# CONTINUOUS-TIME TRAJECTORY GENERATION FROM DISCRETE OBSERVATIONS WITH STOCHASTICITY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Physical systems evolve continuously in time, yet their states are typically observed only at discrete times. Generating trajectories consistent with their probability densities from such observations therefore requires capturing the continuous-time evolution rather than only learning transition mappings between consecutive observations. We propose **PhiBE-Flow**, a framework that directly estimates the probability velocity field induced by the stochastic differential equation (SDE) which governs this continuous-time distributional evolution. PhiBE-Flow learns from discrete observations using a model-free approach requiring neither known SDE coefficients nor score estimation. We establish convergence guarantees for the method, accounting for both time-discretization and finite-sample errors. We evaluate PhiBE-Flow on systems of increasing complexity, from controlled stochastic numerical systems to Navier–Stokes dynamics and real-world videos. Our results show that PhiBE-Flow accurately recovers probability flows of stochastic dynamics, preserves multiscale physical statistics, and improves video generation performance over representative baselines.

## 1 INTRODUCTION

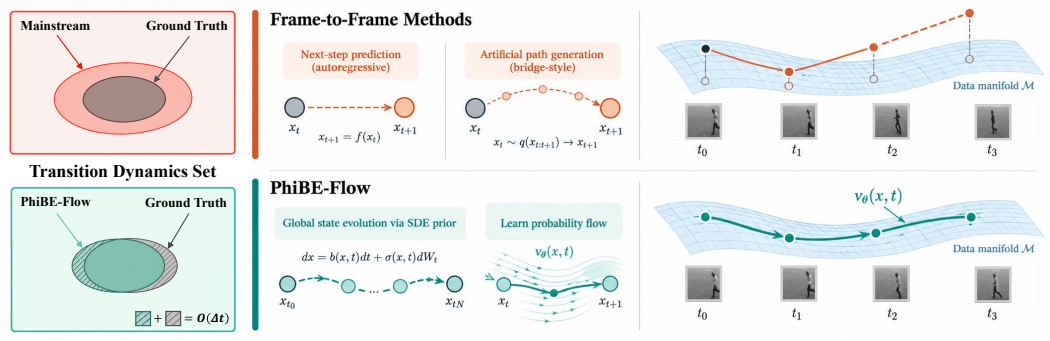



Figure 1: **From discrete transitions to continuous-time dynamics.** (Top) Mainstream frame-to-frame models may contain the ground-truth transitions, but fitting discrete transitions alone does not enforce consistency with continuous-time distributional evolution, leading to off-manifold drift. (Bottom) PhiBE-Flow directly learns the SDE-induced probability velocity to generate trajectories consistent with the evolving data distribution, with an $\mathcal{O}(\Delta t)$ overall velocity error.

Real-world physical systems are typically observed through discrete measurements. Learning their dynamics has attracted growing interest across applications in mechanics (Greydanus et al., 2019; Lutter et al., 2019), fluid dynamics (Brunton et al., 2020; Kochkov et al., 2021; Buaria & Sreenivasan, 2023), and video modeling (Oprea et al., 2022; Finn et al., 2016; Lee et al., 2018). Their underlying dynamics, however, often evolve continuously in physical time and may be stochastic or chaotic. Learning these systems therefore calls for capturing their continuous-time evolution beyond modeling transition mappings between sampled states. This is particularly important for video,



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

where visual realism at the frame level does not necessarily imply coherent motion or consistency with the underlying physical dynamics (Bansal et al., 2026; Gu et al., 2026; Motamed et al., 2026).

Recent advances in generative modeling have opened new avenues for learning dynamical systems from discrete observations (Ho et al., 2022; Lippe et al., 2023; Price et al., 2025). Autoregressive models learn conditional transitions between successive states or tokens, using architectures such as recurrent neural networks (Ranzato et al., 2014; Vlachas et al., 2018) and Transformers (Geneva & Zabaras, 2022; Yan et al., 2021). Bridge-based methods instead introduce an auxiliary path between neighboring states (Albergo et al., 2025), with diffusion models providing a prominent example through an artificial noising–denoising process (Ho et al., 2020; Song et al., 2021). Despite their success, these approaches do not explicitly model the underlying continuous-time evolution in physical time. Neural ordinary differential equations (ODEs) and stochastic differential equations (SDEs) address this limitation by learning continuous-time dynamics (Li et al., 2020; Djeumou et al., 2023). For stochastic systems, however, the underlying drift and diffusion are generally unknown, and the corresponding probability velocity additionally depends on the unknown time-dependent density score. This raises a central question: Can we learn this velocity directly from discrete observations without separately estimating these quantities?

We propose **PhiBE-Flow**, a model-free framework for continuous-time trajectory generation from discrete observations. Using PhiBE theory (Zhu, 2024), we approximate local drift and diffusion information from short-time increments. Combined with the Fokker–Planck equation, these approximations yield a score-free variational objective for learning the probability velocity field induced by the SDE governing the system's stochastic dynamics. With this field, we generate deterministic trajectories whose time-dependent marginal distributions approximate those of the underlying stochastic system. We evaluate PhiBE-Flow on the Ornstein–Uhlenbeck (OU) process, double-pendulum dynamics (Sutton & Barto, 1998), stochastic Navier–Stokes dynamics (Chen et al., 2024), and KTH videos (Schuldt et al., 2004). These experiments demonstrate its ability to recover probability flows, faithfully capture the nonlinear motion in stochastic dynamics, and preserve multiscale physical statistics across systems of increasing complexity. On both the Navier–Stokes and KTH video datasets, PhiBE-Flow outperforms representative diffusion-based and stochastic-transport baselines.

