation of the Schrödinger bridge problem (Schrödinger, 1932; Léonard, 2014) in which the base distribution is a point mass measure.

To demonstrate the utility and scalability of our approach, we consider several examples: an illustrative low-dimensional scenario based on a multi-modal jump diffusion process, a high-dimensional problem generated by a stochastic Navier-Stokes equation on the torus, and video generation on the KTH (Schuldt et al., 2004) and CLEVRER datasets (Yi et al., 2019). In the case of Navier-Stokes, we show that our probabilistic forecasting method is capable of reproducing quantitative metrics, such as the enstrophy spectrum of the dataset, using either high- or low-resolution measurement of the current system state. In each case, we highlight that we are able to obtain diverse samples consistent with the conditional distribution of interest, and demonstrate the need for probabilistic, as opposed to deterministic, forecasting. For the video generation tasks, we show that our learned models are more effective than standard conditional generative modeling. We also show that this forecasting procedure can be iterated autoregressively without retraining to compute a predicted trajectory. Our **main contributions** can be summarized as follows:

- We design new generative models for probabilistic forecasting based on stochastic differential equations (SDEs) that map a point mass measure to a distribution with full support by incorporating stochasticity in a principled way, enabling us to initialize the SDE directly at the measured system state.
- We prove that the drifts entering these SDEs can be learned via square loss regression over the data, and that the resulting loss has bounded variance.
- We show that the drift and noise terms in these SDEs can be adjusted post-training, and that the specific choice of noise that minimizes the Kullback-Leibler (KL) divergence between the path measures of the exact forecasting process and the estimated one is realizable. We show that this drift and diffusion pair gives a Föllmer process.
- We validate our theoretical results empirically on several challenging high-dimensional forecasting tasks, including the Navier-Stokes equation and video prediction.

2. Related Work

There is a vast body of literature on forecasting. Methodologies can be broadly classified into two main categories: forecasting a single output, typically achieved through regression and operator theoretic approaches (Kutz et al., 2016; Alexander & Giannakis, 2020; Li et al., 2021), and probabilistic forecasting (Gneiting & Katzfuss, 2014), based on stochastic and generative modeling.

Deterministic vs Probabilistic Forecasting. For the class of methodologies focused on generating a single output, a widely used approach is regression or supervised learning, which entails directly learning a map in the state space. One can also work in the space of probability densities or functionals on the state space, under the setting of the Frobenius-Perron or Koopman operator approach (Dellnitz & Junge, 1999; Kaiser et al., 2021); this leads to linear dynamics in an infinite-dimensional space, and forecasting reduces to finding a tractable finite-dimensional approximation. In all formulations, the key is to identify and learn accurate representations of the dynamics, for example through nonparametric approaches such as diffusion maps (Berry et al., 2015) and kernel regression (Alexander & Giannakis, 2020), or parametric approaches such as linear regression, dynamical mode decomposition (Kutz et al., 2016), neural networks (Li et al., 2021; Gu et al., 2021) and operators (Lu et al., 2021; Jiang et al., 2023; Li et al., 2020). Ultimately, these approaches produce a function that maps the current state to a single output which is the deterministic forecast. Many methods under this category train models with the MSE (mean square error) or RMSE (root mean square error) as the objective, but these losses may be poor signals for training forecasters for chaotic systems (Jiang et al., 2023).

For dynamics that are inherently stochastic, or with incomplete information, a stochastic forecast is important to incorporate uncertainties. This stochasticity can be introduced by fitting a probabilistic model, such as a stochastic process or a graphical model, to data. They can also be approached by stochastic Koopman operators (Wanner & Mezic, 2022; Zhao & Jiang, 2023). Although many approaches have been proposed, most of them target low-dimensional problems or simple conditional statistics such as Gaussians.

Probabilistic Forecasting with Generative Models. Recently, generative modeling techniques in machine learning have received increasing attention for handling high-dimensional, complex distributions. Probabilistic forecasting can be seen as a form of the conditional generation problem. Various conditional generative models, including conditional GAN (Mirza & Osindero, 2014), VAE (Sohn et al., 2015), and normalizing flows (de Bézenac et al., 2020; Kidger et al., 2021) have been developed. More recently, diffusion generative models (Ho et al., 2020; Song et al., 2020) have gained popularity due to their state-of-the-art performance. The stochastic interpolant methodology, which is related to contemporary work like flow matching (Lipman et al., 2022), is a general framework that encompasses diffusion models. In the literature, there has been some work to develop conditional models for time series (Rasul et al., 2021; Lienen et al., 2023), including diffusion (Ho et al., 2022a; Blattmann et al., 2023) and flow matching models for video prediction (Davtyan et al., 2023). These approaches learn ODEs or SDEs that map a Gaussian base

to the conditional distribution of interest. There are also works that forecast stochastic dynamics by adding noise to the neural network layers (Rühling Cachay et al., 2024) using techniques such as dropout. In contrast, we incorporate stochasticity rigorously through interpolants and direct SDE modeling. Our construction of the stochastic generative model, which maps the current state to the distribution of the forecasted state, is new and can be seen as a direct stochastic extension of the deterministic map approach that is predominant in single output forecasting.

Föllmer processes. The construction of SDEs that map a point mass to a target distribution dates back at least to the Föllmer process (Föllmer, 1986), which is a particular solution of the Schrödinger bridge problem (Schrödinger, 1932; Léonard, 2014; Chen et al., 2021) that minimizes the relative entropy with respect to the Wiener process. This approach has the desirable feature that it offers an entropy-regularized solution to the optimal transport problem. For this reason, the concepts of the Föllmer process and the Schrödinger bridge have found many applications in sampling densities with unknown normalization constants (Zhang & Chen, 2021; Huang et al., 2021; Jiao et al., 2021; Vargas et al., 2023), generative modeling (Tzen & Raginsky, 2019; Wang et al., 2021; De Bortoli et al., 2021; Liu et al., 2023; Peluchetti, 2023; Shi et al., 2024) as well as stochastic analysis and functional inequalities (Lehec, 2013; Eldan & Lee, 2018; Eldan et al., 2020). In this article, we show that the stochastic interpolant framework offers a simple way to construct new types of Föllmer processes. In addition, we give a new interpretation of Föllmer processes as minimizers of a KL divergence between the path measure of an SDE that forecasts exactly and the approximate, learned SDE. These results support the uses of Föllmer processes in probabilistic forecasting.

3. Setup and Main Results

3.1. Conditional PDF

Assume that we are given a joint probability density function (PDF) $\rho(x_0, x_1)$ supported on $\mathbb{R}^d \times \mathbb{R}^d$ and strictly positive everywhere. Our aim is to design a generative model to sample the conditional PDF of x_1 given x_0:

$$\rho_c(x_1|x_0) = \frac{\rho(x_0, x_1)}{\rho_0(x_0)} > 0, \tag{1}$$

where $\rho_0(x_0) = \int_{\mathbb{R}^d} \rho(x_0, x_1) dx_1 > 0$.[1] Phrasing the question this general way allows us to consider several instantiations of interest by appropriately defining the joint PDF ρ.

[1]Our approach actually requires that $x_1|x_0$ has a positive density, but (x_0, x_1) does not have to. In particular, we can target any density $\rho_1(x_1) > 0$ by drawing (x_0, x_1) from the joint distribution $\mu(dx_0, dx_1) = \mu_0(dx_0)\rho_1(x_1)dx_1$ with any μ_0 (e.g. a point mass), since we then have $\rho_c(x_1|x_0) = \rho_1(x_1)$.

In particular, we study the following three problem settings:

Probabilistic forecasting. Suppose that we are given a discrete time-series $\{\ldots, x_{-\tau}, x_0, x_\tau, \ldots\} = \{x_{k\tau}\}_{k\in\mathbb{Z}}$ with each $x_{k\tau} \in \mathbb{R}^d$ containing, for example, daily weather measurements or video frames, acquired every lag-time $\tau > 0$. Assume that this time-series is a stationary process[2] and that the law of successive observations $(x_{k\tau}, x_{(k+1)\tau})$ is captured by the joint PDF $\rho(x_{k\tau}, x_{(k+1)\tau})$. Then sampling from $\rho_c(\cdot|x_{k\tau})$ produces the ensemble of forecasts $x_{(k+1)\tau}$ given the observation $x_{k\tau}$.

Signal recovery from corrupted data. Suppose that, given clean data $x_1 \in \mathbb{R}^d$, we observe the corrupted signal $x_0 \in \mathbb{R}^d$ (e.g. a low-resolution or noisy image obtained from a high-resolution image). If we assume that the joint PDF of (x_0, x_1) is $\rho(x_0, x_1)$, then sampling from $\rho_c(\cdot|x_0)$ produces
corrupted signal x_0.

Probabilistic forecasting from noisy observations. We can combine the previous two setups if we assume that we are given a discrete-time-series $\{x_{k\tau}, \tilde{x}_{k\tau}\}_{k\in\mathbb{Z}}$ with $x_{k\tau}$ the clean data and $\tilde{x}_{k\tau}$ the corrupted observation at time $k\tau$. If we assume that the joint PDF of $(\tilde{x}_{k\tau}, x_{(k+1)\tau})$ is $\rho(\tilde{x}_{k\tau}, x_{(k+1)\tau})$, then sampling from $\rho_c(\cdot|\tilde{x}_{k\tau})$ produces the ensemble of clean forecasts $x_{(k+1)\tau}$ given the noisy observation $\tilde{x}_{k\tau}$.

3.2. Generation with Stochastic Interpolants

The generative models that we develop here are based on stochastic differential equations that map a fixed initial condition $X_{s=0} = x_0$ to samples from the conditional distribution $X_{s=1} \sim \rho_c(\cdot|x_0)$. Towards the design of such SDEs, we first introduce the *stochastic interpolant*

$$I_s = \alpha_s x_0 + \beta_s x_1 + \sigma_s W_s \tag{2}$$

where $(x_0, x_1) \sim \rho(x_0, x_1)$ and $W = (W_s)_{s\in[0,1]}$ is a Wiener process with $W \perp (x_0, x_1)$. In addition, we impose that $\alpha, \beta, \sigma \in C^1([0,1])$ satisfy the boundary conditions $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = \sigma_1 = 0$. To facilitate some calculations, we assume that $\dot{\beta}_s > 0$ for all $s \in (0,1]$ and $\dot{\sigma}_s < 0$ for all $s \in [0,1]$. Here we will use $\alpha_s = \sigma_s = 1 - s$, and $\beta_s = s$ or $\beta_s = s^2$ (see Appendix B.1). This second choice for β_s has some advantages that we discuss below.

The boundary conditions on α, β, and σ guarantee that $I_{s=0} = x_0$ and $I_{s=1} = x_1$, so that the probability distribution of $I_s|x_0$ bridges the point mass measure at x_0 to $\rho_c(\cdot|x_0)$ as s varies from 0 to 1. The following result shows that this probability distribution is also the law of the solution to a SDE that can be used as a generative model.

[2]This assumption can be relaxed; see Appendix A.

Theorem 3.1. *Let $b_s(x, x_0)$ be the unique minimizer over all $\hat{b}_s(x, x_0)$ of the objective*

$$L_b[\hat{b}_s] = \int_0^1 \mathbb{E}\big[|\hat{b}_s(I_s, x_0) - R_s|^2\big]ds, \tag{3}$$

where $\mathbb{E}$ denotes an expectation over $(x_0, x_1) \sim \rho$ and W with $(x_0, x_1) \perp W$, I_s is given in (2), and we defined[3]

$$R_s = \dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s W_s, \tag{4}$$

Then the solutions to the SDE

$$dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s, \quad X_{s=0} = x_0, \tag{5}$$

are such that Law(X_s) = Law$(I_s|x_0)$ for all $(s, x_0) \in [0, 1] \times \mathbb{R}^d$. In particular, $X_{s=1} \sim \rho_c(\cdot|x_0)$.

This theorem is proven in Appendix B.2. The result is formulated in a way that is tailored to practical approximation of the drift b_s, since the objective (3) can be estimated empirically by generating samples of I_s and R_s using sample pairs (x_0, x_1) from ρ and realizations of $W_s \overset{d}{=} \sqrt{s}z$ with $z \sim \mathsf{N}(0, Id)$. That is, b_s may be learned over neural networks by minimizing the simulation-free loss (3) over the parameters. It is easy to see that the minimizer is given by

$$b_s(x, x_0) = \mathbb{E}^{x_0}[R_s|I_s = x], \tag{6}$$

where $\mathbb{E}^{x_0}[\cdot|I_s = x]$ denotes an expectation over $x_1 \sim \rho_c(\cdot|x_0)$ and W with $x_1 \perp W$ conditioned on the event $I_s = x$. The drift (6) is well-defined for all $(s, x, x_0) \in [0, 1] \times \mathbb{R}^d \times \mathbb{R}^d$, and we show in Appendix B.3 that enforcing $\dot{\beta}_0 = 0$ offers additional control on the boundedness and Lipschitz constant of b at the initial time. We observe empirically that this has computational advantages at both optimization and sampling time; see the experiments in Appendix C.

3.3. Generalizations with Tunable Diffusion

We now show that learning the drift coefficient (6) gives access to a broader set of SDEs to use as a generative model beyond just (5), and that selecting from them has appealing theoretical motivation.

Let $\rho_s(x|x_0)$ be the PDF of $X_s \overset{d}{=} I_s|x_0$. From (5), ρ_s solves the Fokker-Planck equation

$$\partial_s \rho_s + \nabla \cdot (b_s(x, x_0)\rho_s) = \tfrac{1}{2}\sigma_s^2 \Delta \rho_s. \tag{7}$$

Given a candidate diffusion coefficient g_s, we can use the identity $\frac{1}{2}\sigma_s^2\Delta\rho_s = \frac{1}{2}g_s^2\Delta\rho_s - \frac{1}{2}(g_s^2 - \sigma_s^2)\nabla \cdot (\rho_s \nabla \log \rho_s)$ to trade diffusion for transport in (7). This construction leads to a family of SDEs with tunable diffusion

[3]Here and below the dot denotes derivative with respect to s.

Theorem 3.2. *Given any $g \in C^0([0, 1])$ such that $\lim_{s\to 0^+} s^{-1}[g_s^2 - \sigma_s^2]$ and $\lim_{s\to 1^-} g_s^2\sigma_s^{-1}$ exist, define*

$$b_s^g(x, x_0) = b_s(x, x_0) + \tfrac{1}{2}(g_s^2 - \sigma_s^2)\nabla \log \rho_s(x|x_0) \tag{8}$$

where $b_s(x, x_0)$ is the minimizer of (3) given in (6) and $\rho_s(x|x_0)$ is the PDF of $X_s \overset{d}{=} I_s|x_0$. Then the solutions to the SDE

$$dX_s^g = b_s^g(X_s^g, x_0)ds + g_s dW_s, \quad X_{s=0}^g = x_0, \tag{9}$$

are such that Law(X_s^g) = Law(X_s) = Law$(I_s|x_0)$ for all $(s, x_0) \in [0, 1] \times \mathbb{R}^d$. In particular $X_{s=1}^g \sim \rho_c(\cdot|x_0)$.

This theorem is proven in Appendix B.4, where we explain why the conditions on g_s guarantee that the SDE (9) is well-posed. Working with this SDE requires the score $\nabla \log \rho_s$. Interestingly, this score can be expressed in terms of the drift b_s. A direct calculation reported in Appendix B.4 shows that

$$\nabla \log \rho_s(x|x_0) = A_s\left[\beta_s b_s(x, x_0) - c_s(x, x_0)\right], \tag{10}$$

where

$$\begin{aligned} A_s &= [s\sigma_s(\dot{\beta}_s\sigma_s - \beta_s\dot{\sigma}_s)]^{-1}, \\ c_s(x, x_0) &= \dot{\beta}_s x + (\beta_s\dot{\alpha}_s - \dot{\beta}_s\alpha_s)x_0. \end{aligned} \tag{11}$$

Using (10) in (8) shows that, to work with the SDE (9), we can estimate b first and then adjust both the noise amplitude g_s and the drift b^g *a-posteriori* without having to retrain b.[4] This offers flexibility at sampling time that can be leveraged to maximize performance, as shown in our numerical experiments below.