Our main contributions are:

- We introduce **PhiBE-Flow**, a model-free framework that learns SDE-induced probability velocity fields from discrete observations for continuous-time trajectory generation.
- We derive a score-free variational objective without assuming known SDE coefficients, with convergence guarantees accounting for time-discretization and finite-sample errors.
- We demonstrate the effectiveness of PhiBE-Flow on stochastic dynamical systems, showing improved generation accuracy, physical consistency, and video prediction metrics over representative baselines on the Navier–Stokes and KTH video datasets.

## 2 Related Work

### 2.1 Autoregressive Generation

Autoregressive methods factorize the conditional distribution of a future sequence and generate predictions one step at a time. Recurrent models propagate a hidden state and directly predict the next state or frame (Mikolov et al., 2010; Sutskever et al., 2014; Shi et al., 2015). Transformer-based models use self-attention to capture longer-range dependencies and often represent dynamical systems as sequences of discrete tokens or latent variables for next-token prediction (Vaswani et al., 2017; Dai et al., 2019). These models support recursive generation over arbitrary horizons and have achieved strong performance across physical and visual systems (Srivastava et al., 2015; Yan et al., 2021; Geneva & Zabaras, 2022). However, they learn transitions at discrete observation times, and do not explicitly model the continuous-time dynamics underlying the observations.

### 2.2 Bridge-Based Generation

Bridge-based methods replace direct prediction with an auxiliary process that transports a source distribution to the target distribution. Diffusion models use a fixed forward process to gradually

corrupt data into noise, while a learned reverse process generates the target, optionally conditioned on past observations (Sohl-Dickstein et al., 2015; Song et al., 2021; Ho et al., 2022; Voleti et al., 2022). Flow Matching directly learns continuous-time vector fields by regressing against velocities associated with prescribed conditional probability paths (Lipman et al., 2023; Tong et al., 2024). Stochastic interpolants construct deterministic or stochastic paths between arbitrary endpoint distributions (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025), including Schrödinger bridges (De Bortoli et al., 2021; Chen et al., 2022) and Föllmer processes (Chen et al., 2024). Although effective for modeling forecast uncertainty, these auxiliary dynamics need not coincide with the physical-time evolution of the observed system.

### 2.3 Learning Stochastic Dynamics

A more direct approach to dynamical learning is to model temporal evolution through SDEs, whose drift and diffusion coefficients are often unknown in data-driven settings. Neural (Tzen & Raginsky, 2019; Liu et al., 2020; Kidger et al., 2021) and latent SDE methods (Hasan et al., 2022; Zeng et al., 2023; Daems et al., 2024) learn these components using neural networks, and stochastic adjoint (Li et al., 2020) and simulation-free methods (Course & Nair, 2023; Bartosh et al., 2025) have been developed to improve their training efficiency. Related physics-informed approaches infer the coefficients through Fokker–Planck equations (Chen et al., 2021). Beyond trajectory data, SDE dynamics have also been learned from unpaired population snapshots via Wasserstein-based matching (Hashimoto et al., 2016; Ma et al., 2021), while drift and diffusion can be identified from a single trajectory via signature-based sparse regression (Ali et al., 2025). However, explicitly learning SDE dynamics can still pose scalability challenges in high-dimensional systems.

PhiBE offers an alternative perspective by incorporating local transition information into a continuous-time PDE formulation without first fitting a complete SDE model. Building on this perspective, we retain the assumption of underlying SDE dynamics but seek a probability flow with the same time-marginal densities, without explicitly fitting drift and diffusion or numerically integrating SDEs during training. This motivates our model-free, score-free variational formulation for learning continuous-time stochastic systems from discrete observations.

## 3 Method

### 3.1 Problem Setting

Consider a stochastic dynamical system governed by the Itô SDE

$$dX_t = b(X_t)\,dt + \sigma(X_t)\,dB_t, \qquad X_0 \sim \mu_0, \tag{1}$$

where $X_t \in \mathbb{R}^d$, while the drift field $b : \mathbb{R}^d \to \mathbb{R}^d$ and the diffusion loading $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times r}$ are unknown. We write $\Sigma(x) := \sigma(x)\sigma(x)^\top$ for the diffusion covariance.

We assume that the continuous trajectory is not directly accessible.

Instead, we observe $N$ independent trajectories at discrete times $0 = t_0 < t_1 < \cdots < t_{N_t} \le T$,

$$\mathcal{D} = \left\{ x_{t_i}^{(k)} \right\}_{k=1,\dots,N;\ i=0,\dots,N_t}. \tag{2}$$

Let $\mu_t = \mathrm{Law}(X_t)$ denote the marginal distribution of the stochastic process at physical time $t$. Our goal is to learn the evolution of $\mu_t$ from the discrete observations without separately identifying the complete drift and diffusion functions. In particular, we seek a deterministic vector field

$$v_\theta : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d, \tag{3}$$

with parameter $\theta \in \mathbb{R}^m$. Then we can sample points at intermediate times by evolving the ODE

$$\frac{dX_t}{dt} = v_\theta(X_t, t), \qquad X_0 \sim \mu_0. \tag{4}$$

This deterministic dynamics is known as the *probability flow* associated with the marginal evolution of the observed stochastic system.

### 3.2 PROBABILITY-FLOW FORMULATION

Let $\rho(x,t)$ denote the probability density of $X_t$. The marginal evolution of the SDE in equation 1 is governed by the Fokker–Planck equation

$$\partial_t \rho(x,t) = -\nabla_x \cdot \big(b(x)\rho(x,t)\big) + \frac{1}{2}\sum_{i,j=1}^{d} \partial_{x_i}\partial_{x_j}\big(\Sigma_{ij}(x)\rho(x,t)\big). \tag{5}$$

Using $\nabla_x \rho(x,t) = \rho(x,t)\nabla_x \log \rho(x,t)$, equation 5 can be written as the continuity equation

$$\partial_t \rho(x,t) + \nabla_x \cdot \big(\rho(x,t)v(x,t)\big) = 0, \tag{6}$$

where

$$v(x,t) = b(x) - \frac{1}{2}\Sigma(x)\nabla_x \log \rho(x,t) - \frac{1}{2}\nabla_x \cdot \Sigma(x). \tag{7}$$

Thus, the marginal evolution of the stochastic system can be represented by a deterministic probability velocity $v$. This motivates learning $v_\theta$ by minimizing its population discrepancy from $v$:

$$\min_{\theta\in\mathbb{R}^m} \int_0^T \mathbb{E}_{X_t\sim\rho(\cdot,t)}\left[\|v_\theta(X_t,t) - v(X_t,t)\|^2\right] dt. \tag{8}$$

Directly evaluating equation 8 is infeasible because the density score $\nabla_x \log \rho(x,t)$ is unknown. Expanding the squared objective, discarding terms independent of $\theta$, and applying integration by parts to the remaining score-dependent term, we obtain

$$\min_{\theta\in\mathbb{R}^m} \int_0^T \mathbb{E}_{X_t\sim\rho(\cdot,t)}\left[\|v_\theta(X_t,t)\|^2 - 2v_\theta(X_t,t)\cdot b(X_t) - \Sigma(X_t) : \nabla_x v_\theta(X_t,t)\right] dt. \tag{9}$$

While the density score is eliminated, the objective still depends on the unknown drift $b$ and diffusion covariance $\Sigma$, and therefore cannot yet be evaluated from the discrete observations alone.

### 3.3 PHIBE-FLOW OBJECTIVE

The score-free objective in equation 9 still requires the unknown drift $b$ and diffusion covariance $\Sigma$. To make the objective accessible from discrete observations, we leverage the local approximations provided by the PhiBE. For a sufficiently small sampling interval $\Delta t$, the drift and diffusion terms are approximated from short-time transitions by

$$\widehat{b}_{\Delta t}(x) = \frac{1}{\Delta t}\mathbb{E}\left[X_{t+\Delta t} - X_t \mid X_t = x\right], \tag{10}$$

and

$$\widehat{\Sigma}_{\Delta t}(x) = \frac{1}{\Delta t}\mathbb{E}\left[(X_{t+\Delta t} - X_t)(X_{t+\Delta t} - X_t)^\top \mid X_t = x\right]. \tag{11}$$

Substituting these approximations into equation 9 yields the PhiBE-Flow population objective

$$\mathcal{J}_{\Delta t}(\theta) = \int_0^T \mathbb{E}_{X_t\sim\rho(\cdot,t)}\Big[\|v_\theta(X_t,t)\|^2 - 2v_\theta(X_t,t)\cdot \widehat{b}_{\Delta t}(X_t) - \widehat{\Sigma}_{\Delta t}(X_t) : \nabla_x v_\theta(X_t,t)\Big] dt. \tag{12}$$

In practice, the conditional expectations are replaced by the observed short-time increments. For each transition, define

$$\Delta x_i^{(k)} = x_{t_{i+1}}^{(k)} - x_{t_i}^{(k)}, \qquad \Delta t_i = t_{i+1} - t_i. \tag{13}$$

The resulting empirical PhiBE-Flow objective is

$$\widehat{\mathcal{J}}(\theta) = \frac{1}{NN_t}\sum_{i=0}^{N_t-1}\sum_{k=1}^{N}\left[\|v_\theta(x_{t_i}^{(k)},t_i)\|^2 - 2v_\theta(x_{t_i}^{(k)},t_i)\cdot\frac{\Delta x_i^{(k)}}{\Delta t_i} - \frac{1}{\Delta t_i}\Delta x_i^{(k)\top}\nabla_x v_\theta(x_{t_i}^{(k)},t_i)\Delta x_i^{(k)}\right]. \tag{14}$$

After solving this optimization problem, given a new initial sample $X_0 \sim \mu_0$, we can generate future samples by solving

$$\dot{X}_t = v_\theta(X_t,t), \qquad X_t = X_0 + \int_0^t v_\theta(X_s,s)\,ds.$$

The covariance-weighted Jacobian term can be costly in high dimensions. Some systems admit constant isotropic diffusion, and this term simplifies to $\sigma^2\nabla_x \cdot v_\theta$. See Appendix A.4 for details.

# 4 THEORETICAL GUARANTEES

We establish convergence guarantees for PhiBE-Flow by separating two sources of error: the population approximation error induced by discrete observations and the finite-sample error arising from finitely many transitions.

The finite-sample error is for fixed linear-basis models. Population bounds are evaluated on a compact set $\Omega \subset \mathbb{R}^d$; statistical bounds use the observation-time sampling measure $\mu_L$ in Appendix A.6.

## 4.1 POPULATION APPROXIMATION

We first state the first-order consistency of the PhiBE coefficients.

**Lemma 1** (PhiBE coefficient approximation)**.** *Suppose the population conditions in Appendix A.1 hold. Let $\widehat{b}$ and $\widehat{\Sigma}$ denote the first-order PhiBE estimates of the drift $b(x)$ and the diffusion covariance $\Sigma(x)$, respectively. Then there exist constants $L_b, L_\Sigma, L_{\mathrm{div}} > 0$, independent of $\Delta t$, such that*

$$\|b - \widehat{b}\|_\infty \le L_b \Delta t, \tag{15}$$

$$\|\Sigma - \widehat{\Sigma}\|_\infty \le L_\Sigma \Delta t + o(\Delta t), \tag{16}$$

*and*

$$\|\nabla \cdot (\Sigma - \widehat{\Sigma})\|_\infty \le L_{\mathrm{div}} \Delta t + o(\Delta t). \tag{17}$$

The corresponding error in the induced probability velocity is characterized as follows.