3.4. KL Optimization and Föllmer Processes

In light of Theorem 3.2, it is natural to ask if a specific choice of g_s is optimal in a suitable sense. To provide one answer to this question, we consider the KL divergence between the path measure of the process $X^g = (X_s^g)_{s\in[0,1]}$ (which solves the ideal SDE (9)) and the path measure of the process $\hat{X}^g = (\hat{X}_s^g)_{s\in[0,1]}$ (which solves an approximate, learned version of (9) obtained through an estimate $\hat{b}$ of b). Because Law(X_s^g) = Law$(I_s|x_0)$ for all $(s, x_0) \in [0, 1] \times \mathbb{R}^d$, this KL divergence is given by (see Appendix B.5 for details)

$$D_{\mathrm{KL}}(X^g||\hat{X}^g) = \int_0^1 \frac{|1 + \frac{1}{2}\beta_s A_s(g_s^2 - \sigma_s^2)|^2 L_s}{2|g_s|^2} ds \tag{12}$$

where $L_s = \mathbb{E}^{x_0}\big[|\hat{b}_s(I_s, x_0) - b_s(I_s, x_0)|^2\big]$. Eq. (12) measures how the estimation error on b impacts the generative

[4]Using (10) in (8) requires some care at $s = 0$ and $s = 1$ due to the factor $[s\sigma_s]^{-1}$ in A_s, but this leads to no issue see Algorithm 2 and also Appendices B.1 and B.4.

process, and as such it is natural to minimize it over g. Since L_s is independent of g_s, this minimization can be performed analytically. The result is that (12) is minimized if we set $g_s = g_s^{\mathrm{F}}$ with

$$g_s^{\mathrm{F}} = \left| 2s\sigma_s(\beta_s^{-1}\dot{\beta}_s\sigma_s - \dot{\sigma}_s) - \sigma_s^2 \right|^{1/2}. \tag{13}$$

This expression is well-defined for all $s \in [0,1]$, since $\lim_{s\to 0^+} 2s\beta_s^{-1}\dot{\beta}_s < \infty$ because β_s is differentiable at $s = 0$ by assumption. The result in (13) is also amenable to an interesting interpretation:

Theorem 3.3. *If $\beta_s/[\sqrt{s}\sigma_s]$ is non-decreasing, then the process $X^{\mathrm{F}} \equiv X^{g^{\mathrm{F}}}$ that solves (9) with $g_s = g_s^{\mathrm{F}}$ is a Föllmer process.*

This theorem is proven in Appendix B.6. To understand its significance, recall that the Föllmer process is the solution to the Schrödinger bridge problem when one of the endpoint measures is a point mass (in this case, at x_0). As such, it offers an entropy-regularized solution to the optimal transport problem. The Föllmer process is usually defined by minimizing its KL divergence with respect to the Wiener process subject to constraints on the endpoints. Theorem 3.3 offers a generalization and new interpretation of this process as the minimizer of the KL divergence of the exact forecasting process from the estimated one, which is more tailored to statistical inference. For more details about Föllmer processes and the Schrödinger bridge problem we refer the reader to Appendix B.5. We also test the performance of (13) in Appendix C.2.

3.5. Implementation

For concreteness, we consider the problem of probabilistic forecasting, but the alternative problem settings covered by our framework can be handled similarly. Given the truncated time series $\mathcal{S}_K = \{x_{k\tau}\}_{k=0}^{K+1}$ with $K \in \mathbb{N}$ and $K' \leq K$, we can approximate the objective in (3) by the empirical loss

$$L_b^K[\hat{b}] = \frac{1}{K'} \sum_{k\in B_{K'}} \int_0^1 |\hat{b}_s(I_s^k, x_{k\tau}) - R_s^k|^2 ds, \tag{14}$$

where $B_{K'} \subset \{0:K\}$ is a subset of indices of cardinality K' and

$$\begin{aligned} I_s^k &= \alpha_s x_{k\tau} + \beta_s x_{(k+1)\tau} + \sqrt{s}\sigma_s z_k \\ R_s^k &= \dot{\alpha}_s x_{k\tau} + \dot{\beta}_s x_{(k+1)\tau} + \sqrt{s}\dot{\sigma}_s z_k \end{aligned} \tag{15}$$

with $z_k \sim \mathsf{N}(0, Id)$, $z_k \perp (x_{k\tau}, x_{(k+1)\tau})$. To arrive at (14) we used that $W_s \overset{d}{=} \sqrt{s}z$ with $z \sim \mathsf{N}(0, Id)$ at all $s \in [0,1]$. In (14) and (16) below, the physical lag $\tau > 0$ is fixed, while s varies over $[0,1]$, and the integral over s can be approximated via an empirical expectation over draws of $s \sim \mathsf{U}([0,1])$. By approximating $\hat{b}$ in an expressive parametric class such as a class of neural networks, we can optimize (14) over the parameters with standard gradient-based methods. This can be performed by batching over subsequences in the available time series, or via online learning if a stream of data is continuously observed.

Having learned an approximation $\hat{b}$, we can construct an approximation of $\hat{b}^g$ using (8), (10), and (11). We may then form our model given the new observation $x_{k\tau}$ by solving

$$d\hat{X}_s^k = \hat{b}_s^g(\hat{X}_s^k, x_{k\tau})ds + g_s dW_s, \quad \hat{X}_{s=0}^k = x_{k\tau}, \tag{16}$$

with various realizations of the noise W_s to generate a set of $\hat{X}_{s=1}^k$ that approximately samples $\rho_c(\cdot|x_{k\tau})$. This generates an ensemble of forecasts with statistics consistent with those of the time-series seen during training. This process can also be iterated autoregressively by setting $\hat{X}_{s=0}^{k+1} = \hat{X}_{s=1}^k$ and by solving the SDE (16) with k replaced by $k+1$ to get an approximate sample $\hat{X}_{s=1}^{k+1}$ of $\rho_c(\cdot|x_{(k+1)\tau})$. This iteration does not require any additional training, since it uses the same $\hat{b}_s$. These procedures are summarized in Algorithms 1 and 2. The first step to get $X_{s=s_1} = \hat{X}_1$ in Algorithm 2 is consistent and designed so that it avoids computing $\hat{b}_{s=0}^g$, since using (8) and (10) can exhibit numerical singularities even though $b_{s=0}^g$ is well defined when the conditions of Theorem 3.2 are met.

Algorithm 1 Training

1: **Input**: Data set $\mathcal{S}_K = \{x_{k\tau}\}_{k=0}^{K+1}$; minibatch size $K' \leq K$; coefficients $\alpha_s, \beta_s, \sigma_s$.
2: **repeat**
3: Compute the empirical loss $L_b^K[\hat{b}]$ in (14).
4: Take gradient step on $L_b^K[\hat{b}]$ to update $\hat{b}_s$.
5: **until** converged
6: **Return**: drift $\hat{b}_s(x, x_0)$.

Algorithm 2 Sampling

1: **Input**: Observation $x_{k\tau}$; model $\hat{b}_s(x, x_0)$; noise coefficient g_s, grid $s_0 = 0 < s_1 \cdots < s_N = 1$ with $N \in \mathbb{N}$; *iid* $\eta_n \sim \mathsf{N}(0, Id)$ for $n = 0 : N-1$.
2: Set $\Delta s_n = s_{n+1} - s_n$, $n = 0 : N-1$.
3: Set $\hat{X}_1 = x_{k\tau} + \hat{b}_{s_0}(x_{k\tau}, x_{k\tau})\Delta s_0 + \sigma_{s_0}\sqrt{\Delta s_0}\eta_0$.
4: **for** $n = 1 : N-1$ **do**
5: Compute $\hat{b}_{s_n}^g(\hat{X}_n, x_{k\tau})$ from (8) and (10).
6: Set $\hat{X}_{n+1} = \hat{X}_n + \hat{b}_{s_n}^g(\hat{X}_n, x_{k\tau})\Delta s_n + g_{s_n}\sqrt{\Delta s_n}\eta_n$.
7: **end for**
8: **Return**: $\hat{X}_{N+1} \sim \hat{\rho}_c(\cdot|x_{k\tau}) \approx \rho_c(\cdot|x_{k\tau})$.

4. Numerical illustrations

In what follows, we test our proposed method in several application domains. For all tests, an interpolant with coefficients $\alpha_s = 1 - s$, $\sigma_s = \varepsilon(1-s)$ for some $\varepsilon > 0$, and

$\beta_s = s^2$ is used. The condition that $\dot{\beta}_0 = 0$ empirically ensures that the norm of the parameter gradients used to train our neural networks are well behaved. We report results with the diffusion coefficient in (16) chosen to be $g_s = \sigma_s$, as we found that the impact of learning $\hat{b}_s$ well by choice of the right interpolant outweighed the effect of varying the SDE for the systems we study. For additional numerical experiments with $\alpha_s = 1 - s, \sigma_s = \varepsilon(1 - s)$ and $\beta_s = s$, and with $g_s = g_s^{\mathrm{F}}$, we refer the reader to Appendix C.2. Investigation of the Föllmer SDE described in Theorem 3.3, in both theory and experiment, will be saved for future work.

4.1. Multi-modal jump diffusion process

Our first example is synthetic, and consists of forecasting a two-dimensional jump-diffusion process with invariant PDF given by a Gaussian mixture (Fig. 1). We study a particle governed by Langevin dynamics that is randomly kicked in the counterclockwise direction, where the times between kicks are specified by a Poisson process (see Appendix C.1 for details).

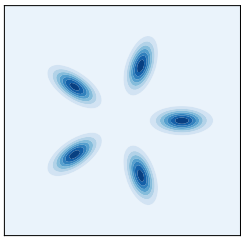

Figure 1. Invariant PDF for the jump-diffusion process.

In this example, if the process starts at a point in one mode, its PDF spreads in the mode and leaks into the other nearby modes in the counterclockwise direction (see Fig. 2). Correspondingly, the conditional PDF of x_τ given x_0 is itself a Gaussian mixture PDF that is sharply peaked around x_0 when τ is small, and which slowly evolves into the 5-mode invariant measure associated with the potential as τ increases. We generate a long time series of this process and use it at different lags τ in the empirical loss (14) to learn the drift velocity $\hat{b}$, which we model as a fully connected neural network. We then use the estimated $\hat{b}^g$ in the SDE (16) to generate probabilistic forecasts. The results (Fig. 2) indicate that the law of these forecasts is in excellent agreement with the true $\rho_c(x_\tau|x_0)$. We can also iterate using the procedure described in Sec. 3.5 to estimate $\rho_c(x_{k\tau}|x_0)$ for $k > 1$ without additional retraining. We find excellent agreement in doing so, including beyond the decorrelation time, when the conditional PDF relaxes into the equilibrium distribution independent of x_0. We note that this is an example in which probabilistic forecasting is key, as regressing x_τ given x_0 would give results with little information. Probabilistic forecasting is needed to capture the non-Gaussian and multimodal nature of the forecasts, which would be hard to capture with deterministic methods.

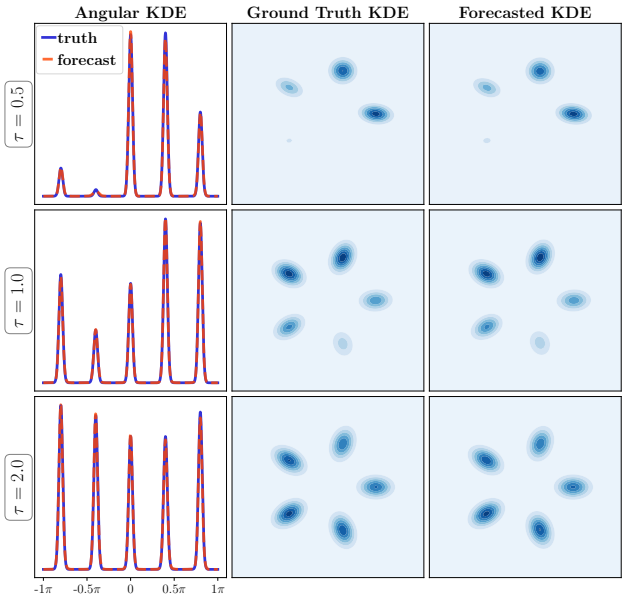

Figure 2. **Forecasting comparison for the jump-diffusion process.** (Left) Comparison of the truth to the forecasted prediction in the angular coordinates. (Middle) Ground truth KDEs at various lag times τ. (Right) Forecasted KDEs at the same lag times τ.

4.2. Forecasting the 2d Navier-Stokes Equations

Our second numerical example considers forecasting the dynamics of the solution to the 2d Navier-Stokes equations with random forcing on the torus $\mathbb{T}^2 = [0, 2\pi]^2$. With the vorticity formulation, the Navier-Stokes equations read

$$d\omega + v \cdot \nabla\omega dt = \nu\Delta\omega dt - \alpha\omega dt + \varepsilon d\eta. \tag{17}$$

Here $v = \nabla^\perp\psi = (-\partial_y\psi, \partial_x\psi)$ is the velocity expressed in terms of the stream function ψ, which is a solution to $-\Delta\psi = \omega$, $d\eta$ is white-in-time random forcing acting on a few Fourier modes, and $\nu, \alpha, \varepsilon > 0$ are parameters (see Appendix C.2 for details). We work in a setting where (17) is provably ergodic with a unique invariant measure (Hairer & Mattingly, 2006). Our objective is to forecast the solution to (17) at time $t + \tau$ given its solution at time t after the process has reached a statistically steady state. We do so using both full- and low-resolution data at time t, but our goal is always to forecast at full resolution.

Vorticity Data. We employ a pseudo-spectral method to simulate (17) and hence to obtain a dataset of snapshots of the vorticity field. We set the timestep $\Delta t = 10^{-4}$ and grid size to 256×256. We store snapshots at regular intervals of $\Delta t = 0.5$. We conduct simulations for 2000 trajectories within the time range of $t \in [0, 100]$; we then exclude the initial phase $t \in [0, 50]$ from our data. Ultimately, we collect a total of 2×10^5 snapshots, which are treated as samples from the invariant measure of (17). To reduce memory

requirements, we downsize the dataset to a resolution of 128×128. In all our experiments, we used a UNet (Ho et al., 2020) as our network for approximating the velocity field. Detailed parameters for the training and dataset generation can be found in the Appendix C.2.

Forecasting at Full Resolution. First, we consider predicting the distribution of the vorticity field that may evolve from a given realization. To this end, we learn the SDE that samples the conditional distribution of vorticity fields after lag $\tau = 0.5$, and we iterate this SDE to get forecasted predictions after lag 2τ, 3τ, etc. In the top row of Fig. 3, the first panel shows a snapshot of a vorticity field, while the next three panels show different samples of vorticity fields generated after lag 2 by iterating our forecasting procedure. Note that these generated vorticity fields are different from one another, emphasizing the need for probabilistic forecasting. This is corroborated by the true and forecasted conditional means of the field shown in the first two panels on the bottom row of Fig. 3. While correctly captured by our approach, this conditional mean is clearly not informative on its own, as it averages over the spatial features present in the actual forecast. The spread of this ensemble of forecasts is also apparent from the standard deviation of the field, shown in the third and fourth panel on the bottom row of Fig. 3. Also shown in the right panel is the enstrophy spectrum of the true vorticity field and the ensemble of forecasts (see Appendix C.2), showing that our method captures this important physical quantity correctly.

Forecasting from Low-Resolution Data. We now consider forecasting the vorticity field at a resolution of 128×128 from a downsized version at resolution 32×32 by learning the SDE as before with a drift velocity that is conditioned on the low-resolution field used as x_0. The outcome of this task is shown in Fig. 4, where the four panels on the right show the low-resolution field used as input surrounded by three full-resolution forecasts generated after lag $= 1$. We also plot the enstrophy spectrum of the low-resolution field (which does not go past wavenumber 16) and the spectra of the true and the forecasted field at full resolution in the left panel. As can be seen, our approach recovers the true spectrum very accurately.

We present additional experiments such as superresolution, comparisons between using σ_s and g_s^{F} (Föllmer process) in terms of KL accuracy, and comparisons to flow matching and deterministic forecasting in Appendix C.2. In addition, in Appendix C.2, we present results showing that the SDE forecasting can be $100\times$ faster than directly simulating the stochastic PDE in the Navier-Stokes example.

4.3. Video Forecasting

We model the KTH and CLEVRER datasets. We follow RIVER (Davtyan et al., 2023) and model these videos in the latent space of a VQGAN (Esser et al., 2021) trained to auto-encode the datasets, as is common for high-resolution image and video synthesis (Vahdat et al., 2021; Rombach et al., 2022; Peebles & Xie, 2023; Blattmann et al., 2023; Davtyan et al., 2023; Ma et al., 2024).

Task Description. For video frames x^t with C channels and resolution $H \times W$, the VQGAN maps each video frame to a latent image $y^t = \mathrm{Encode}(x^t)$ with C_ℓ latent channels and latent resolution $H_\ell \times W_\ell$. Our aim is to sample from the conditional density $\rho_c(y^t | y^{t-1}, \ldots, y^{t-C})$ describing the probability of a frame given a sequence of previous frames, which can then be decoded $x^t = \mathrm{Decode}(y^t)$. We set the interpolant base distribution to a point mass on $y_{s=0} = y^{t-1}$ and we model $y_{s=1} = y^t$.

Generation. Because conditioning on the whole set of C previous time slices is costly, we follow RIVER (Davtyan et al., 2023) and use a Monte-Carlo estimator that generates our estimate of the tth latent frame $\hat{y}^t$ conditional on latent frame y^{t-1} and an additional frame y^{t-j} randomly chosen for $1 \geq j < t-1$. To give the network context for the conditioned frame, we also condition on the time index $t-j$. This random conditioning set $(y^{t-1}, y^{t-j}, t-j)$ avoids the need to compute functions of the entire conditioning context. Samples are then decoded to produce images $x^t = \mathrm{Decode}(y^t)$. To sample a full video, we apply this forecasting strategy autoregressively. For further details, see Algorithm 3 in Appendix C.3.

FVD metric. The Fréchet Video Distance (FVD) (Unterthiner et al., 2018) extends the Fréchet Inception Distance (FID) (Heusel et al., 2017). We select 256 test set videos and generate 100 completions for each one, thereby comparing 256 real videos to 25,600 generated videos. We also report several qualitative features of the generated videos. One consequence of using the VQGAN is that the performance is bounded by the FVD of the decoded encoded data since the generative model targets the encoded distribution. More information on evaluation is provided at https://github.com/interpolants/forecasting.