**Theorem 1** (General population approximation)**.** *Suppose the population conditions in Appendix A.1 hold, and let $\widehat{b}$ and $\widehat{\Sigma}$ be the first-order PhiBE coefficients in Lemma 1. Then there exist constants $L_v, K > 0$, independent of $\Delta t$, such that*

$$\|v(x,t) - \widehat{v}(x,t)\| \le L_v \Delta t + o(\Delta t), \tag{18}$$

*and*

$$\|x_t - \widehat{x}_t\| \le \frac{e^{Kt} - 1}{K} L_v \Delta t + o(\Delta t). \tag{19}$$

The proof is provided in Appendices A.2 and A.3. For the OU process, the approximate probability velocity and the associated error bounds admit explicit expressions; see Appendix A.5.

## 4.2 FINITE-SAMPLE CONVERGENCE

We give a linear-basis proof for $\mathcal{V}_p = \{v_\theta = \sum_{\ell=1}^p \theta_\ell \psi_\ell : \theta \in \mathbb{R}^p\}$, with basis fields fixed independently of the fitting sample. The basis fields may be nonlinear in $(x,t)$; only their coefficients are fitted. Here $N$ counts independent trajectories, while transitions within each trajectory may be dependent. The sampling measure $\mu_L$ and estimator $v_N$ are defined in Appendix A.6.

**Theorem 2** (Finite-sample convergence for linear-basis models)**.** *Under Assumption 1, including $\widehat{v} \in \mathcal{V}_p$, there exist constants $c_0, C_{\mathcal{V}} > 0$, independent of $N, \Delta t, \delta$, such that, for $\Delta t \in (0,1]$ and $\delta \in (0,1)$, if*

$$N \ge c_0 p \log(2p/\delta),$$

*then, with probability at least $1 - \delta$,*

$$\|v_N - \widehat{v}\|_{L^2(\mu_L)} \le C_{\mathcal{V}} \sqrt{\frac{p \log(2p/\delta)}{N \Delta t}}. \tag{20}$$

The proof is in Appendix A.6. Appendix A.7 gives a separate affine OU bound relative to the population minimizer within the affine class.

**Remark.** Many neural networks reduce to a linear-basis model by removing the output activation, freezing the hidden features, and training only the final affine layer. The theorem applies under its assumptions when the feature map is fixed independently of the fitting trajectories. If $\widehat{v} \notin \mathcal{V}_p$, an approximation term is also required; see Appendix A.6.

### 4.3 Overall Convergence

For a compact evaluation set $\Omega$, write

$$\|f\|^2_{L^2(\mu_L;\Omega)} := \frac{1}{L}\sum_{j=0}^{L-1}\int_\Omega \|f(x,t_j)\|^2 \rho_{t_j}(x)\,dx.$$

The statistical and population errors satisfy

$$\|v_N - v\|_{L^2(\mu_L;\Omega)} \le \underbrace{\|v_N - \widehat{v}\|_{L^2(\mu_L)}}_{\text{finite-sample error}} + \underbrace{\|\widehat{v} - v\|_{L^2(\mu_L;\Omega)}}_{\text{PhiBE approximation error}}. \tag{21}$$

**Corollary 1** (Overall convergence of PhiBE-Flow)**.** *Under the assumptions of Theorems 1 and 2, including $N \ge c_0 p\log(2p/\delta)$, with probability at least $1-\delta$,*

$$\|v_N - v\|_{L^2(\mu_L;\Omega)} \le L_v\Delta t + C_{\mathcal{V}}\sqrt{\frac{p\log(2p/\delta)}{N\Delta t}} + o(\Delta t).$$

The proof is in Appendix A.8. This bound concerns velocity error at the observation times; it does not by itself give a trajectory bound for the learned field.

## 5 Experiments

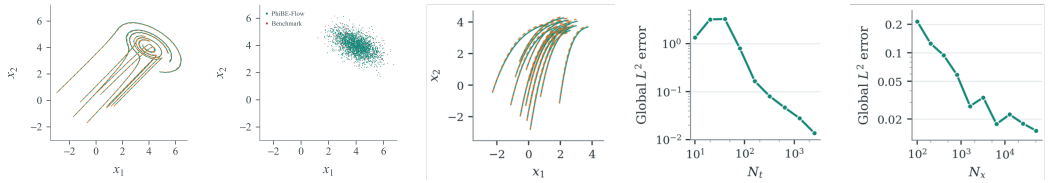
We evaluate PhiBE-Flow on four problems of increasing complexity: the Ornstein–Uhlenbeck process, the double pendulum system, stochastic Navier–Stokes dynamics, and KTH video generation. For the Navier–Stokes and KTH benchmarks, we compare against MCVD (Voleti et al., 2022) and PFI (Chen et al., 2024). Additional details are provided in Appendix B.

### 5.1 Learning from Stochastic Dynamical Systems

**Ornstein–Uhlenbeck Process** Consider a two-dimensional OU process with constant diffusion,

$$dX_t = \Gamma(X_t - \mu)\,dt + \sigma\,dB_t, \qquad X_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \tag{22}$$

where $\Gamma \in \mathbb{R}^{2\times 2}$, $\sigma \in \mathbb{R}^{2\times 2}$, $\mu \in \mathbb{R}^2$. The OU process is particularly useful for validating PhiBE-Flow because its marginal distributions remain Gaussian and its probability velocity admits a closed-form affine expression. We generate trajectories from equation 22 and use pairs of consecutive observations at discrete time points to learn the spatiotemporal probability velocity field $v_\theta(x,t)$.




(a) Probability-flow (Non-reversible). (b) Samples $t = 15$. (Non-reversible) (c) Probability-flow (Reversible). (d) Error versus $N_t$. (Reversible) (e) Error versus $N$. (Reversible)

Figure 2: **PhiBE-Flow on reversible and non-reversible OU processes.** (a) PhiBE-Flow (green) and analytical (orange) probability-flow trajectories for the non-reversible case. (b) PhiBE-Flow (blue) and OU-process (red) samples at $t = 15$. (c) PhiBE-Flow (green) and analytical (orange) trajectories for the reversible case. (d) Global $L^2$ velocity error versus temporal observations $N_t$. (e) Global $L^2$ velocity error versus observed trajectories $N$.

We first evaluate PhiBE-Flow on a non-reversible OU process whose drift and diffusion matrices, $\Gamma$ and $\sigma$, do not commute. At each observation time $t_i$, we use the affine parameterization $v_\theta(x,t_i) = A_i x + b_i$, which exactly represents the analytical probability velocity. Despite convergence of the marginal distribution to a stationary Gaussian (Figure 2(b)), the process retains a nonzero steady-state probability current due to broken detailed balance (Godrèche & Luck, 2019;

Sekizawa et al., 2024). PhiBE-Flow accurately reproduces these circulating probability-flow trajectories (Figure 2(a)).

We next consider a reversible OU process using a multilayer perceptron (MLP) parameterization. PhiBE-Flow closely recovers the analytical probability velocity and trajectories (Figure 2(c)). Figures 2(d) and 2(e) further show global $L^2$ velocity errors decaying approximately as $O(N_t^{-1})$ and $O(N^{-1/2})$, respectively. While the finite-sample bound does not directly cover the jointly trained MLP, these trends are consistent with the population time-discretization and affine finite-sample scalings in Corollaries 2 and 3.

We refer readers to Appendix B.1 for further details on data generation and the implementation of the algorithm.

**Double Pendulum System** We next consider the nonlinear Acrobot system, consisting of two serially connected rigid links. Following the standard Acrobot dynamics, the state is represented as

$$X_t = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)_t,$$

where $\theta_1$ and $\theta_2$ denote the two joint angles. We generate the dataset by numerically simulating the corresponding equations of motion and introduce additional stochastic perturbations to the angular accelerations,

$$\begin{aligned} d\theta_1 &= \dot{\theta}_1\, dt, \qquad d\theta_2 = \dot{\theta}_2\, dt, \\ d\dot{\theta}_1 &= f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)\, dt + \sigma_1\, dB_t^{(1)}, \\ d\dot{\theta}_2 &= f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)\, dt + \sigma_2\, dB_t^{(2)}, \end{aligned} \tag{23}$$

where $f_1$ and $f_2$ are determined by the Acrobot equations of motion and $\sigma_1, \sigma_2$ control the strength of the artificial noise. For the state-space experiment, we simulate 1000 trajectories, each lasting 8 seconds at 30 frames per second, resulting in 240 observations per trajectory. Initial joint angles are sampled uniformly from $[-\pi, \pi]$, while the initial angular velocities are sampled from $[-0.1, 0.1]$. In both experiments, we use an 80/20 training–test split. Additional details of the physical parameters, numerical simulation, and training procedure are provided in Appendix B.2.

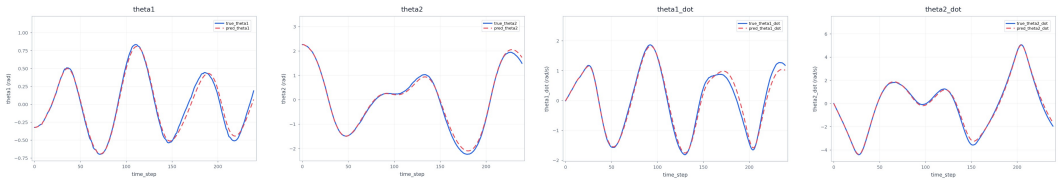



Figure 3: **State-space learning on the stochastic double-pendulum system.** Ground-truth and PhiBE-Flow predictions for two joint angles $\theta_1, \theta_2$ and angular velocities $\dot{\theta}_1, \dot{\theta}_2$ over physical time.

We first apply PhiBE-Flow directly in the state space and use a residual MLP to learn the probability velocity from consecutive observations. Figure 3 compares the predicted trajectories with the ground truth. PhiBE-Flow closely follows the nonlinear evolution of both joint angles and their corresponding angular velocities over the generation horizon.

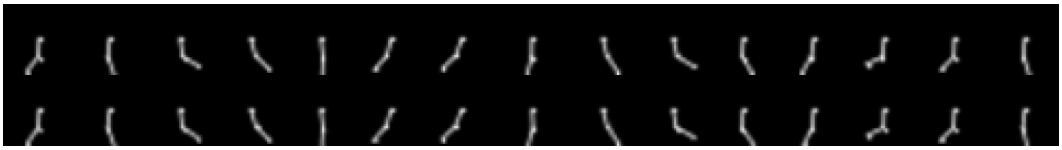


Figure 4: **Image-based Learning on the stochastic double-pendulum system.** The top row shows the ground-truth image sequence. The bottom row shows results from PhiBE-Flow.

We further construct an image-based learning task by rendering each pendulum state into a grayscale $28 \times 28$ image. A geometry-preserving encoder maps each frame to a three-dimensional latent

representation; the latent positions are then augmented with finite-difference velocities to form a six-dimensional dynamical state. PhiBE-Flow learns the probability velocity in this latent space, and the predicted latent states are decoded back to image space.

As shown in Figure 4, the predicted rollout closely follows the ground-truth motion while preserving the geometry of the two-link system. This experiment demonstrates that PhiBE-Flow can model the same stochastic physical dynamics both from explicit states and from learned visual representations.

### 5.2 2D Navier–Stokes Equations Generation

We next consider learning the stochastic two-dimensional Navier–Stokes equations on the torus in the vorticity formulation,

$$d\omega + \mathbf{v} \cdot \nabla \omega dt = \nu \Delta \omega dt - \alpha \omega dt + \epsilon d\eta, \tag{24}$$

where $\omega$ is the vorticity field, $\mathbf{v}$ is the corresponding velocity field, and $d\eta$ represents stochastic forcing.