Baseline. We compare with the setup from Davtyan et al. (2023), which learns a deterministic flow using flow matching (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022) to map a Gaussian sample to the next video frame, conditioned on the same information as in our setup. By contrast, we generate with an SDE sampler initialized at the previous video frame, which is more proximal to the next target frame than pure noise. We use the VQGAN checkpoints from RIVER so that we can study our proposed generative modeling method in a controlled context.

Datasets. The KTH dataset (Schuldt et al., 2004) consists of black-and-white videos of 25 people completing one of 6 actions such as jogging and hand-waving. We use the last

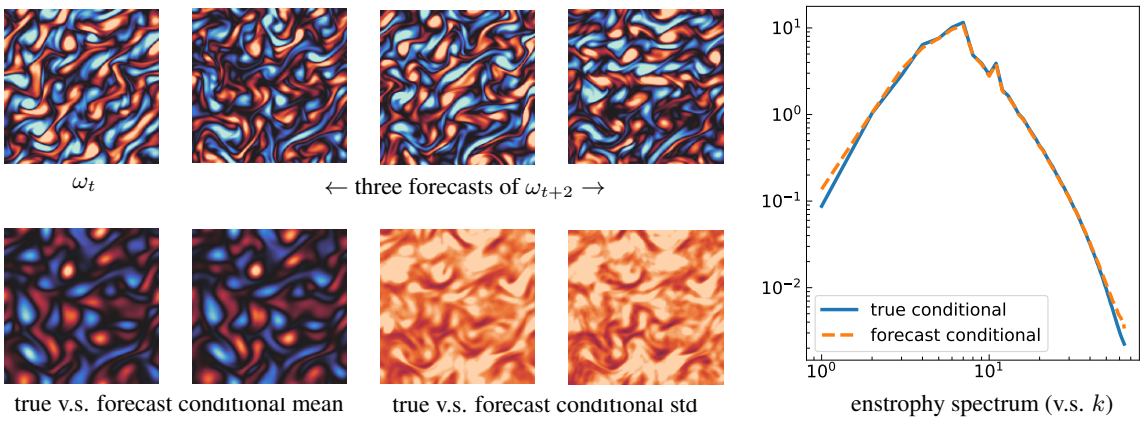

Figure 3. **Temporal forecasting on stochastically-forced Navier Stokes.** (Top left) Different forecasts from our method at lag $\tau = 2$, for a fixed ω_t. (Bottom left) Comparisons between the forecast sample mean and standard deviation for this ω_t against the truths. (Right) Enstrophy spectrum of the true and forecasted conditional distribution. Note: All the NS figures in this paper share the same color bars for the vorticity field (on a scale from -5 to 5) and std (on a scale from 0 to 3) respectively.

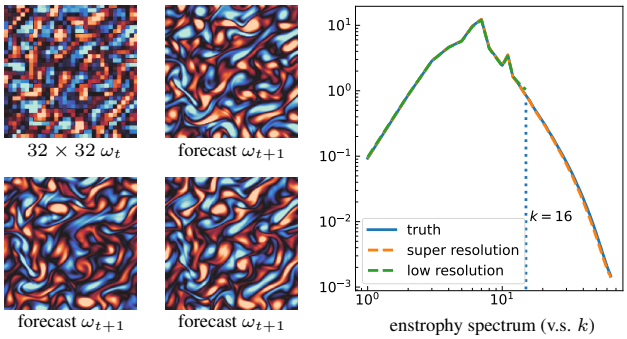

Figure 4. **Spatiotemporal forecasting on stochastically-forced Navier Stokes.** (Left) Low resolution ω_t and three of our forecasted samples at $t + 1$. (Right) Enstrophy spectrums of the low resolution ω_t, the superresolution forecast of ω_{t+1}, and the true ω_{t+1}.

5 people as the test set. The $1 \times 64 \times 64$-dimensional data space is mapped to a $4 \times 8 \times 8$-dimensional latent space by the VQGAN. During generation, we start with 10 given video frames, and we generate the next 30 frames. The CLEVRER dataset[5] (Yi et al., 2019) contains videos created for studying reasoning and physics tasks. The videos feature cubes, spheres, and other shapes traveling across the screen and interacting through collisions while being subject to forces and rotations. One interpretation of this data is that a generative model needs to deduce physical phenomena to succeed at generation. For example, objects should not go through each other, and instead should bounce off one another. The $3 \times 128 \times 128$-dimensional data space is mapped to a $4 \times 16 \times 16$ latent space. During generation, we condition on just 2 real frames and generate 14.

[5] http://clevrer.csail.mit.edu/

	KTH		*CLEVRER*	
Method	100k	250k	100k	250k
RIVER	46.69	41.88	60.40	48.96
PFI (ours)	**44.38**	**39.13**	**54.7**	**39.31**
Auto-enc.	33.45	33.45	2.79	2.79
Shifted FVD				
RIVER	13.24	8.43	57.61	46.17
PFI (ours)	**10.93**	**5.68**	**51.91**	**36.52**

Table 1. **Video Results.** FVD computed on 256 test set videos, with the model generating 100 completions for each video. Results are reported for 100k and for 250k gradient steps. "Auto-enc." represents the FVD of the pretrained encoder-decoder compared to the real data. It serves as a bound on the possible model performance, because the modeling is done in the latent space computed by the encoder-decoder pair. We also supply a shifted FVD, where encoded-decoded FVD is subtracted from FVD values to show they approach this approximate bound.

Results. Training details are in Appendix C.3. Table 1 shows the FVD performance of our model, probabilistic forecasting with interpolants (PFI), as compared to the RIVER baseline. We train both models under the same conditions for a controlled comparison. On both the KTH and CLEVRER datasets, PFI surpasses the standard flow matching approach. In addition to the numerical comparison, we demonstrate that our models produce diverse forecasts that capture physical rules inherent to the videos. In Figure 5, we illustrate two trajectories of the animation. For each, we supply the initial condition, the dataset trajectory, and a generated trajectory based off of the same initial condition to show that the continuation of frames is probabilistic.

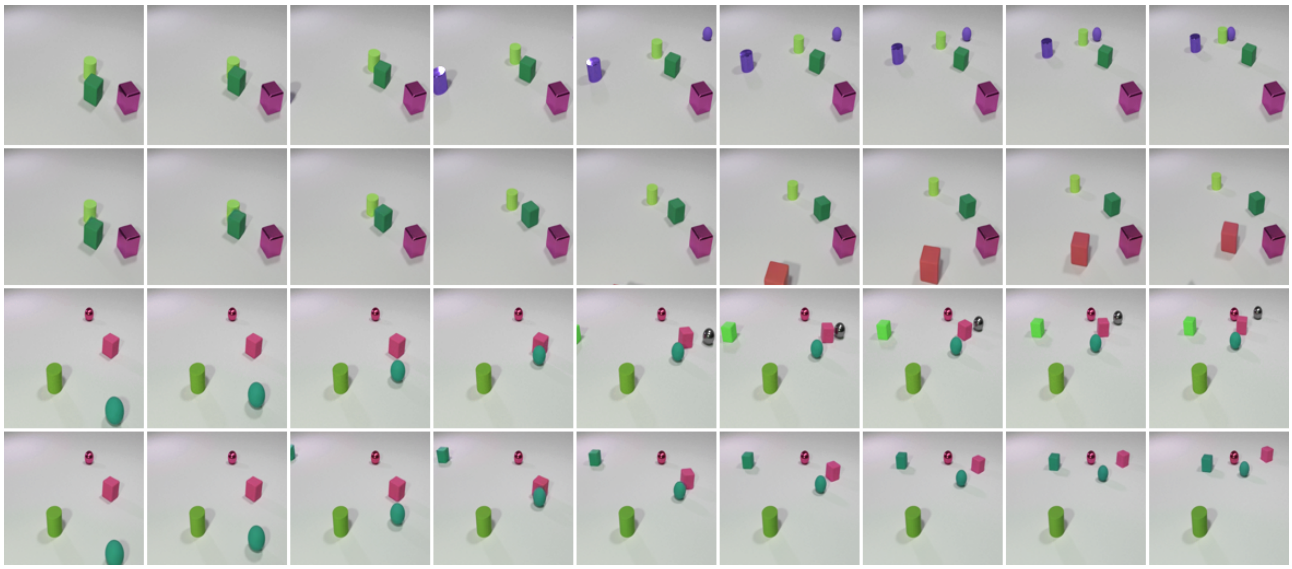

Figure 5. **Video generation on the CLEVRER dataset.** (Top row) Real trajectory. (Second row) Generated trajectory. A new, red cube enters the scene. (Third row) Real trajectory. (Fourth row) Generated trajectory. A new green cube enters the scene, and collision physics is respected (green ball hits red cube).

Figure 6. **Video generation on the KTH dataset.** (Top row) Real trajectory. (Middle and bottom rows) Generated trajectories. As time proceeds, the generated videos drift from the true video, but still display temporally-consistent hand-waving motions.

For example, there is variation in the dataset trajectory of a green cube colliding with a green cylinder as compared to the forecasted trajectory, while also preserving the animated physics. In Figure 6, we show that, from the same initial frame, the forecasts give varied realizations of the hand-waving video category.

5. Conclusion and Future Work

In this work, we introduced a principled approach to the use of generative modeling for probabilistic forecasting. By introducing stochastic processes that transport a point mass centered at a current observation of the system to a distribution over future states of that system, our proposed method uses dynamical measure transport in a way that naturally aligns with the framework of probabilistic forecasting. It also allows us to minimize the impact of the estimation error by tuning of the diffusion coefficient, which can be done after training and offers a new perspective on the Föllmer process. We have shown various uses of this approach, ranging from predicting the evolution of stochastic fluid dynamics to video completion tasks. Future work will consider using these models for empirical weather data and incorporation of physical structure into the generative model. Other generalizations will include mitigating the cost of simulating the SDE e.g. by tuning the diffusion coefficient to that effect, or using data sampled at random time-lags τ by conditioning the drift on these τ.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. We note, however, that video generation, which is one of the possible applications of the forecasting framework, is a less explored domain that could promote harm through biases in the model. We surmise that

it carries the same risks as image generation technologies.

Acknowledgments

We thank Joan Bruna, Jon Niles-Weed, Loucas Pillaud-Vivien, and Valentin De Bortoli for useful discussions. YC, MSA and NMB are funded by the ONR project entitled Mathematical Foundation and Scientific Applications of Machine Learning. MH is supported on the Meta Platforms, Inc project entitled Probabilistic Deep Learning with Dynamical Systems. EVE is supported by the National Science Foundation under Awards DMR-1420073, DMS-2012510, and DMS-2134216, by the Simons Collaboration on Wave Turbulence, Grant No. 617006, and by a Vannevar Bush Faculty Fellowship.

References

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Alexander, R. and Giannakis, D. Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques. *Physica D: Nonlinear Phenomena*, 409:132520, 2020.

Berry, T., Giannakis, D., and Harlim, J. Nonparametric forecasting of low-dimensional dynamical systems. *Physical Review E*, 91(3):032915, 2015.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Buaria, D. and Sreenivasan, K. R. Forecasting small-scale dynamics of fluid turbulence using deep neural networks. *Proceedings of the National Academy of Sciences*, 120 (30):e2305765120, 2023.

Chen, R. T., Amos, B., and Nickel, M. Neural spatio-temporal point processes. *arXiv preprint arXiv:2011.04583*, 2020.

Chen, Y., Georgiou, T. T., and Pavon, M. Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrodinger bridge. *Siam Review*, 63(2):249–313, 2021.

Davtyan, A., Sameni, S., and Favaro, P. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23263–23274, 2023.

de Bézenac, E., Rangapuram, S. S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P., and Januschowski, T. Normalizing kalman filters for multivariate time series analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2995–3007, 2020.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709, 2021.

Dellnitz, M. and Junge, O. On the approximation of complicated dynamical behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515, 1999.

Dresdner, G., Kochkov, D., Norgaard, P., Zepeda-Núñez, L., Smith, J. A., Brenner, M. P., and Hoyer, S. Learning to correct spectral methods for simulating turbulent flows. *arXiv preprint arXiv:2207.00556*, 2022.

Eldan, R. and Lee, J. R. Regularization under diffusion and anticoncentration of the information content. *Duke Mathematical Journal*, 167(5):969–993, 2018.

Eldan, R., Lehec, J., and Shenfeld, Y. Stability of the logarithmic sobolev inequality via the föllmer process. In *Annales de l'Institut Henri Poincaré-Probabilités et Statistiques*, volume 56, pp. 2253–2269, 2020.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, volume 29, 2016.

Föllmer, H. Time reversal on wiener space. *Stochastic Processes—Mathematics and Physics*, pp. 119–129, 1986.

Giannakis, D., Henriksen, A., Tropp, J. A., and Ward, R. Learning to forecast dynamical systems from streaming data. *SIAM Journal on Applied Dynamical Systems*, 22 (2):527–558, 2023.

Gneiting, T. and Katzfuss, M. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

Gneiting, T., Raftery, A. E., Westveld, A. H., and Goldman, T. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Hairer, M. and Mattingly, J. C. Ergodicity of the 2d navier-stokes equations with degenerate stochastic forcing. *Annals of Mathematics*, pp. 993–1032, 2006.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, volume 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, volume 33, pp. 6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022b.

Huang, J., Jiao, Y., Kang, L., Liao, X., Liu, J., and Liu, Y. Schrödinger-Föllmer sampler: sampling without ergodicity. *arXiv preprint arXiv:2106.10880*, 2021.

Jiang, R., Lu, P. Y., Orlova, E., and Willett, R. Training neural operators to preserve invariant measures of chaotic attractors. *arXiv preprint arXiv:2306.01187*, 2023.

Jiao, Y., Kang, L., Liu, Y., and Zhou, Y. Convergence analysis of Schrödinger-Föllmer sampler without convexity. *arXiv preprint arXiv:2107.04766*, 2021.

Kaiser, E., Kutz, J. N., and Brunton, S. L. Data-driven discovery of koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(3):035023, 2021.

Kidger, P., Foster, J., Li, X., and Lyons, T. J. Neural sdes as infinite-dimensional gans. In *International conference on machine learning*, pp. 5453–5463. PMLR, 2021.

Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., and Levine, S. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Lehec, J. Representation formula for the entropy and functional inequalities. In *Annales de l'IHP Probabilités et statistiques*, volume 49, pp. 885–899, 2013.

Léonard, C. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems-Series A*, 34(4):1533–1574, 2014.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Li, Z., Liu-Schiaffini, M., Kovachki, N., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Learning dissipative dynamics in chaotic systems. *arXiv preprint arXiv:2106.06898*, 2021.

Lienen, M., Lüdke, D., Hansen-Palmus, J., and Günnemann, S. From zero to turbulence: Generative modeling for 3d flow simulation. In *The Twelfth International Conference on Learning Representations*, 2023.

Lim, B. and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. I^2SB: Image-to-image Schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

Masini, R. P., Medeiros, M. C., and Mendes, E. F. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826, 2020.

Palmer, T., Molteni, F., Mureau, R., Buizza, R., Chapelet, P., and Tribbia, J. Ensemble prediction. In *Proc. ECMWF Seminar on Validation of models over Europe*, volume 1, pp. 21–66, 1993.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Peluchetti, S. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rühling Cachay, S., Zhao, B., Joren, H., and Yu, R. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Schrödinger, E. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 3, pp. 269–310, 1932.

Schuldt, C., Laptev, I., and Caputo, B. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pp. 32–36. IEEE, 2004.

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. Diffusion schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Smagorinsky, J. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, volume 28, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Tzen, B. and Raginsky, M. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pp. 3084–3114. PMLR, 2019.

Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. In *Advances in neural information processing systems*, volume 34, pp. 11287–11302, 2021.

Vargas, F., Ovsianas, A., Fernandes, D., Girolami, M., Lawrence, N. D., and Nüsken, N. Bayesian learning via neural schrödinger–föllmer flows. *Statistics and Computing*, 33(1):3, 2023.

Wang, G., Jiao, Y., Xu, Q., Wang, Y., and Yang, C. Deep generative learning via schrödinger bridge. In *International Conference on Machine Learning*, pp. 10794–10804. PMLR, 2021.

Wanner, M. and Mezic, I. Robust approximation of the stochastic koopman operator. *SIAM Journal on Applied Dynamical Systems*, 21(3):1930–1951, 2022.

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

Zhang, Q. and Chen, Y. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2021.

Zhao, M. and Jiang, L. Data-driven probability density forecast for stochastic dynamical systems. *Journal of Computational Physics*, 492:112422, 2023.

A. Discussion on the stationarity assumption

The stationarity assumption of the time series in Section 3 implies that $\rho(x_{k\tau}, x_{(k+1)\tau})$ is independent of time. This assumption is sufficient, but not necessary for the use of our algorithm. More specifically, we can decompose the possible scenarios into 4 categories:

	Markovian	Non-Markovian
Stationary	Works as currently presented	Works by the stationarity assumption
Non-stationary	Works; learn a time-independent transition kernel	Works; need to condition on time in the model

In more details:

- If the time series is stationary and Markovian, then the method works as it is currently presented.
- If the time series is non-stationary but Markovian, our algorithm can still accurately learn the conditional distribution $\rho_c(\cdot|x_0)$, provided that the transition kernel is time-independent. This can be seen from our formulation, which only requires a joint distribution $\rho(x_0, x_1)$ to be time-independent. We have the flexibility to use any distribution for the input x_0 and use it with the conditional distribution $\rho_c(\cdot|x_0)$ to define the joint distribution.
- If the time series is stationary but non-Markovian, then the method works again by the stationarity assumption.
- If the time series is non-stationary and non-Markovian, then the method can still work if it is adapted so that the drift also conditions on the physical time.

The Navier-Stokes experiments we present are in the category 'stationary and Markovian' since to obtain the training data, we ran the simulation for a sufficiently long time and discarded the initial samples. The video experiments seem to belong to the category 'non-stationary but Markovian', though this assertion is hard to test.

B. Details on stochastic interpolants

B.1. Analytical formulas for some specific $\alpha_s, \beta_s, \sigma_s$

In this section, we present the formulas for some specific choices of α_s, β_s, and σ_s, the corresponding optimal (Föllmer) drift g_s^{F}, and the expression for b_s^g in terms of b_s.

For $\alpha_s = 1 - s, \sigma_s = \varepsilon(1-s)$ where $\varepsilon > 0$ is a tunable parameter, and $\beta_s = s$, we have

$$\begin{aligned} b_s(x, x_0) &= \mathbb{E}^{x_0}[x_1 - x_0 - \varepsilon\sqrt{s}z|(1-s)x_0 + sx_1 + \varepsilon(1-s)\sqrt{s}z = x], \qquad \text{for } s \in (0,1) \\ b_0(x_0, x_0) &= \mathbb{E}^{x_0}[x_1] - x_0, \\ b_1(x, x_0) &= x - x_0. \end{aligned} \tag{18}$$

Moreover, b_s is the minimizer of the loss (with respect to $\hat{b}$)

$$L_b[\hat{b}] = \int_0^1 \mathbb{E}\big[|\hat{b}_s((1-s)x_0 + sx_1 + (1-s)\varepsilon\sqrt{s}z, x_0) - (x_1 - x_0 - \varepsilon\sqrt{s}z)|^2\big]\, ds. \tag{19}$$

We also have

$$A_s = \frac{1}{\varepsilon^2 s(1-s)}, \qquad c_s(x, x_0) = x - x_0 \tag{20}$$

so that, for any g_s such that $\lim_{s\to 0^+} s^{-1}[g_s^2 - \varepsilon^2(1-s)^2]$ and $\lim_{s\to 1^-} g_s^2/(\varepsilon(1-s))$ exists,

$$\begin{aligned} b_s^g(x, x_0) &= b_s(x, x_0) + \tfrac{1}{2}\big(g_s^2 - \varepsilon^2(1-s)^2\big)\left(\frac{b_s(x, x_0)}{\varepsilon^2(1-s)} - \frac{x - x_0}{\varepsilon^2 s(1-s)}\right) \qquad \text{for } s \in (0,1), \\ b_0^g(x_0, x_0) &= b_0(x_0, x_0) = \mathbb{E}^{x_0}[x_1] - x_0, \\ b_1^g(x, x_0) &= b_1(x, x_0) = x - x_0. \end{aligned} \tag{21}$$

In addition, we have

$$g_s^{\mathrm{F}} = \varepsilon\sqrt{(1-s)(1+s)}, \tag{22}$$

so that

$$b_s^{\mathrm{F}}(x,x_0) = (1+s)b_s(x,x_0) - x + x_0 \qquad \text{for } s \in [0,1]. \tag{23}$$

For $\alpha_s = 1-s, \sigma_s = \varepsilon(1-s)$, and $\beta_s = s^2$, we have

$$\begin{aligned} b_s(x,x_0) &= \mathbb{E}^{x_0}[2sx_1 - x_0 - \varepsilon\sqrt{s}z | (1-s)x_0 + s^2x_1 + \varepsilon(1-s)\sqrt{s}z = x], \quad \text{for } s \in (0,1) \\ b_0(x_0,x_0) &= -x_0, \\ b_1(x,x_0) &= x - x_0. \end{aligned} \tag{24}$$

Moreover, b_s is the minimizer of the loss

$$L_b[\hat{b}] = \int_0^1 \mathbb{E}\big[|\hat{b}_s((1-s)x_0 + s^2x_1 + \varepsilon(1-s)\sqrt{s}z, x_0) - (2sx_1 - x_0 - \varepsilon\sqrt{s}z)|^2\big]ds. \tag{25}$$

We also have

$$A_s = \frac{1}{\varepsilon^2 s^2(1-s)(2-s)}, \qquad c_s(x,x_0) = 2sx - s(2-s)x_0, \tag{26}$$

so that, for any g_s such that $\lim_{s\to 0^+} s^{-1}[g_s^2 - \varepsilon^2(1-s)^2]$ and $\lim_{s\to 1^-} g_s^2/(\varepsilon(1-s))$ exist,

$$\begin{aligned} b_s^g(x,x_0) &= b_s(x,x_0) + \tfrac{1}{2}\big(g_s^2 - \varepsilon^2(1-s)^2\big)\left(\frac{b_s(x,x_0)}{\varepsilon^2(1-s)(2-s)} - \frac{x-x_0}{\varepsilon^2 s^2(1-s)(2-s)}\right) \quad \text{for } s \in (0,1) \\ b_0^g(x_0,x_0) &= b_0(x_0,x_0) = \mathbb{E}^{x_0}[x_1] - x_0 \\ b_1^g(x,x_0) &= b_1(x,x_0) = x - x_0. \end{aligned} \tag{27}$$

In addition we have

$$g_s^{\mathrm{F}} = \varepsilon\sqrt{(1-s)(3-s)} \tag{28}$$

so that

$$\begin{aligned} b_s^{\mathrm{F}}(x,x_0) &= \Big(1 + \frac{1}{2-s}\Big)b_s(x,x_0) - \frac{1}{s(2-s)}(2x - (2-s)x_0) \qquad \text{for } s \in (0,1], \\ b_0^{\mathrm{F}}(x_0,x_0) &= -2x_0. \end{aligned} \tag{29}$$

We summarize the above calculations in the following table:

α_s	β_s	σ_s	g_s^{F}	A_s	$c_s(x,x_0)$	$b_s^{\mathrm{F}}(x,x_0)$
$1-s$	s	$\varepsilon(1-s)$	$\varepsilon\sqrt{(1-s)(1+s)}$	$\frac{1}{\varepsilon^2 s(1-s)}$	$x - x_0$	$(1+s)b_s - x + x_0$
$1-s$	s^2	$\varepsilon(1-s)$	$\varepsilon\sqrt{(3-s)(1-s)}$	$\frac{1}{\varepsilon^2 s^2(1-s)(2-s)}$	$2sx - s(2-s)x_0$	$(1+\frac{1}{2-s})b_s - \frac{2x-(2-s)x_0}{s(2-s)}$

B.2. Proof of Theorem 3.1

Recall that:

Definition B.1. The stochastic interpolant I_s is the stochastic process defined as

$$I_s = \alpha_s x_0 + \beta_s x_1 + \sigma_s W_s \qquad s \in [0,1], \tag{30}$$

where

- $\alpha, \beta, \sigma \in C^1([0,1])$ satisfy $\alpha_s^2 + \beta_s^2 + \sigma_s^2 > 0$ for all $s \in [0,1]$, $\dot{\beta}_s > 0$ for all $s \in (0,1]$, and $\dot{\sigma}_s < 0$ for all $s \in [0,1]$, as well as the boundary conditions $\alpha_0 = \beta_1 = 1$, $\alpha_1 = \beta_0 = \sigma_1 = 0$.
- The pair (x_0, x_1) are jointly drawn from a PDF $\rho(x_0,x_1)$ such that $\mathbb{E}_{(x_0,x_1)\sim\rho}\big[|x_0|^2 + |x_1|^2\big] < \infty$.

- $W = (W_s)_{s\in[0,1]}$ is a standard Wiener process with $W \perp (x_0, x_1)$.

In view of this definition, let us give a more precise formulation of Theorem 3.1:

Theorem B.2. *Let I_s be the stochastic interpolant introduced in Definition B.1 and let*

$$\begin{aligned} &\forall s \in [0,1]: && x_s = \alpha_s x_0 + \beta_s x_1 + \sigma_s \sqrt{s} z \\ &\forall (s, x, x_0) \in (0,1] \times \mathbb{R}^d \times \mathbb{R}^d: && b_s(x, x_0) = \mathbb{E}^{x_0}[\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s \sqrt{s}\, z | x_s = x] \end{aligned} \tag{31}$$

where $\mathbb{E}^{x_0}[\cdot|x_s = x]$ denotes an expectation over $x_1 \sim \rho_c(\cdot|x_0)$ and $z \sim \mathsf{N}(0, Id)$ conditional on $x_s = x$. Moreover, set $b_{s=0}(x_0, x_0) := \dot{\alpha}_0 x_0 + \dot{\beta}_0 \mathbb{E}^{x_0}[x_1]$. Then the solutions to the SDE

$$dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s, \qquad X_{s=0} = x_0, \tag{32}$$

are such that Law(X_s) = Law$(I_s|x_0)$ at all $(s, x_0) \in [0,1] \times \mathbb{R}^d$. In particular $X_{s=1} \sim \rho_c(\cdot|x_0)$. In addition the drift b is the unique minizer over all $\hat{b}$ of the objective function

$$L_b[\hat{b}] = \int_0^1 \mathbb{E}\big[|\hat{b}_s(x_s, x_0) - (\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s \sqrt{s}\, z)|^2\big] ds, \tag{33}$$

where $\mathbb{E}$ denotes an expectation over $(x_0, x_1) \sim \rho(x_0, x_1)$ and $z \sim \mathsf{N}(0, Id)$ with $(x_0, x_1) \perp z$.

Note that the objective (33) is the same as (3) because $x_s \stackrel{d}{=} I_s$ and $\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s \sqrt{s}\, z \stackrel{d}{=} R_s$ at all $s \in [0,1]$. Note also that, since $\alpha, \beta, \sigma \in C^1([0,1])$, the factors $\dot{\alpha}_s, \dot{\beta}_s, \dot{\sigma}_s \sqrt{s}$ in the loss (33) are all bounded.

Proof of Theorem B.2. Notice that the process I_s defined in (30) has the same law at any $s \in [0,1]$ as x_s. If we denote by $\mu(s, dx|x_0)$ the measure of $I_s|x_0$ or $x_s|x_0$, and by $\phi : \mathbb{R}^d \to \mathbb{R}$ a twice-differentiable test function with compact support, by definition we have

$$\forall (s, x_0) \in [0,1] \times \mathbb{R}^d \quad : \quad \int_{\mathbb{R}^d} \phi(x)\mu(s, dx|x_0) = \mathbb{E}[\phi(x_s)|x_0] = \mathbb{E}[\phi(I_s)|x_0], \tag{34}$$

where the first conditional expectation is over $(x_0, x_1) \sim \rho$ and $z \sim \mathsf{N}(0, Id)$ and the second over $(x_0, x_1) \sim \rho$ and W. By the Itô formula we have

$$d\phi(I_s) = (\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s W_s) \cdot \nabla\phi(I_s)ds + \tfrac{1}{2}\sigma_s^2 \Delta\phi(I_s)ds + \nabla\phi(I_s) \cdot dW_s. \tag{35}$$

Integrating this equation in time over $[0, s]$, taking the expectation conditional on x_0, and using both $W_s \perp (x_0, x_1)$ and the Itô isometry, we deduce that

$$\mathbb{E}[\phi(I_s)|x_0] = \phi(x_0) + \int_0^s \Big(\mathbb{E}\big[(\dot{\alpha}_r x_0 + \dot{\beta}_r x_1 + \dot{\sigma}_r W_r) \cdot \nabla\phi(I_r)\big|x_0\big] + \tfrac{1}{2}\sigma_r^2 \mathbb{E}[\Delta\phi(I_r)|x_0]\Big)\, dr. \tag{36}$$

Inserting (36) into (34), we deduce that, $\forall (s, x_0) \in [0,1] \times \mathbb{R}^d$,

$$\begin{aligned} \int_{\mathbb{R}^d} \phi(x)\mu(s, dx|x_0) = \phi(x_0) + \int_0^s \Big(&\mathbb{E}\big[(\dot{\alpha}_r x_0 + \dot{\beta}_r x_1 + \dot{\sigma}_r \sqrt{r}\, z) \cdot \nabla\phi(x_r)\big|x_0\big] \\ &+ \tfrac{1}{2}\sigma_r^2 \mathbb{E}[\Delta\phi(x_r)|x_0]\Big)\, dr, \end{aligned} \tag{37}$$

where we used the fact that x_s and I_s share the same law at each s. Also, W_s and $\sqrt{s}z$ share the same law at each s.

Using the tower property of the conditional expectation, (37) can also be written as

$$\begin{aligned}
&\int_{\mathbb{R}^d} \phi(x)\mu(s,dx|x_0) \\
&= \phi(x_0) + \int_0^s \int_{\mathbb{R}^d} \mathbb{E}\big[(\dot\alpha_r x_0 + \dot\beta_r x_1 + \dot\sigma_r\sqrt{r}\, z)\cdot\nabla\phi(x_r)\big|x_r = x, x_0\big]\mu(r,dx|x_0)dr \\
&\qquad + \tfrac{1}{2}\int_0^s \sigma_r^2 \int_{\mathbb{R}^d} \Delta\phi(x)\mu(r,dx|x_0)dr \\
&= \phi(x_0) + \int_0^s \int_{\mathbb{R}^d} \mathbb{E}\big[(\dot\alpha_r x_0 + \dot\beta_r x_1 + \dot\sigma_r\sqrt{r}\, z)\big|x_r = x, x_0\big]\cdot\nabla\phi(x)\mu(r,dx|x_0)dr \\
&\qquad + \tfrac{1}{2}\int_0^s \sigma_r^2 \int_{\mathbb{R}^d} \Delta\phi(x)\mu(r,dx|x_0)dr \\
&= \phi(x_0) + \int_0^s \int_{\mathbb{R}^d} \big(b_r(x,x_0)\cdot\nabla\phi(x) + \tfrac{1}{2}\sigma_r^2\Delta\phi(x)\big)\,\mu(r,dx|x_0)
\end{aligned} \tag{38}$$

where we used the definition of b in (31) to get the last equality. If we now repeat the same steps to derive an evolution equation for $\mathbb{E}[\phi(X_s)]$ where X_s solves the SDE (5) we arrive at the same equation (38) for the measure of this process, indicating that this measure is also $\mu(s,dx|x_0)$.

It remains to show that the drift $b_s(x,x_0) = \mathbb{E}[\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|x_s = x, x_0]$ is the unique minimizer of the objective function (3). To this end, notice that b is the unique minimizer over all $\hat b$ of

$$\begin{aligned}
&\int_0^1 \mathbb{E}\big[|\hat b_s(x_s,x_0) - b_s(x_s,x_0)|^2 ds, \\
&= \int_0^1 \Big(\mathbb{E}\big[|\hat b_s(x_s,x_0) - (\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z)|^2 + \mathrm{var}^{x_0}[\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|x_s]\big]\Big)\, ds,
\end{aligned} \tag{39}$$

where $\mathbb{E}$ denotes an expectation over $(x_0,x_1)\sim\rho(x_0,x_1)$ and $z\sim\mathsf{N}(0,Id)$ with $(x_0,x_1)\perp z$ and where

$$\begin{aligned}
\mathrm{var}^{x_0}[\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|x_s] &= \mathbb{E}^{x_0}\big[|\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|^2|x_s\big] - \big|\mathbb{E}^{x_0}[\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|x_s]\big|^2, \\
&= \mathbb{E}^{x_0}\big[|\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|^2|x_s\big] - |b_s(x_s,x_0)|^2,
\end{aligned} \tag{40}$$

and where we used the tower property of the conditional expectation $\mathbb{E}[\mathbb{E}[|\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|^2|x_s]] = \mathbb{E}[|\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|^2|x_s]$. Since $\mathrm{var}^{x_0}[\dot\alpha_s x_0 + \dot\beta_s x_1 + \dot\sigma_s\sqrt{s}\, z|x_s = x]$ is independent of $\hat b$, we can drop this term to arrive at the objective (33). □

B.3. Regularity of the drift at $s = 0$

In this section, we discuss the regularity of the drift $b_s(x,x_0)$.

Assumption B.3. The conditional distribution $\rho_c(\cdot|x_0)$ is exponential tailed. More precisely, there exist constants $C_1, C_2 > 0$ (which can depend on x_0), such that

$$\rho_c(x|x_0) \le C_2\exp(-C_1|x|)\,,$$

for any $x\in\mathbb{R}^d$.

The aforementioned assumption is needed for technical reasons, and is used to ensure the validity of a step that involves the interchange of limits and integrations in the proof of Theorem B.4.