We use the PFI benchmark (Chen et al., 2024), comprising 1,000 simulated trajectories and $1 \times 10^5$ single-channel vorticity snapshots on a $128 \times 128$ periodic grid, with an 80/20 training–test split. Given an observed vorticity field, the objective is to generate its future evolution over multiple time steps. PhiBE-Flow is compared against MCVD and PFI; additional dataset details are provided in Appendix B.3.

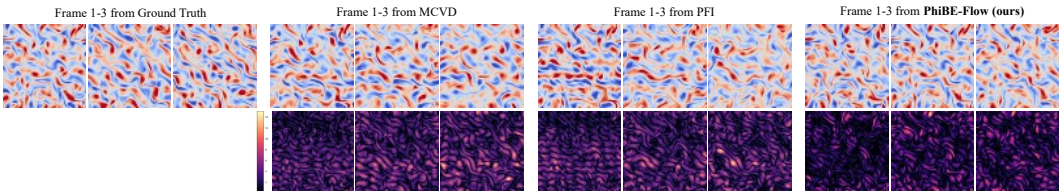



Figure 5: **Temporal generation results on Navier–Stokes.** Ground-truth vorticity fields and results from MCVD, PFI, and PhiBE-Flow, together with their corresponding error maps.

We evaluate Navier–Stokes generation results using the enstrophy spectrum, which can evaluate whether the results preserve the distribution of vorticity across spatial scales. For a vorticity field $\omega$, we compute its Fourier transform $\widehat{\omega}$ and define the spectral enstrophy as

$$E(\mathbf{k}) = \frac{1}{2} \left| \widehat{\omega}(\mathbf{k}) \right|^2 .$$

The spectrum is obtained by averaging $E(\mathbf{k})$ over radial wavenumber shells and subsequently averaging the resulting spectra over the evaluation samples. The same procedure is applied to the ground-truth fields and generation results from all compared methods.

As shown in Figure 5, PhiBE-Flow more accurately captures the evolution of the vorticity field and preserves fine-scale spatial structures over the generation horizon. Its error maps exhibit smaller deviations from the ground truth than those of MCVD and PFI.
The difference is also reflected in the enstrophy spectrum (Figure 6), which characterizes the distribution of vorticity across spatial scales. PhiBE-Flow remains closer to the ground-truth spectrum over a broad range of wavenumbers, including the high-frequency regime associated with small-scale flow structures. In contrast, MCVD exhibits a stronger loss of high-frequency energy, while PFI shows larger deviations at small scales. These results indicate that PhiBE-Flow better preserves both the generated flow structures and their multiscale physical statistics.

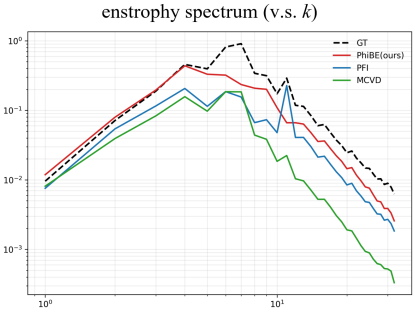



Figure 6: Enstrophy spectrum comparison on the Navier–Stokes dataset.

### 5.3 VIDEO GENERATION

Finally, we evaluate PhiBE-Flow on the KTH human action dataset (Schuldt et al., 2004), which contains 2,391 video sequences covering six action categories. We resize frames to $64 \times 64$ pixels and use an 80/20 training–test split. Following latent video generation approaches, video frames are encoded into a compact latent representation using a pretrained VQ-VAE, and PhiBE-Flow models their temporal evolution in the latent space. The predicted latent states are subsequently decoded back into video frames. We compare against MCVD and PFI and evaluate the results using mean squared error (MSE), Fréchet Video Distance (FVD) (Unterthiner et al., 2018), peak signal-to-noise ratio (PSNR), and structural similarity (SSIM) (Wang et al., 2004). Details of the representation and evaluation protocol are provided in Appendix B.4.

Table 1: **Metric results on KTH.** Standard metrics compare generated results with the original ground truth. Adjusted metrics compare generated results with the reconstructed ground truth.

| | Standard metrics | | | | Adjusted metrics | | | |
|---|---|---|---|---|---|---|---|---|
| *Method* | MSE($\times 10^{-2}$)↓ | FVD↓ | PSNR↑ | SSIM↑ | MSE($\times 10^{-2}$)↓ | FVD↓ | PSNR↑ | SSIM↑ |
| MCVD | 0.8550 | 307.3 | 20.68 | 0.6986 | – | – | – | – |
| PFI | 0.6637 | 280.1 | 21.78 | 0.7254 | 0.6576 | 247.1 | 21.82 | 0.7372 |
| **PhiBE-Flow** | **0.4405** | **227.5** | **23.56** | **0.7663** | **0.4305** | **192.7** | **23.66** | **0.7800** |

As shown in Table 1, PhiBE-Flow achieves the best performance across all four metrics. In particular, it substantially reduces both MSE and FVD while improving PSNR and SSIM compared with MCVD and PFI. The qualitative results in Figure 7 are consistent with the quantitative comparison. PhiBE-Flow better preserves the human-body structure and motion over the generation horizon and exhibits less accumulated motion drift than MCVD and PFI. These results suggest that directly learning the physical-time probability velocity provides a stable representation of temporal evolution even for high-dimensional visual observations.



Figure 7: **Video generation on the KTH dataset.** The top row shows the ground-truth trajectory. The following three rows show generation results from MCVD, PFI, and PhiBE-Flow, respectively.

## 6 CONCLUSION

We introduced **PhiBE-Flow**, a model-free framework that learns the SDE-induced probability velocity field from discrete observations to generate continuous-time trajectories consistent with density evolution. We established convergence guarantees and demonstrated its effectiveness on low-dimensional stochastic systems, Navier–Stokes dynamics, and video generation.