By linearity of x_s in x_0, x_1, and z, we can also establish a few properties of the velocity b in (31) which we state as:

Theorem B.4. *Under Assumption B.3, if $\dot\beta_{s=0} = 0$, then the velocity field $b_s(x,x_0)$ can be decomposed as*

$$\forall(s,x,x_0)\in[0,1]\times\mathbb{R}^d\times\mathbb{R}^d \quad b_s(x,x_0) = \dot\alpha_s x_0 + \dot\beta_s\eta_1(s,x,x_0) + \dot\sigma_s\eta_z(s,x,x_0), \tag{41}$$

where

$$\forall (s,x,x_0) \in (0,1] \times \mathbb{R}^d \times \mathbb{R}^d : \quad \begin{cases} \eta_1(s,x,x_0) = \mathbb{E}^{x_0}[x_1|x_s = x], \\ \eta_z(s,x,x_0) = \sqrt{s}\,\mathbb{E}^{x_0}[z|x_s = x], \end{cases}$$
$$\forall (x,x_0) \in \mathbb{R}^d \times \mathbb{R}^d : \quad \begin{cases} \eta_1(0,x,x_0) = \lim_{s\to 0} \eta_1(s,x,x_0) = \mathbb{E}^{x_0}[x_1], \\ \eta_z(0,x,x_0) = \lim_{s\to 0} \eta_z(s,x,x_0) = \dfrac{x - x_0}{\sigma_0}. \end{cases} \tag{42}$$

In addition, these two functions satisfy the constraint

$$\forall (s,x,x_0) \in [0,1] \times \mathbb{R}^d \times \mathbb{R}^d \quad : \quad x = \alpha_s x_0 + \beta_s \eta_1(s,x,x_0) + \sigma_s \eta_z(s,x,x_0). \tag{43}$$

We have $\lim_{s\to 0} b_s(x,x_0) = \dot{\alpha}_0 x_0 + \frac{\dot{\sigma}_0}{\sigma_0}(x - x_0)$ *and* $\lim_{s\to 0} \nabla_x b_s(x,x_0) = \frac{\dot{\sigma}_0}{\sigma_0} Id$ *for any* $x, x_0 \in \mathbb{R}^d$.

Remark B.5. Note that (43) implies that we can get η_1 from η_z at any time such that $\beta_s \neq 0$, and η_z from η_1 at any $s \in (0,1]$ such that $\sigma_s > 0$: in particular

$$\forall (s,x,x_0) \in (0,1] \times \mathbb{R}^d \times \mathbb{R}^d \text{ with } \sigma_s > 0 \ : \quad \eta_z(s,x,x_0) = \frac{x - \alpha_s x_0 - \beta_s \eta_1(s,x,x_0)}{\sigma_s}. \tag{44}$$

The proof of Theorem B.4 will rely the following result:

Lemma B.6. *We have*

$$\forall (s,x,x_0) \in (0,1) \times \mathbb{R}^d \times \mathbb{R}^d : \quad \eta_1(s,x,x_0) = \frac{\int_{\mathbb{R}^d} x_1 \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2 + m_s x_1 \cdot (x - \alpha_s x_0)} dx_1}{\int_{\mathbb{R}^d} \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2 + m_s x_1 \cdot (x - \alpha_s x_0)} dx_1} \tag{45}$$

where $\eta_1(s,x,x_0) = \mathbb{E}^{x_0}[x_1|x_s = x, x_0]$ *and we defined*

$$M_s = \frac{\beta_s^2}{s\sigma_s^2}, \qquad m_s = \frac{\beta_s}{s\sigma_s^2}. \tag{46}$$

Proof: By definition,

$$\eta_1(s,x,x_0) = \frac{\int_{\mathbb{R}^d} x_1 \rho_c(x_1|x_0) e^{-\frac{1}{2}|z|^2} \delta(x - \alpha_s x_0 - \beta_s x_1 - \sigma_s \sqrt{s} z) dz dx_1}{\int_{\mathbb{R}^d} \rho_c(x_1|x_0) e^{-\frac{1}{2}|z|^2} \delta(x - \alpha_s x_0 - \beta_s x_1 - \sigma_s \sqrt{s} z) dz dx_1}, \tag{47}$$

where $\delta(x)$ denotes the Dirac delta distribution. For any $s \in (0,1]$ such that $\sigma_s > 0$, we can perform the integration over z explicitly to get

$$\eta_1(s,x,x_0) = \frac{\int_{\mathbb{R}^d} x_1 \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2 + m_s x_1 \cdot (x - \alpha_s x_0)} e^{-\frac{1}{2}s^{-1}\sigma_s^{-2}|x - \alpha_s x_0|^2} dx_1}{\int_{\mathbb{R}^d} \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2 + m_s x_1 \cdot (x - \alpha_s x_0)} e^{-\frac{1}{2}s^{-1}\sigma_s^{-2}|x - \alpha_s x_0|^2} dx_1}. \tag{48}$$

Since the factors $e^{-\frac{1}{2}s^{-1}\sigma_s^{-2}|x - \alpha_s x_0|^2}$ at the numerator and the denominator do not depend on x_1, they can be taken out of the integrals and simplified, and we arrive at (45). □

Proof of Theorem B.4. The only statement that is not a direct consequence of Theorem B.2 is (42). To establish these limits, notice that, since $\beta \in C^2([0,1])$ and $\dot{\beta}_0 = 0$, we must have $\beta_s = O(s^2)$ as $s \to 0$. As a result, since $\sigma \in C^1([0,1])$ and $\sigma_0 > 0$, we have

$$\lim_{s\to 0} M_s = \lim_{s\to 0} m_s = 0. \tag{49}$$

As a result,

$$\forall (x,x_0) \in \mathbb{R}^d \times \mathbb{R}^d \ : \quad \lim_{s\to 0} \eta_1(s,x,x_0) = \mathbb{E}^{x_0}[x_1], \qquad \lim_{s\to 0} \eta_z(s,x,x_0) = \frac{x - x_0}{\sigma_0}, \tag{50}$$

which establishes the limits in (42), where we used the formula in (44) to derive the limits of η_z from that of η_1. Note that in the above derivation, we need to verify the interchange of limits and integrations; it is guaranteed by using Assumption B.3 and the Lebesgue dominated convergence theorem since, for a fixed x, x_0, when s is sufficiently small, the factor

$\rho_c(x_1|x_0)e^{-\frac{1}{2}M_s|x_1|^2+m_s x_1\cdot(x-\alpha_s x_0)}$ is dominated by $\rho_c(x_1|x_0)e^{\frac{1}{2}C_1|x|}$, which is integrable as a function of x_1 due to our Assumption B.3.

To analyze the limit of $\nabla_x b_s(x,x_0)$, using Lemma B.6, we have

$$\nabla_x \eta_1(s,x,x_0) = m_s \frac{P(s,x,x_0)Q(s,x,x_0) - R(s,x,x_0)R^T(s,x,x_0)}{Q(s,x,x_0)^2}, \tag{51}$$

where

$$\begin{aligned} P(s,x,x_0) &= \int_{\mathbb{R}^d} x_1 x_1^T \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2+m_s x_1\cdot(x-\alpha_s x_0)} dx_1, \\ Q(s,x,x_0) &= \int_{\mathbb{R}^d} \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2+m_s x_1\cdot(x-\alpha_s x_0)} dx_1, \\ R(s,x,x_0) &= \int_{\mathbb{R}^d} x_1 \rho_c(x_1|x_0) e^{-\frac{1}{2}M_s|x_1|^2+m_s x_1\cdot(x-\alpha_s x_0)} dx_1. \end{aligned}$$

To derive the above formula, we also need verify the interchange of limits and integrations; it is again guaranteed by using Assumption B.3 and the Lebesgue dominated convergence theorem, for sufficiently small s. We know that

$$\lim_{s\to 0} \frac{P(s,x,x_0)Q(s,x,x_0) - R(s,x,x_0)R^T(s,x,x_0)}{|Q(s,x,x_0)|^2} = \mathbb{E}[x_1 x_1^T|x_0] - \mathbb{E}[x_1|x_0]\mathbb{E}[x_1^T|x_0], \tag{52}$$

and thus $\lim_{s\to 0}\nabla_x \eta_1(s,x,x_0) = 0$ as $\lim_{s\to 0} m_s = 0$. Using the formula in (44), we get $\lim_{s\to 0}\nabla_x \eta_z(s,x,x_0) = \frac{1}{\sigma_0}Id$. Therefore,

$$\lim_{s\to 0} \nabla_x b_s(x,x_0) = \frac{\dot\sigma_0}{\sigma_0} Id, \tag{53}$$

which completes the proof. □

B.4. Changing the Diffusion Coefficient: Proof of Theorem 3.2

Theorem 3.2 and formula (10) are consequences of the following result:

Theorem B.7. *Let $g \in C^0([0,1])$ be such that $\lim_{s\to 0^+} s^{-1}[g_s^2 - \sigma_s^2]$ and $\lim_{s\to 1^-} g_s^2\sigma_s^{-1}$ exist. Let b be given by (31) and define*

$$\begin{aligned} &\forall s \in (0,1): && A_s = [s\sigma_s(\dot\beta_s\sigma_s - \beta_s\dot\sigma_s)]^{-1} \\ &\forall (s,x,x_0) \in (0,1)\times\mathbb{R}^d\times\mathbb{R}^d: && c_s(x,x_0) = \dot\beta_s x + (\beta_s\dot\alpha_s - \dot\beta_s\alpha_s)x_0. \end{aligned} \tag{54}$$

Then we have

$$\forall (s,x,x_0) \in (0,1)\times\mathbb{R}^d\times\mathbb{R}^d: \quad \nabla \log \rho_s(x|x_0) = A_s\left[\beta_s b_s(x,x_0) - c_s(x,x_0)\right], \tag{55}$$

where $\rho_s(x|x_0)$ denotes the PDF of $X_s \overset{d}{=} X_s^g \overset{d}{=} I_s|x_0$. In addition the drift

$$\begin{aligned} b_s^g(x,x_0) &= b_s(x,x_0) + \tfrac{1}{2}(g_s^2 - \sigma_s^2)\nabla\log\rho_s(x|x_0) \\ &= b_s(x,x_0) + \tfrac{1}{2}(g_s^2 - \sigma_s^2)A_s\left[\beta_s b_s(x,x_0) - c_s(x,x_0)\right] \end{aligned} \tag{56}$$

is well-defined for all $(s,x,x_0) \in (0,1)\times\mathbb{R}^d\times\mathbb{R}^d$, and has finite limits at $s = 0,1$ for all $(x,x_0) \in \mathbb{R}^d\times\mathbb{R}^d$. Finally, the solutions to the SDE

$$dX_s^g = b_s^g(X_s^g,x_0)ds + g_s dW_s, \qquad X_{s=0}^g = x_0, \tag{57}$$

are such that Law(X_s^g) = Law(X_s) = Law$(I_s|x_0)$ for all $(s,x_0) \in [0,1]\times\mathbb{R}^d$. In particular $X_{s=1}^g \sim \rho_c(\cdot|x_0)$.

Proof. Let us first establish (55). By a direct extension of Stein's formula (Albergo et al., 2023), we have

$$\forall (s,x,x_0) \in (0,1)\times\mathbb{R}^d\times\mathbb{R}^d: \quad \nabla\log\rho_s(x|x_0) = -\frac{1}{\sqrt{s}\sigma_s}\mathbb{E}^{x_0}[z|x_s = x] \tag{58}$$

where $x_s = \alpha_s x_0 + \beta_s x_1 + \sqrt{s}\sigma_s z$ and where we used $x_s|x_0 \overset{d}{=} I_s|x_0 \overset{d}{=} X_s$ for all $s \in [0,1]$. Since

$$\begin{aligned} b_s(x,x_0) &= \dot\alpha_s x_0 + \dot\beta_s\mathbb{E}^{x_0}[x_1|x_s = x] + \sqrt{s}\dot\sigma_s\mathbb{E}^{x_0}[z|x_s = x] \\ x &= \alpha_s x_0 + \beta_s\mathbb{E}^{x_0}[x_1|x_s = x] + \sqrt{s}\sigma_s\mathbb{E}^{x_0}[z|x_s = x] \end{aligned} \tag{59}$$

we deduce that

$$\mathbb{E}^{x_0}[z|x_s = x] = \frac{\beta_s b_s(x, x_0) - \dot{\beta}_s x - (\beta_s \dot{\alpha}_s - \dot{\beta}_s \alpha_s) x_0}{\sqrt{s}(\dot{\sigma}_s \beta_s - \sigma_s \dot{\beta}_s)} \tag{60}$$

which can be inserted in (58) to show that (55) holds.

Second, note that the PDF $\rho_s(x|x_0)$ of the solution to the SDE (57) is the same as the PDF of the solution to

$$dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s, \quad X_{s=0} = x_0, \tag{61}$$

since we can use the identity $\frac{1}{2}\sigma_s^2 \Delta\rho_s = \frac{1}{2} g_s^2 \Delta\rho_s - \frac{1}{2}(g_s^2 - \sigma_s^2)\nabla \cdot (\rho_s \nabla \log \rho_s)$ to show that both densities satisfy the same Fokker-Planck equation (7).

It remains to show that the drift coefficient (56) is well-defined. To this end let us write it explicitly as

$$b_s^g(x, x_0) = b_s(x, x_0) + \left(\frac{g_s^2 - \sigma_s^2}{2s\sigma_s}\right)\left(\frac{\beta_s b_s(x, x_0) - \dot{\beta}_s x - (\beta_s \dot{\alpha}_s - \dot{\beta}_s \alpha_s) x_0}{\sigma_s \dot{\beta}_s - \dot{\sigma}_s \beta_s}\right). \tag{62}$$

Our assumptions that $\alpha, \beta, \sigma \in C^1([0,1])$ and satisfy $\alpha_0 = \beta_1 = 1$, $\alpha_1 = \beta_0 = \sigma_1 = 0$, and $\dot{\beta}_s > 0$ for all $s \in (0,1]$ and $\dot{\sigma}_s < 0$ for all $s \in [0,1]$ guarantee that $\beta_s > 0$ for all $s \in (0,1]$ and $\sigma_s > 0$ for all $s \in [0,1)$ and, as a result, $[s\sigma_s(\dot{\beta}_s \sigma_s - \beta_s \dot{\sigma}_s)]^{-1}$ is positive and finite for all $s \in (0,1)$. These assumptions also guarantee that $\beta_s[\sigma_s \dot{\beta}_s - \dot{\sigma}_s \beta_s]^{-1}$ and $\dot{\beta}_s[\sigma_s \dot{\beta}_s - \dot{\sigma}_s \beta_s]^{-1}$ have finite limits at $s = 0, 1$. Therefore the only factor in (62) that can be singular is $(g_s^2 - \sigma_s^2)/(2s\sigma_s)$ at $s = 0$, because of the factor s^{-1}, and at $s = 1$ because of the factor σ_1^{-1}. These singularities disappear under our assumptions that $\lim_{s\to 0^+} s^{-1}[g_s^2 - \sigma_s^2]$ and $\lim_{s\to 1^-} g_s^2 \sigma_s^{-1}$ exist and are finite. Therefore the drift b_s^g has the same regularity properties has b_s. □

B.5. Maximizing the likelihood with respect to the noise schedule

Let $X^g = (X_s^g)_{s\in[0,1]}$ be the process defined by the SDE (57), and let $\hat{X}^g = (\hat{X}_s^g)_{s\in[0,1]}$ denote an approximate, learned process governed by

$$d\hat{X}_s^g = \left(1 + \tfrac{1}{2}\beta_s A_s (g_s^2 - \sigma_s^2)\right) \hat{b}_s(\hat{X}_s^g, x_0)ds - A_s c_s(\hat{X}_s^g, x_0)ds + g_s dW_s, \qquad \hat{X}_{s=0}^g = x_0, \tag{63}$$

With a slight abuse of notation, let us denote by $D_{\mathrm{KL}}(X^g || \hat{X}^g)$ the KL divergence of the path measure of X^g from the path measure of $\hat{X}^g$. By Girsanov's theorem, it is given by

$$D_{\mathrm{KL}}(X^g || \hat{X}^g) = \frac{1}{2}\int_0^1 g_s^{-2} |1 + \tfrac{1}{2}\beta_s A_s (g_s^2 - \sigma_s^2)|^2 \mathbb{E}|b_s^g(X_s^g, x_0) - \hat{b}_s^g(X_s^g, x_0)|^2 ds, \tag{64}$$

where $\mathbb{E}$ denotes an expectation over the law of X^g. Using $X_s^g \overset{d}{=} I_s|x_0$, (64) can also be written as

$$\begin{aligned} D_{\mathrm{KL}}(X^g || \hat{X}^g) &= \frac{1}{2}\int_0^1 g_s^{-2} |1 + \tfrac{1}{2}\beta_s A_s (g_s^2 - \sigma_s^2)|^2 L_s ds, \\ &= \frac{1}{2}\int_0^1 g_s^{-2} |1 - \tfrac{1}{2}\beta_s A_s \sigma_s^2 + \tfrac{1}{2}\beta_s A_s g_s^2|^2 L_s ds, \end{aligned} \tag{65}$$

where A_s is given in (11) and $L_s = \mathbb{E}^{x_0}\left[|\hat{b}_s(I_s, x_0) - b_s(I_s, x_0)|^2\right]$. We now view the KL divergence (65) as an objective function for g, and solve the optimization problem

$$\min_g D_{\mathrm{KL}}(X^g || \hat{X}^g). \tag{66}$$

Since L_s is independent of g_s, minimizing (65) amounts to minimizing

$$g_s^{-2} |1 - \tfrac{1}{2}\beta_s A_s \sigma_s^2 + \tfrac{1}{2}\beta_s A_s g_s^2|^2 \tag{67}$$

for all $s \in [0,1]$. Since $A_s > 0$, (67) is minimized at

$$g_s^2 = \frac{\frac{1}{2}\beta_s A_s \sigma_s^2 - 1}{\frac{1}{2}\beta_s A_s} \qquad \text{when} \quad 1 - \tfrac{1}{2}\beta_s A_s \sigma_s^2 \le 0 \qquad \text{with} \quad (67) = 0, \tag{68}$$

and at

$$g_s^2 = \frac{1-\frac{1}{2}\beta_s A_s \sigma_s^2}{\frac{1}{2}\beta_s A_s} \qquad \text{when} \quad 1-\tfrac{1}{2}\beta_s A_s\sigma_s^2 > 0 \qquad \text{with} \quad (67) = 2\beta_s A_s(1-\tfrac{1}{2}\beta_s A_s \sigma_s^2) > 0. \tag{69}$$

Together, these relations show that the minimizer of (67) is $g_s = g_s^{\mathrm{F}}$ with

$$g_s^{\mathrm{F}} = \left|\frac{1-\frac{1}{2}\beta_s A_s\sigma_s^2}{\frac{1}{2}\beta_s A_s}\right|^{1/2} = \left|2s\sigma_s(\beta_s^{-1}\dot\beta_s\sigma_s - \dot\sigma_s) - \sigma_s^2\right|^{1/2}, \tag{70}$$

where we used the expression for A_s in (11) to get the second equality. This result is (13). Notice that we can also write

$$2s\sigma_s(\beta_s^{-1}\dot\beta_s\sigma_s - \dot\sigma_s) - \sigma_s^2 = 2s\sigma_s^2 \frac{d}{ds}\log\frac{\beta_s}{\sqrt{s}\sigma_s}. \tag{71}$$

Since the sign of this expression is the same as the sign of $1-\frac{1}{2}\beta_s A_s\sigma_s^2$, it shows that (68) and (69) arise at values of $s\in[0,1]$ where $\beta_s/[\sqrt{s}\sigma_s]$ is respectively decreasing or increasing in s. Since $\dot\beta_s > 0$ and $\dot\sigma_s < 0$, $\beta_s/[\sqrt{s}\sigma_s]$ cannot be decreasing for all $s\in[0,1]$, i.e. the minimum of (67) cannot be zero for all $s\in[0,1]$. As a result, the minimum of the KL divergence (65) must be positive if $L_s > 0$.

Finally, let us investigate the conditions on g_s in Theorem B.7 if we use the SDE (63) with $g_s = g_s^{\mathrm{F}}$. It is easy to see from (70) that the second condition is always satisfied since

$$\lim_{s\to 1^-} |g_s^{\mathrm{F}}|^2\sigma_s^{-1} = \lim_{s\to 1^-}[2s(\beta_s^{-1}\dot\beta_s\sigma_s - \dot\sigma_s) - \sigma_s] = -2\dot\sigma_1 > 0 \tag{72}$$

since $\sigma_1 = 0$. Regarding the first condition, we have

$$\lim_{s\to 0^+} s^{-1}[|g_s^{\mathrm{F}}|^2 - \sigma_s^2] = \lim_{s\to 0^+}[2\sigma_s(\beta_s^{-1}\dot\beta_s\sigma_s - \dot\sigma_s) - 2s^{-1}\sigma_s^2] = 2\lim_{s\to 0^+}[\beta_s^{-1}\dot\beta_s - s^{-1}] - 2\dot\sigma_0\sigma_0 \tag{73}$$

since $\sigma_0 > 0$, $\dot\sigma_0 < 0$. Since $\beta_0 = 0$, if $\dot\beta_0 > 0$ and $\ddot\beta_0$ exists, we have

$$\lim_{s\to 0^+}[\beta_s^{-1}\dot\beta_s - s^{-1}] = \tfrac{1}{2}\ddot\beta_0\dot\beta_0^{-1}, \tag{74}$$

so that

$$\lim_{s\to 0^+} s^{-1}[|g_s^{\mathrm{F}}|^2 - \sigma_s^2] = \sigma_0^2\ddot\beta_0\dot\beta_0^{-1} - 2\dot\sigma_0\sigma_0 \qquad \text{if } \dot\beta_0 > 0 \text{ and } \ddot\beta_0 \text{ exists.} \tag{75}$$

If, however $\dot\beta_0 = 0$, then $g_0^{\mathrm{F}} \neq \sigma_0$, and the first condition is not satisfied since $\lim_{s\to 0^+} s^{-1}[|g_s^{\mathrm{F}}|^2 - \sigma_s^2]$ does not exist. In this case, we need to consider more carefully how to define the solution to the SDE (63). In the specific case when $\alpha_s = 1-s$, $\sigma_s = \varepsilon(1-s)$ and $\beta_s = s^2$, if we set $g_s = g_s^{\mathrm{F}} = \varepsilon\sqrt{(1-s)(3-s)}$ in (63), this SDE reduces to (see the explicit formulas given in Appendix B.1 and denoting $X_s^{\mathrm{F}} = X_s^{g^{\mathrm{F}}}$):

$$dX_s^{\mathrm{F}} = \Big(1+\frac{1}{2-s}\Big)b_s(X_s^{\mathrm{F}}, x_0)ds - \frac{1}{s(2-s)}(2X_s^{\mathrm{F}} - (2-s)x_0)ds + \varepsilon\sqrt{(1-s)(3-s)}dW_s, \qquad X_{s=0}^{\mathrm{F}} = x_0. \tag{76}$$

The drift in this equation is singular at $s=0$ because of the term $2(X_s^{\mathrm{F}} - x_0)/[s(2-s)]$. Nevertheless, the solution to this SDE is well-defined for the initial condition $X_{s=0}^{\mathrm{F}} = x_0$, and satisfies the integral equation

$$X_s^{\mathrm{F}} = x_0 + \frac{2-s}{s}\int_0^s \frac{u}{2-u}\left(\Big(1+\frac{1}{2-u}\Big)b_u(X_u^{\mathrm{F}}, x_0) - \frac{1}{(2-u)}x_0\right)du + \varepsilon\frac{2-s}{s}\int_0^s \frac{u\sqrt{(1-u)(3-u)}}{2-u}dW_u. \tag{77}$$

Since $b_0(x_0, x_0) = -x_0$ when $\dot\beta_0 = 0$, this equation implies that

$$X_s^{\mathrm{F}} \stackrel{d}{=} x_0(1-s) + \varepsilon W_s + o(s). \tag{78}$$

which is also the law of $I_s|x_0$ as it should.

B.6. Connection with Föllmer Processes and Proof of Theorem 3.3

To begin, we give some background on the Föllmer process (Föllmer, 1986; Tzen & Raginsky, 2019). Originally, the Föllmer process was defined as the process $X = (X_s)_{s\in[0,1]}$ whose path measure has minimal KL divergence from the path measure of the Wiener process $W = (W_s)_{s\in[0,1]}$ under the constraint that $X_{s=1}$ be distributed according to some target distribution. This Föllmer process can be generalized to "reference processes" that differ from the standard Wiener process: as we will show next, in our context the natural reference process is the solution to the linear SDE

$$dY_s = a_s(Y_s - \alpha_s x_0)ds + \dot{\alpha}_s x_0 ds + g_s^{\mathrm{F}} dW_s, \qquad Y_{s=0} = x_0, \tag{79}$$

where g_s^{F} is given by (70) and where we have defined

$$\begin{aligned} a_s &:= \beta_s^{-1}\dot{\beta}_s - \frac{2s\sigma_s(\beta_s^{-1}\dot{\beta}_s\sigma_s - \dot{\sigma}_s) - \sigma_s^2}{\beta_s^2 + s\sigma_s^2}, \\ &= \beta_s^{-1}\dot{\beta}_s - \frac{2\beta_s^{-1}\dot{\beta}_s(\beta_s^2 + s\sigma_s^2) - 2\beta_s\dot{\beta}_s - 2s\sigma_s\dot{\sigma}_s - \sigma_s^2}{\beta_s^2 + s\sigma_s^2}, \\ &= -\beta_s^{-1}\dot{\beta}_s + \frac{2\beta_s\dot{\beta}_s + 2s\sigma_s\dot{\sigma}_s + \sigma_s^2}{\beta_s^2 + s\sigma_s^2} = \frac{d}{ds}\log\frac{\beta_s^2 + s\sigma_s^2}{\beta_s}. \end{aligned} \tag{80}$$

We can then define the Föllmer process $X^{\mathrm{F}} = (X_s^{\mathrm{F}})_{s\in[0,1]}$ by adjusting the drift $\check{b}_s(x, x_0)$ in

$$d\check{X}_s = \check{b}_s(\check{X}_s, x_0)ds + g_s^{\mathrm{F}} dW_s, \qquad \check{X}_{s=0}^{\mathrm{F}} = x_0, \tag{81}$$

in such a way that the KL divergence of the path measure of $\check{X}$ from the path measure of Y (the solution to (79)) is minimized subject to the constraint that $\check{X}_{s=1} \sim \rho_c(\cdot|x_0)$. That is, the Föllmer process is defined via

$$\min_{\check{b}} D_{\mathrm{KL}}(\check{X}||Y) \qquad \text{subject to} \quad \check{X}_{s=1} \sim \rho_c(\cdot|x_0). \tag{82}$$

This traditional minimization problem for the Föllmer process is distinct from the minimization problem (66) considered in Appendix B.5. Nevertheless, our next result shows that the minimizers of (66) and (82) coincide:

Theorem B.8. *Assume that $\beta_s/[\sqrt{s}\sigma_s]$ is non-decreasing on $s \in [0,1]$. Then, the Föllmer process associated with the reference process $(Y_s)_{s\in[0,1]}$ that solves (79) is the process $X^{\mathrm{F}} \equiv (X_s^{g^{\mathrm{F}}})_{s\in[0,1]}$ that solves (57) with $g_s = g_s^{\mathrm{F}}$ given by (70).*

Theorem 3.3 is implied by this result.

Proof. We closely follow the steps of Föllmer's original construction involving time-reversal (Föllmer, 1986). To begin, notice that we can solve (55) to express b_s in terms of the score $\nabla \log \rho_s$. We can then use the resulting expression in (56) to write the SDE (57) as

$$dX_s^g = \left(\beta_s^{-1}A_s^{-1} + \tfrac{1}{2}(g_s^2 - \sigma_s^2)\right)\nabla \log \rho_s(X_s^g|x_0)ds + \beta_s^{-1}c_s(X_s^g, x_0)ds + g_s dW_s, \qquad X_{s=0}^g = x_0, \tag{83}$$

where A_s and $c_s(x, x_0)$ are defined in (11). Using the score, we can time-reverse (83) and derive the following SDE for $X_s^{\mathrm{R}} \stackrel{d}{=} X_{1-s}^g$

$$dX_s^{\mathrm{R}} = -\left(\beta_{1-s}^{-1}A_{1-s}^{-1} - \tfrac{1}{2}(g_{1-s}^2 + \sigma_{1-s}^2)\right)\nabla \log \rho_{1-s}(X_s^{\mathrm{R}}|x_0)ds - \beta_{1-s}^{-1}c_{1-s}(X_s^{\mathrm{R}}, x_0)ds + g_{1-s}dW_s. \tag{84}$$

If we take $X_{s=0}^{\mathrm{R}} \sim \rho_c(\cdot|x_0)$ in (84), then by construction we have that $X_{s=1}^{\mathrm{R}} = x_0$. Remarkably, if we use $g_s = g_s^{\mathrm{F}}$ with g_s^{F} given in (70), and if we choose $1 - \frac{1}{2}\beta_s A_s \sigma_s^2 \geq 0$ (i.e. choose $\beta_s/[\sqrt{s}\sigma_s]$ nondecreasing), the SDE (84) reduces to

$$dX_s^{\mathrm{R}} = -\beta_{1-s}^{-1}c_{1-s}(X_s^{\mathrm{R}}, x_0)ds + g_{1-s}^{\mathrm{F}}dW_s, \qquad X_{s=0}^{\mathrm{R}} \sim \rho_c(\cdot|x_0). \tag{85}$$

This reverse-time SDE has the remarkable property that its drift is independent of the score $\nabla \log \rho_s$, meaning that the information about the target PDF $\rho_c(\cdot|x_0)$ only enters through its initial condition $X_{s=0}^{\mathrm{R}} \sim \rho_c(\cdot|x_0)$. This means that we can change the initial condition in (85) to construct (after reversing time back) a reference process. We can use any density for this purpose, but for simplicity it is convenient to choose a Gaussian, and therefore to consider

$$dY_s^{\mathrm{R}} = -\beta_{1-s}^{-1}c_{1-s}(Y_s^{\mathrm{R}}, x_0)ds + g_{1-s}^{\mathrm{F}}dW_s, \qquad Y_{s=0}^{\mathrm{R}} \sim \mathsf{N}(0, Id). \tag{86}$$

Using the explicit form of c_s in (54), we can write this SDE explicitly as

$$dY_s^{\mathrm{R}} = -\beta_{1-s}^{-1}\dot{\beta}_{1-s}(Y_s^{\mathrm{R}} - \alpha_{1-s}x_0)ds - \dot{\alpha}_{1-s}x_0 + g_{1-s}^{\mathrm{F}}dW_s, \qquad Y_{s=0}^{\mathrm{R}} \sim \mathsf{N}(0, Id). \tag{87}$$

Using that $d\alpha_{1-s} = -\dot{\alpha}_{1-s}ds$ and $d\beta_{1-s} = -\dot{\beta}_{1-s}ds$, we may rewrite this as

$$d\big(\beta_{1-s}^{-1}(Y_s^{\mathrm{R}} - \alpha_{1-s}x_0)\big) = \beta_{1-s}^{-1}g_{1-s}^{\mathrm{F}}dW_s, \qquad Y_{s=0}^{\mathrm{R}} \sim \mathsf{N}(0, Id). \tag{88}$$

Hence,

$$Y_s^{\mathrm{R}} = \alpha_{1-s}x_0 + \beta_{1-s}Y_{s=0}^{\mathrm{R}} + \beta_{1-s}\int_0^s \beta_{1-u}^{-1}g_{1-u}^{\mathrm{F}}dW_u. \tag{89}$$

Using the explicit form of g_s^{F} given in (70) together with (71), we deduce that

$$\begin{aligned}
\mathbb{E}\left[\left|\beta_{1-s}\int_0^s \beta_{1-u}^{-1}g_{1-u}^{\mathrm{F}}dW_u\right|^2\right] &= \beta_{1-s}^2\int_0^s \beta_{1-u}^{-2}|g_{1-u}^{\mathrm{F}}|^2du, \\
&= \beta_{1-s}^2\int_{1-s}^1 \beta_u^{-2}|g_u^{\mathrm{F}}|^2du, \\
&= \beta_{1-s}^2\int_{1-s}^1 \frac{2u\sigma_u^2}{\beta_u^2}\frac{d}{du}\log\frac{\beta_u}{\sqrt{u}\sigma_u}du, \\
&= -\beta_{1-s}^2\int_{1-s}^1 d\left(\frac{u\sigma_u^2}{\beta_u^2}\right), \\
&= (1-s)\sigma_{1-s}^2,
\end{aligned} \tag{90}$$

where we used $\sigma_1 = 0$. This means that

$$Y_{1-s}^{\mathrm{R}} \overset{d}{=} \alpha_s x_0 + \beta_s z + \sigma_s W_s \sim \mathsf{N}(\alpha_s x_0, \beta_s^2 + s\sigma_s^2), \tag{91}$$

where $z \sim \mathsf{N}(0, Id)$ with $z \perp W$. Note that, unsurprisingly, the process on the right-hand side is simply the stochastic interpolant (2) conditioned on x_0 fixed, with x_1 replaced by a Gaussian z. Denoting by $\rho_s^Y(y|x_0)$ the PDF of Y_{1-s}^{R}, (91) implies that

$$\nabla \log \rho_s^Y(y|x_0) = -\frac{y - \alpha_s x_0}{\beta_s^2 + s\sigma_s^2}. \tag{92}$$

We can use this result to time reverse (86) and obtain the following SDE for $Y_s \overset{d}{=} Y_{1-s}^{\mathrm{R}}$

$$dY_s = \beta_s^{-1}\dot{\beta}_s(Y_s - \alpha_s x_0)ds + \dot{\alpha}_s x_0 + |g_s^{\mathrm{F}}|^2\nabla \log \rho_s^Y(Y_s|x_0)ds + g_s^{\mathrm{F}}dW_s \qquad Y_{s=0} = x_0. \tag{93}$$

If we insert the explicit form of $|g_s^{\mathrm{F}}|^2 = 2s\sigma_s(\beta_s^{-1}\dot{\beta}_s\sigma_s - \dot{\sigma}_s) - \sigma_s^2$ given in (70) into (93) we get

$$dY_s = a_s(Y_s - \alpha_s x_0)ds + \dot{\alpha}_s x_0 ds + g_s^{\mathrm{F}}dW_s, \qquad Y_{s=0} = x_0, \tag{94}$$

with

$$a_s = \beta_s^{-1}\dot{\beta}_s - \frac{2s\sigma_s(\beta_s^{-1}\dot{\beta}_s\sigma_s - \dot{\sigma}_s) - \sigma_s^2}{\beta_s^2 + s\sigma_s^2}, \tag{95}$$

i.e. we recover the SDE (79). Since $a_s = \frac{d}{ds}\log\frac{\beta_s^2 + s\sigma_s^2}{\beta_s}$, we can solve the above the SDE to obtain the reference process as

$$Y_s = \alpha_s x_0 + \frac{\beta_s^2 + s\sigma_s^2}{\beta_s}\int_0^s \frac{\beta_u g_u^{\mathrm{F}}}{\beta_u^2 + u\sigma_u^2}dW_u.$$

We can then verify that

$$\begin{aligned}
\mathbb{E}\left[\left|\frac{\beta_s^2 + s\sigma_s^2}{\beta_s}\int_0^s \frac{\beta_u g_u^{\mathrm{F}}}{\beta_u^2 + u\sigma_u^2}dW_u\right|^2\right] &= (\frac{\beta_s^2 + s\sigma_s^2}{\beta_s})^2\int_0^s \frac{\beta_u^2\sigma_u^2 u}{(\beta_u^2 + \sigma_u^2 u)^2}\frac{d}{du}\log\frac{\beta_u^2}{\sigma_u^2 u}du \\
&= (\frac{\beta_s^2 + s\sigma_s^2}{\beta_s})^2\int_0^s \frac{1}{((\beta_u^2/\sigma_u^2 u) + 1)^2}d(\beta_u^2/\sigma_u^2 u) \\
&= (\frac{\beta_s^2 + s\sigma_s^2}{\beta_s})^2\frac{\beta_s^2}{\beta_s^2 + s\sigma_s^2} = \beta_s^2 + s\sigma_s^2,
\end{aligned} \tag{96}$$

which implies that $Y_s \stackrel{d}{=} \alpha_s x_0 + \beta_s z + \sigma_s W_s \sim \mathsf{N}(\alpha_s x_0, \beta_s^2 + s\sigma_s^2)$; this matches our previous calculation (91).

It remains to show that the process $X^{\mathrm{F}} \equiv X^{g^{\mathrm{F}}}$ defined by solution to the SDE (57) with $g_s = g_s^{\mathrm{F}}$ given in (70) is the Föllmer process associated with the process Y defined by the solution to (93). To this end, recall that the KL divergence between two path measures is invariant under time-reversal, so that

$$D_{\mathrm{KL}}(\check{X}||Y) = D_{\mathrm{KL}}(\check{X}^{\mathrm{R}}||Y^{\mathrm{R}}) \tag{97}$$

where $\check{X}$ is the process defined by the solution to the SDE (81), and where $\check{X}^{\mathrm{R}}$ is its time-reversal. We can now use following decomposition, known as "disintegration" (Léonard, 2014), of the Kullback-Leibler divergence

$$D_{\mathrm{KL}}(\check{X}^{\mathrm{R}}||Y^{\mathrm{R}}) = \int_{\mathbb{R}^d} D_{\mathrm{KL}}(\check{X}^{\mathrm{R},x}||Y^{\mathrm{R},x})\rho_c(x|x_0)dx + D_{\mathrm{KL}}[\rho_c(\cdot|x_0)\|\mathsf{N}(x_0, Id)] \tag{98}$$

where $\check{X}^{\mathrm{R},x}$ and $Y^{\mathrm{R},x}$ denote, respectively, the processes $\check{X}^{\mathrm{R}}$ and Y^{R} conditioned to start from x (i.e. on $\check{X}^{\mathrm{R},x}_{s=0} = Y^{\mathrm{R},x}_{s=0} = x$). In addition, we used the fact that $Y^{\mathrm{R}}_{s=0} \sim \mathsf{N}(0, Id)$ whereas $\check{X}^{\mathrm{R}}_{s=0} \sim \rho_c(\cdot|x_0)$ by the constraint imposed in the minimization problem (82). The second term at the right hand side of (98) is fixed due to this constraint; the first term is always non-negative, but we can make it zero if we take $\check{X}^{\mathrm{R}} = X^{\mathrm{R}}$ with X^{R} defined as the solution to (84) since this process is the same as the one defined by the solution to (86) if we condition both on $X^{\mathrm{R}}_{s=0} = Y^{\mathrm{R}}_{s=0} = x$. Therefore $\check{X}^{\mathrm{R}} = X^{\mathrm{R}}$ minimizes (98), which means that its time-reversal $X^{g^{\mathrm{F}}}$ minimizes (82). □

B.7. Analytic formula of b_s for Gaussian mixture distributions

When $\rho_c(x|x_0)$ is a Gaussian mixture model (GMM), the drift b_s is available analytically:

Proposition B.9. *Let the target density be a GMM with $J \in \mathbb{N}$ modes*

$$\rho_c(x|x_0) = \sum_{j=1}^{J} p_j \mathsf{N}(x; m_j, C_j) \tag{99}$$

where $p_j \geq 0$ with $\sum_{j=1}^{J} p_j = 1$, $m_j \in \mathbb{R}^d$, and $C_j = C_j^T \in \mathbb{R}^d \times \mathbb{R}^d$ positive-definite (with both m_j and C_j possibly dependent on x_0). Then

$$\begin{aligned} b_s(x, x_0) = \dot{\alpha}_s x_0 + \dot{\beta}_s \frac{\sum_{j=1}^{J} p_j m_j \mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))}{\sum_{j=1}^{J} p_j \mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))} \\ + \frac{\sum_{j=1}^{J} p_j (\beta_s \dot{\beta}_s C_j + s\sigma_s \dot{\sigma}_s Id)\overline{C}_j^{-1}(s)(x - \overline{m}_j(s))\mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))}{\sum_{j=1}^{J} p_j \mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))} \end{aligned} \tag{100}$$

where

$$\overline{m}_j(s) = \alpha_s x_0 + \beta_s m_j, \qquad \overline{C}_j(s) = \beta_s^2 C_j + s\sigma_s^2 Id. \tag{101}$$

Proof. By definition

$$\begin{aligned} b_s(x, x_0) &= \dot{\alpha}_s x_0 + \mathbb{E}[\dot{\beta}_s x_1 + \dot{\sigma}_s W_s | I_s = x, x_0] \\ &= \dot{\alpha}_s x_0 + \mathbb{E}[\dot{\beta}_s \beta_s^{-1}(x - \alpha_s x_0 - \sigma_s W_s) + \dot{\sigma}_s W_s | I_s = x, x_0] \\ &= \dot{\alpha}_s x_0 + \dot{\beta}_s \beta_s^{-1}(x - \alpha_s x_0) + s\sigma_s(\sigma_s \dot{\beta}_s \beta_s^{-1} - \dot{\sigma}_s)\nabla \log \rho_s(x|x_0). \end{aligned} \tag{102}$$

where we used the fact $\nabla \log \rho_s(x|x_0) = -[s\sigma_s]^{-1}\mathbb{E}^{x_0}[W_s|I_s = x]$. For the GMM,

$$\rho_s(x|x_0) = \sum_{j=1}^{J} p_j \mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s)), \tag{103}$$

so that

$$\nabla \log \rho_s(x|x_0) = -\frac{\sum_{j=1}^{J} p_j \overline{C}_j^{-1}(s)(x - \overline{m}_j(s))\mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))}{\sum_{j=1}^{J} p_j \mathsf{N}(x; \overline{m}_j(s), \overline{C}_j(s))}. \tag{104}$$

Inserting this expression in (102) we obtain

$$\begin{aligned}
&\frac{\dot\beta_s}{\beta_s}(x-\alpha_s x_0) + s\sigma_s^2\frac{\dot\beta_s}{\beta_s}\nabla\log\rho_s(x|x_0)\\
=&\frac{\dot\beta_s}{\beta_s}\left(x-\alpha_s x_0 - \frac{\sum_{j=1}^J p_j(Id-\beta_s^2 C_j\overline{C}_j^{-1}(s))(x-\overline{m}_j(s))\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}{\sum_{j=1}^J p_j\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}\right)\\
=&\frac{\dot\beta_s}{\beta_s}\left(\frac{\sum_{j=1}^J p_j\big(\beta_s m_j+\beta_s^2 C_j\overline{C}_j^{-1}(x-\overline{m}_j)\big)\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}{\sum_{j=1}^J p_j\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}\right)\\
=&\dot\beta_s\frac{\sum_{j=1}^J p_j m_j\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}{\sum_{j=1}^J p_j\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))} + \frac{\sum_{j=1}^J p_j\beta_s\dot\beta_s C_j\overline{C}_j^{-1}(x-\bar m_j)\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))}{\sum_{j=1}^J p_j\mathsf{N}(x;\overline{m}_j(s),\overline{C}_j(s))},
\end{aligned} \tag{105}$$

where in the first and second identities, we used the fact that $s\sigma_s^2 Id = Id - \beta_s^2 C_j\overline{C}_j^{-1}(s)$ and $x-\alpha_s x_0 = x-\overline{m}_j(s)+\beta_s m_j$.

Now, using $b_s(x,x_0) = \dot\alpha_s x_0 + \dot\beta_s\beta_s^{-1}(x-\alpha_s x_0) + s\sigma_s^2(\dot\beta_s\beta_s^{-1}-\dot\sigma_s)\nabla\log\rho_s(x|x_0)$, we get the final formula. □

C. Details of Numerical Experiments

C.1. Multi-modal jump-diffusion process

Inspired by (Chen et al., 2020), we create the synthetic example by starting with a Gaussian component $\mathsf{N}(m_0, C_0)$ with $m_0 = [5, 0]$ and

$$C_0 = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

We rotate this distribution counterclockwise by $2\pi/5$ four times to obtain the remaining four Gaussian modes. We assign each of the five modes equal weights to obtain our 2D Gaussian mixture model with density $\rho_{\text{GMM}}(x)$

The 2D particle jump-diffusion dynamics is constructed as follows. Between the jumps, the particle moves according to the Langevin dynamics

$$dx_t = \nabla\log\rho_{\text{GMM}}(x_t)dt + \sqrt{2}dW_t.$$

At jump times specified by a Poisson process with rate $\lambda = 2$, the particle is rotated counterclockwise by an angle $2\pi/5$.

We simulate this dynamics using the following scheme with $\delta t = 0.01$:

$$x_{(n+1)\delta t} = \begin{cases} x_{n\delta t} + \delta t\nabla\log p(x_{n\delta t}) + \sqrt{2\delta t}\xi & \text{with probability } 1-\lambda\delta t \\ R_{2\pi/5}(x_{n\delta t} + \delta t\nabla\log p(x_{n\delta t}) + \sqrt{2\delta t}\xi) & \text{with probability } \lambda\delta t \end{cases}$$

where $\xi \sim \mathsf{N}(0, I_{2\times 2})$ and $R_{2\pi/5}$ is the counterclockwise rotation operator in 2D with angle $2\pi/5$. We integrate this dynamics long enough to reach equilibrium and get enough data.

We keep the data at a regular time interval of $\Delta t = 0.5$ and use paired $(x_t, x_{t+0.5})$ as training data for learning the conditional distribution at lag $\tau = 0.5$. In total we store 10^5 training data pairs. We use a fully connected neural network with 5 hidden layes with hidden dimension 500 to approximate the velocity field $b_s(x, x_0)$ in the SDEs. The input to the net is of dimension 5 and the output is of dimension 2. We train the network using a batch size of 10^4, default AdamW optimizer with base learning rate $l = 10^{-3}$ and cosine scheduler that decreases in each epoch the learning rate eventually to 0 after 300 epochs. We test the SDEs for new simulated trajectory data.

C.2. 2D Stochastic Navier-Stokes Example

We set $\nu = 10^{-3}, \alpha = 0.1, \varepsilon = 1$. We consider the following random forcing

$$\begin{aligned}
\eta(t,x,y) =&\big(W_1(t)\sin(6x) + W_2(t)\cos(7x) + W_3(t)\sin(5(x+y)) + W_4(t)\cos(8(x+y))\big)\\
&+\big(W_5(t)\cos(6x) + W_6(t)\sin(7x) + W_7(t)\cos(5(x+y)) + W_8(t)\sin(8(x+y))\big),
\end{aligned} \tag{106}$$

where $W_i(t)$, $1 \leq i \leq 8$ are independent Wiener processes. For such random forcing, the NSE (17) has a unique invariant measure, as proved in Hairer & Mattingly (2006). Note that (106) is a Gaussian random field with covariance function $C(t, t', x, y, x', y') = \min(t, t')[\cos(6(x - x')) + \cos(7(x - x')) + \cos(5(x - x' + y - y') + \cos(8(x - x' + y - y'))]$ which is translation invariant in space. The damping term $-\alpha\omega$ is used to accelerate the mixing of the dynamics as it damps the vorticity at large length-scales/small Fourier modes to avoid all energy accumulating in large vortices.

We use a pseudo-spectral solver with Euler-Maruyama time-stepping scheme to solve the stochastic PDE in time and we use the jax-cfd package (Dresdner et al., 2022) for the mesh generation and domain discretization. We perform simulations with a grid sizes 256×256 which is fine enough for resolving the numerical solutions of the 2D stochastic Navier-Stokes equations with our specifications.

We notice that there are existing works on forecasting with generative models that experiment with the 2D NSE (Lienen et al., 2023) (Rühling Cachay et al., 2024); however, they are interested in deterministic forecasting. In contrast, we are interested in probabilistic forecasting which is why we add a stochastic forcing to the equation. The precise nature of this forcing is probably not important in practice. In all of our experiments, we normalize the training and testing data so that on average the L^2 norm of each snapshot field in the training data is 1; we found such normalization is useful to ensure a stable gradient that does not blow up during training. We use a standard UNet architecture popularized by Ho et al. (2020) to learn the drift velocity b_θ; the conditioning in the vector field is achieved by channel concatenation in the input, as this is the usual way to incorporate high dimensional conditioning in UNets for Diffusion (Ho et al., 2022b). The total parameters for the UNet is around 2 millions. We employ the default AdamW optimizer (Loshchilov & Hutter, 2017). We use a cosine annealing schedule to decrease the learning rate in each epoch. The base learning rate is $l = 10^{-3}$. We split the data into 90% training data and 10% test data. We use a batch size of 100. In total we train 50 epochs. The model is trained on a single Nvidia A100 GPU and it takes less than 1 day. Once trained, we test our SDEs model (e.g. see Fig. 3) by examining the conditional distributions. We fix 50 initial conditions drawn from the invariant measure and run simulations or our SDE models to the desired lags, and collect an ensemble of solutions (we choose the number of ensembles to be 300). We compute mean and std of these ensembles and compare them with those obtained by using the simulation data. We also consider the enstrophy spectrum of the conditional distribution, which measures the size of the vorticity at each scale and reveals important physical information. More precisely, we compute the enstrophy of the vorticity field ω at wavenumber $k \in \mathbb{R}_+$ is defined as

$$\text{Enstrophy}(k) = \sum_{k \leq |m| \leq k+1} |\hat{\omega}(m)|^2,$$

where $\hat{\omega}$ with $m \in \mathbb{Z}^2$ denotes the Fourier coefficients of ω. The enstrophy spectrum curve is the graph of the function $k \to \text{Enstrophy}(k)$. We can average the enstrophy over different simulation trajectories of ω to get the spectrum of the distribution of the field. We often smooth the curve to get rid of the sawtooth.

We provide some additional numerical results on the 2D stochastic Navier-Stokes example below.

Superresolution We consider superresolution, i.e., predicting the field with a resolution of 128×128, from the downsized version with a resolution of 32×32. In Fig. C.2, we show a 32×32 resolution field, as well as the true 128×128 resolution field and the mean of samples drawn from our SDEs. Our method achieves an outstanding recovery. Additionally, we calculate the standard deviation of the samples. The spatial distribution of this standard deviation serves as a tool for uncertainty quantification; notably, there exists a pronounced correlation between the pattern of standard deviation and the vorticity field.

Forecasting efficiency We conducted a comparative analysis of prediction speed between our models and the SPDE solver utilized for the 2D stochastic Navier-Stokes equation. For a scenario involving a short time lag of $\tau = 0.5$, wherein our model accurately captures the resulting conditional distribution, sampling from our SDE forecaster with 200 Euler-Maruyama steps takes 0.05 seconds, while executing the SPDE solver on the same Nvidia RTX8000 GPU requires 8 seconds. This observation highlights that our method accelerates forecasting by over 100 times without sacrificing physical information inherent in the conditional distribution. Furthermore, our approach could stand to gain even more acceleration in scenarios with lower viscosity, which typically necessitates a substantially reduced time step size for the SPDE solver.

Comparison with a deterministic approach We also contrast our probabilistic forecasting method with a deterministic approach. In the deterministic approach, we utilize the same UNet architecture to parameterize a point estimator, denoted as $\widehat{\omega}_{t+\tau} = f(\omega_t; \theta)$, and train it using the Mean Squared Error (MSE) loss function $\mathbb{E}\|\widehat{\omega}_{t+\tau} - \omega_{t+\tau}\|^2$, where $\tau = 0.5$

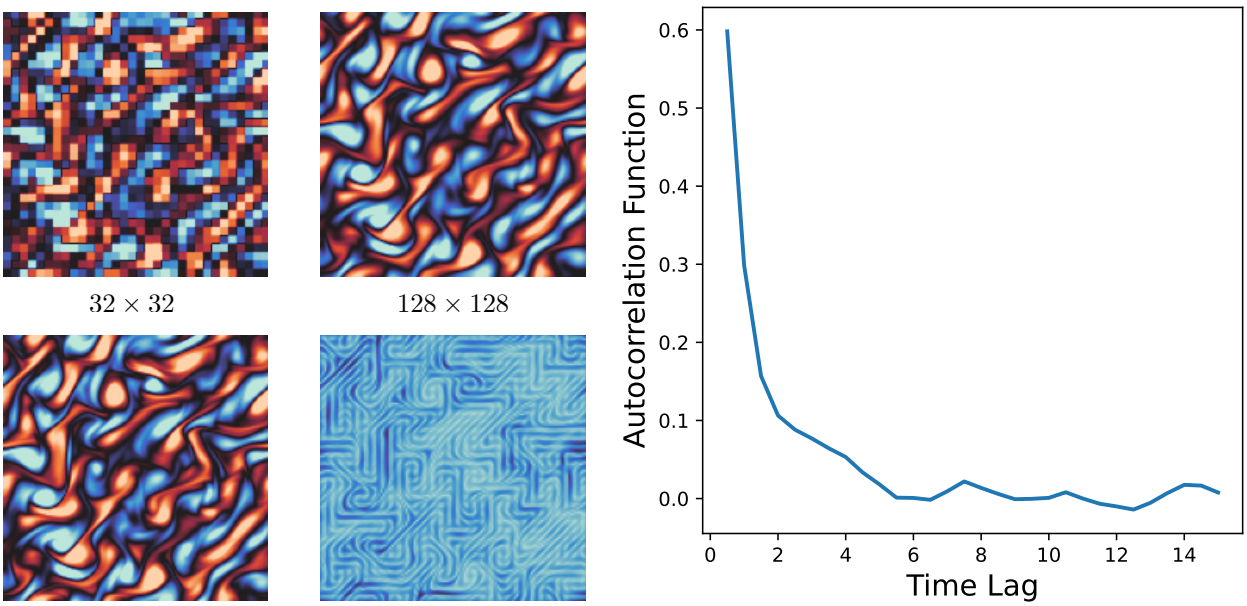

Figure 7. **Left panel:** Superresolution. Low-resolution and high-resolution field, mean and std of samples drawn from our learned SDE model. **Right panel:** We calculate the auto-correlation of our data as a function of time. The auto-correlation function is computed via averaging over the auto-correlation on all the grid points. Note that this averaging can be done because our random forcing term is spatially homogeneous and each grid point is identical in a statistical sense. We notice that the auto-correlation drops to about 0.25 at a time lag $\tau = 1$ and about 0.1 at a time lag $\tau = 2$, which indicates the potential diversity of the conditional distribution and thus the difficulty of our forecasting task.

represents the time lag we aim to predict. This Minimum Mean Square Error (MMSE) estimator is designed to capture the mean of the conditional probability distribution. While the deterministic point estimator performs adequately in recovering the mean, it falls short in long-term total enstrophy prediction due to its inability to account for fluctuations in variance; see Table 2. More precisely, we report the short time (lag = 0.5) error of the conditional mean prediction: for the probabilistic approach we compute the mean of generated samples and compare it with the true conditional mean, and for the deterministic approach we use the point estimator to compare. For the probabilistic approach, we can also compute the predicted conditional std using samples and compare with the true std. We note that such comparison is regarding the relative L^2 err of the field of conditional mean and std. More precisely,

$$\mathrm{err}(\mathrm{mean}) = \frac{\|\hat{m} - m\|_{L^2}}{\|m\|_{L^2}}, \quad \mathrm{err}(\mathrm{std}) = \frac{\|\hat{\sigma} - \sigma\|_{L^2}}{\|\sigma\|_{L^2}},$$

where $\hat{m}, \hat{\sigma}$ are estimated mean and std (in this numerical example, it is an image of size 128×128) and m, σ are the true mean and std computed from simulation data.

These short time metrics characterize the accuracy of the forecasting in one lag time. We can also iterate the learned SDE or deterministic maps for many steps and use the generated dynamics to make predictions on the invariant distribution of the true dynamics. We do so by iterating 100 steps and compute the averaged total enstrophy (which is the squared L^2 norm of the vorticity field). We compare the predicted total enstrophy of the invariant distribution with the truth, in relative error. Here, the averaged total enstrophy is a scalar quantity; in contrast to the conditional mean and std. We obtain much better estimate of averaged total enstrophy with our SDEs. The result demonstrates the necessity of using probabilistic forecasting for stochastic dynamical systems.

Comparisons between different SDE generative models All the above experiments on NSE is done with the interpolant $\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$, as it performs the best. Below in Figures 8 and 9, we also post the loss and gradient norm curves, as well as the enstrophy spectra of the generated samples, for other choices of interpolants and Föllmer processes (trained with the same network, data, and number of epochs). We observe that $\beta_s = s^2$, namely $\dot{\beta}_0 = 0$, is important to ensure a stable gradient norm curve. This experiment demonstrates the superiority of the interpolant

Relative error	Our SDEs	Deterministic
Short time L^2 error of mean	**1.1e-1**	2.3e-1
Short time L^2 error of std	6.8e-2	N/A
Long time error of averaged total enstrophy	**5.6e-3**	3.0e-1

Table 2. Short-time and long-time relative accuracy comparisons between the SDE approach and the deterministic approach. The short-time accuracy is measured by the relative L^2 norm between the true conditional mean and the conditional mean of the samples in our SDE approach or the single output given by the deterministic network, at a lag $\tau = 0.5$. For SDEs, we also have the prediction of the conditional std. For the long-time comparisons, we iterate the SDE and the deterministic map both for 100 steps and use the trajectories to estimate the averaged total enstrophy of the invariant distribution and compare them against the truth computed using samples from the true simulation data.

$\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$. Moreover, by changing the diffusion coefficients from $\sigma_s dW_s$ to the optimal $g_s^{\mathrm{F}} dW_s$ does not influence the spectrums significantly in this example.

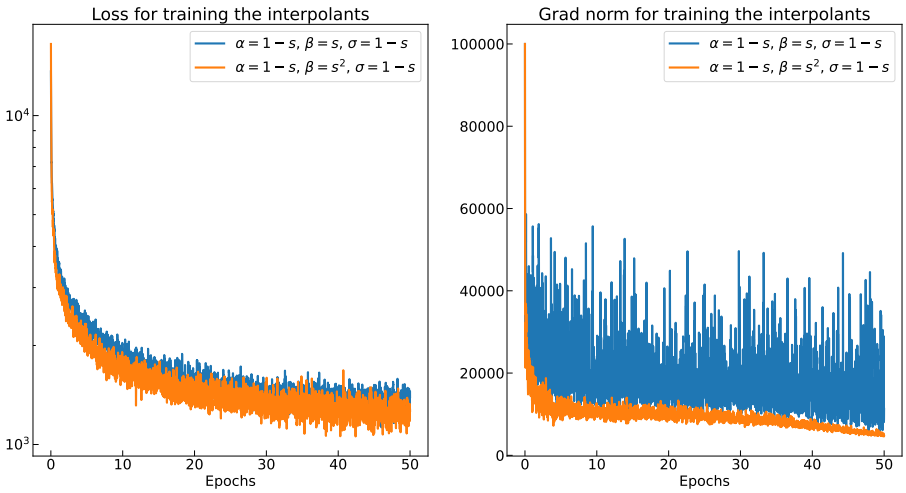

Figure 8. The training loss on the stochastic Navier-Stokes experiments and the norm of gradients of parameters during the training for stochastic interpolants with $\beta_s = s$ or $\beta_s = s^2$. We observe that the choice $\beta_s = s^2$ makes the gradient norm more stable.

KL divergence comparisons between $\sigma_s dW_s$, $g_s^{\mathrm{F}} dW_s$, and Gaussian base ODE We fix one initial vorticity field and compare the KL divergence between the true conditional distribution and the generated distribution. Here the KL divergence is calculated for the 1D conditional distributions of total enstrophy and energy, for a fixed ω_t and $\tau = 1$. The goal here is to test whether changing the diffusion coefficient from σ_s to g_s^{F} could improve the KL divergence accuracy of the generated distribution at $s = 1$. Note that our theory in Theorem 3.3 shows that changing from σ_s to g_s^{F} could improve the path KL divergence, which is an upper bound on the KL divergence of the generated marginal distribution at $s = 1$.

We consider the generative SDEs corresponding to the stochastic interpolant with $\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$. We can vary the diffusion coefficient of the generative SDEs; we choose $\sigma_s = 1 - s$, or $g_s^{\mathrm{F}} = \sqrt{(3-s)(1-s)}$ that corresponds to a Föllmer process. We also compare the results with the Gaussian base ODE generative model (equivalent to flow matching (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022), corresponding to $\alpha_s = 1 - s, \beta_s = s$, and x_0 in the interpolant obeys an $\mathsf{N}(0, Id)$ distribution). For this Gaussian base ODE, we use the same UNet architecture described in the beginning of Appendix C.2; again, the conditioning in the vector field is achieved by channel concatenation in the input. The quantitative KL results are reported in Table 3 and the densities are shown in Figure 10. We observe that SDE approaches lead to better KL accuracy. Moreover, changing the diffusion coefficients could potentially improve the KL accuracy, justifying the flexibility and usefulness of the interpolant approach.

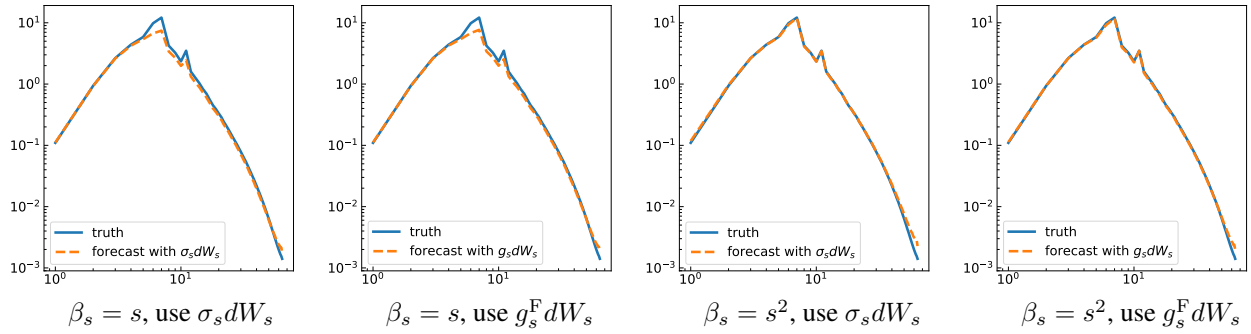

Figure 9. The enstrophy spectra of obtained samples from different generative models, compared to the truth, for interpolants $\alpha_s = 1 - s, \beta_s = s, \sigma_s = 1 - s$ and interpolants $\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$. The generative SDE is chosen such that the diffusion term are $\sigma_s dW_s$ or $g_s^{\mathrm{F}} dW_s$. The spectrum is computed by averaging over 50 different ω_t (independently drawn from the invariant distribution) and 300 independent ensembles of $\omega_{t+\tau}$ (simulation data) or $\hat{\omega}_{t+\tau}$ (forecast results). All models are trained for lag $\tau = 1$, with the same 2M-parameter Unet, 200K data, and the same training procedures as outlined in the beginning of this section. We observe that using $\beta_s = s^2$ leads to better performance in terms of the spectrums; this indicates that the stable gradient norm for $\beta_s = s^2$ (see Figure 8) is important in the generation quality. Changing g_s from σ_s to the optimal g_s^{F} does not influence the spectrum significantly.

KL: truth versus generation	density of total enstrophy	density of total energy
SDE with $\sigma_s dW_s$	8.49e-3±1.57e-3	**4.01e-3±8.95e-4**
SDE with $g_s^{\mathrm{F}} dW_s$	**2.79e-3±9.19e-4**	7.21e-3±1.58e-3
Gaussian base ODE	3.63e-3±9.63e-4	2.17e-2±2.50e-3

Table 3. Computed KL divergence between several 1D distributions of generated conditional samples and the true simulated samples. The 1D distributions are regarding total enstrophy and total energy of $\omega_{t+\tau}$, given a fixed ω_t and $\tau = 1$. We first use kernel density estimation based on 6000 ensembles to compute the density, and then compute the KL divergence with respect to the truth. Here we compare between the generated samples via SDEs with $\sigma_s dW_s$, via SDE with $g_s^{\mathrm{F}} dW_s$ which corresponds to a Föllmer process, and via ODEs with Gaussian base (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022). Here to construct the SDEs, we use the interpolant $\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$. In every column, the smallest value is highlighted in bold. The numbers are presented in the format **mean±std** where we do bootstrap to resample 6000 ensembles with replacement and get different KL results and report the **mean and std**; this std provides useful information of the sensitivity of KL with respect to resampling. We observe SDE performs better in terms of KL.

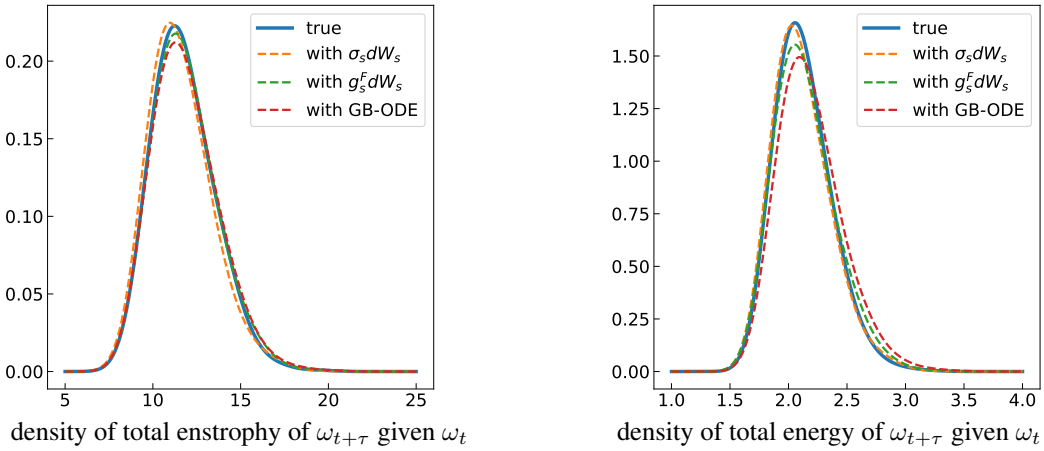

Figure 10. The 1D conditional distributions of total enstrophy and total energy of $\omega_{t+\tau}$, given a fixed initial vorticity field ω_t and $\tau = 1$. Here we compare between the truth, generated samples via SDEs with $\sigma_s dW_s$, via SDE with $g_s^{\mathrm{F}} dW_s$ which corresponds to a Föllmer process, and via ODEs with Gaussian base (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022). We use kernel density estimation based on 6000 ensembles to compute the density. Here to construct the SDEs, we use the interpolant $\alpha_s = 1 - s, \beta_s = s^2, \sigma_s = 1 - s$. Quantitative KL divergence between these distributions with truth are reported in Table 3. We observe that SDEs lead to better density quality compared to ODEs. For the enstrophy, $g_s^{\mathrm{F}} dW_s$ behaves better while for the energy, $\sigma_s dW_s$ works better, for this fixed ω_t. We leave as a future work to study the effect of the Föllmer g_s^{F} in more details.

C.3. Video generation

Algorithm 3 VideoEM: (Euler-Marayuma in Latent Space).

Input: Model $\hat{b}_s(y, y_0; y^t, t)$; previous frames $x^{1,\dots,t-1}$; grid $s_0 = 0 < s_1 \cdots < s_N = 1$ with $N \in \mathbb{N}$; *iid* $\eta_n \sim \mathsf{N}(0, Id)$ for $n = 0 : N-1$.
Set $\Delta s_n = s_{n+1} - s_n$ for $n = 0 : N-1$.
Set $Y_0 = \texttt{Encode}(x^{t-1})$
for $n = 0 : N-1$ **do**
  Draw $j \in \texttt{Unif}(2, \dots, t-1)$
  $y^{t-j} = \texttt{Encode}(x^{t-j})$
  $Y_{n+1} = Y_n + \hat{b}_{s_n}(Y_n, Y_0; y^{t-j}, t-j)\Delta s_n + \sigma_{s_n}\sqrt{\Delta s_n}\eta_n$
end for
$\hat{x}^t = \texttt{Decode}(Y_N)$
Return: Generated frame $\hat{x}^t$

Training. We train models for 250k gradient steps using AdamW starting at a learning rate of 2e-4. We use the UNet architecture popularized[6] in Ho et al. (2020). We modify the architecture to condition on past frames by concatenating them along the channel dimension of the input. Each model is trained on four A100 GPUs for approximately 1-2 days.

[6] We use the lucidrains repository.