**Limitations and Future Work.** PhiBE-Flow assumes that the observed dynamics admit an underlying SDE representation, which may not hold for all real-world temporal processes. Moreover, directly learning the probability velocity in high-dimensional observation spaces can be computationally expensive, motivating the use of latent representations. The resulting performance therefore depends on the encoder–decoder, whose latent space should preserve the essential geometric and dynamical structure of the original data. Future work will extend the framework and its theoretical analysis to more general stochastic dynamics and more complex real-world systems.

### AI USE STATEMENT

We used generative AI tools to polish the language and presentation of the manuscript, organize existing code, and assist in developing mathematical proofs. The authors reviewed all AI-assisted content, manually searched the literature to verify the authenticity of AI-suggested references, and checked the mathematical arguments. We also performed smoke tests, including a small-scale overfitting test, to check that the reorganized code functioned as intended. The authors take full responsibility for the final manuscript, code, and mathematical claims, including all AI-assisted content.

### REPRODUCIBILITY STATEMENT

The theoretical assumptions and proofs are provided in Appendix A. Appendix B describes the experimental setups, model implementations, and evaluation metrics. The code and accompanying reproduction instructions are available in an anonymous repository at `https://anonymous.4open.science/r/PhiBE-Flow/`. The repository includes environment specifications, data preparation instructions, experiment configurations and random seeds, and commands for training, generation, and evaluation.

## REFERENCES


Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. Journal of Machine Learning Research, 2025. URL https://jmlr.org/papers/v26/23-1605.html.

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In International Conference on Learning Representations, 2023. URL https://arxiv.org/abs/2209.15571.

Munawar Ali, Purba Das, Qi Feng, Liyao Gao, and Guang Lin. Noise estimation of SDE from a single data trajectory. arXiv preprint arXiv:2509.25484, 2025.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. VideoPhy-2: A Challenging Action-Centric Physical Commonsense Evaluation in Video Generation. In International Conference on Learning Representations, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c02f6a1d5c55e16db50d339dad905b4d-Abstract-Conference.html.

Grigory Bartosh, Dmitry Vetrov, and Christian A. Naesseth. SDE Matching: Scalable and Simulation-Free Training of Latent Stochastic Differential Equations. In Proceedings of the 42nd International Conference on Machine Learning, 2025. URL https://proceedings.mlr.press/v267/bartosh25a.html.

Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. Annual Review of Fluid Mechanics, 2020. doi: 10.1146/annurev-fluid-010719-060214. URL https://doi.org/10.1146/annurev-fluid-010719-060214.

Dhawal Buaria and Katepalli R. Sreenivasan. Forecasting small-scale dynamics of fluid turbulence using deep neural networks. Proceedings of the National Academy of Sciences, 2023. doi: 10.1073/pnas.2305765120. URL https://doi.org/10.1073/pnas.2305765120.

Tianrong Chen, Guan-Horng Liu, and Evangelos A. Theodorou. Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. In International Conference on Learning Representations, 2022. URL https://arxiv.org/abs/2110.11291.

Xiaoli Chen, Liu Yang, Jinqiao Duan, and George Em Karniadakis. Solving Inverse Stochastic Problems from Discrete Particle Observations Using the Fokker–Planck Equation and Physics-Informed Neural Networks. SIAM Journal on Scientific Computing, 2021. doi: 10.1137/20M1360153. URL https://doi.org/10.1137/20M1360153.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael Samuel Albergo, Nicholas Matthew Boffi, and Eric Vanden-Eijnden. Probabilistic Forecasting with Stochastic Interpolants and Föllmer

Processes. In Proceedings of the 41st International Conference on Machine Learning, 2024. URL https://proceedings.mlr.press/v235/chen24n.html.

Kevin Course and Prasanth Nair. Amortized Reparametrization: Efficient and Scalable Variational Inference for Latent SDEs. In Advances in Neural Information Processing Systems, 2023. doi: 10.52202/075280-3423. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f72d4fdfd5eb425cd81df9fe6272a533-Abstract-Conference.html.

Rembert Daems, Manfred Opper, Guillaume Crevecoeur, and Tolga Birdal. Variational Inference for SDEs Driven by Fractional Noise. In International Conference on Learning Representations, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8aff4ffcf2a9d41692a805b3987e29ea-Abstract-Conference.html.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient Video Prediction via Sparsely Conditioned Flow Matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Davtyan_Efficient_Video_Prediction_via_Sparsely_Conditioned_Flow_Matching_ICCV_2023_paper.html.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In Advances in Neural Information Processing Systems, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/940392f5f32a7ade1cc201767cf83e31-Abstract.html.

Franck Djeumou, Cyrus Neary, and Ufuk Topcu. How to Learn and Generalize From Three Minutes of Data: Physics-Constrained and Uncertainty-Aware Neural Stochastic Differential Equations. In Proceedings of the 7th Conference on Robot Learning, 2023. URL https://proceedings.mlr.press/v229/djeumou23a.html.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.html.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. In Advances in Neural Information Processing Systems, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/d9d4f495e875a2e075a1a4a6e1b9770f-Abstract.html.

Nicholas Geneva and Nicholas Zabaras. Transformers for modeling physical systems. Neural Networks, 2022. doi: 10.1016/j.neunet.2021.11.022. URL https://doi.org/10.1016/j.neunet.2021.11.022.

Claude Godrèche and Jean-Marc Luck. Characterising the nonequilibrium stationary states of Ornstein–Uhlenbeck processes. Journal of Physics A: Mathematical and Theoretical, 2019. doi: 10.1088/1751-8121/aaf190. URL https://doi.org/10.1088/1751-8121/aaf190.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian Neural Networks. In Advances in Neural Information Processing Systems, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html.

Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, and Xin Wang. PhyWorldBench: A Comprehensive Evaluation of Physical Realism in Text-to-Video Models. In International Conference on Learning Representations, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/79d86433c2acd12b6fa98553435d226e-Abstract-Conference.html.

Ali Hasan, João M. Pereira, Sina Farsiu, and Vahid Tarokh. Identifying Latent Stochastic Differential Equations. IEEE Transactions on Signal Processing, 2022. doi: 10.1109/tsp.2021.3131723. URL https://doi.org/10.1109/tsp.2021.3131723.

Tatsunori Hashimoto, David Gifford, and Tommi Jaakkola. Learning population-level diffusions with generative RNNs. In Proceedings of the 33rd International Conference on Machine Learning, volume 48 of PMLR, pp. 2417–2426, 2016.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In Advances in Neural Information Processing Systems, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. In Advances in Neural Information Processing Systems, 2022. doi: 10.52202/068431-0628. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html.

Patrick Kidger, James Foster, Xuechen Li, and Terry J. Lyons. Neural SDEs as Infinite-Dimensional GANs. In Proceedings of the 38th International Conference on Machine Learning, 2021. URL https://proceedings.mlr.press/v139/kidger21b.html.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. Proceedings of the National Academy of Sciences, 2021. doi: 10.1073/pnas.2101784118. URL https://doi.org/10.1073/pnas.2101784118.

Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic Adversarial Video Prediction. arXiv preprint arXiv:1804.01523, 2018. URL https://arxiv.org/abs/1804.01523.

Wonjun Lee, Riley C. W. O'Neill, Dongmian Zou, Jeff Calder, and Gilad Lerman. Geometry-Preserving Encoder/Decoder in Latent Generative Models. arXiv preprint arXiv:2501.09876, 2025. URL https://arxiv.org/abs/2501.09876.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable Gradients for Stochastic Differential Equations. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 2020. URL https://proceedings.mlr.press/v108/li20i.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. In Advances in Neural Information Processing Systems, 2023. doi: 10.52202/075280-2946. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html.

Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. How Does Noise Help Robustness? Explanation and Exploration under the Neural SDE Framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. doi: 10.1109/cvpr42600.2020.00036. URL https://doi.org/10.1109/cvpr42600.2020.00036.

Michael Lutter, Christian Ritter, and Jan Peters. Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=BklHpjCqKm.

Shaojun Ma, Shu Liu, Hongyuan Zha, and Haomin Zhou. Learning stochastic behaviour from aggregate data. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of PMLR, pp. 7258–7267, 2021.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Proceedings of Interspeech 2010, 2010. doi: 10.21437/Interspeech.2010-343. URL https://www.isca-archive.org/interspeech_2010/mikolov10_interspeech.html.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do Generative Video Models Understand Physical Principles? In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2026. URL https://openaccess.thecvf.com/content/WACV2026/html/Motamed_Do_Generative_Video_Models_Understand_Physical_Principles_WACV_2026_paper.html.

Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A Review on Deep Learning Techniques for Video Prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. doi: 10.1109/tpami.2020.3045007. URL https://doi.org/10.1109/tpami.2020.3045007.

Grigorios A. Pavliotis. Stochastic Processes and Applications: Diffusion Processes, the Fokker–Planck and Langevin Equations Springer, 2014. doi: 10.1007/978-1-4939-1323-7. URL https://link.springer.com/book/10.1007/978-1-4939-1323-7.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. Nature, 2025. doi: 10.1038/s41586-024-08252-9. URL https://www.nature.com/articles/s41586-024-08252-9.

MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604, 2014. URL https://arxiv.org/abs/1412.6604.

Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. doi: 10.1109/icpr.2004.1334462. URL https://www.csc.kth.se/cvap/actions/.

Daiki Sekizawa, Sosuke Ito, and Masafumi Oizumi. Decomposing Thermodynamic Dissipation of Linear Langevin Systems via Oscillatory Modes and Its Application to Neural Dynamics. Physical Review X, 2024. doi: 10.1103/PhysRevX.14.041003. URL https://journals.aps.org/prx/abstract/10.1103/PhysRevX.14.041003.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Advances in Neural Information Processing Systems, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Proceedings of the 32nd International Conference on Machine Learning, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In International Conference on Learning Representations, 2021. URL https://arxiv.org/abs/2011.13456.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. In Proceedings of the 32nd International Conference on Machine Learning, 2015. URL https://proceedings.mlr.press/v37/srivastava15.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/5a18e133cbf9f257297f410bb7eca942-Abstract.html.

Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998. URL https://mitpress.mit.edu/9780262193986/reinforcement-learning/.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport. Transactions on Machine Learning Research, 2024. URL https://openreview.net/forum?id=CD9Snc73AW.

Belinda Tzen and Maxim Raginsky. Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit. arXiv preprint arXiv:1905.09883, 2019. URL https://arxiv.org/abs/1905.09883.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges. arXiv preprint arXiv:1812.01717, 2018. URL https://arxiv.org/abs/1812.01717.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Pantelis R. Vlachas, Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2018. doi: 10.1098/rspa.2017.0844. URL https://doi.org/10.1098/rspa.2017.0844.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. MCVD - Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In Advances in Neural Information Processing Systems, 2022. doi: 10.52202/068431-1698. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/944618542d80a63bbec16dfbd2bd689a-Abstract-Conference.html.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 2004. doi: 10.1109/tip.2003.819861. URL https://ece.uwaterloo.ca/~z70wang/research/ssim/.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video Generation using VQ-VAE and Transformers. arXiv preprint arXiv:2104.10157, 2021. URL https://arxiv.org/abs/2104.10157.

Sebastian Zeng, Florian Graf, and Roland Kwitt. Latent SDEs on Homogeneous Spaces. In Advances in Neural Information Processing Systems, 2023. doi: 10.52202/075280-3328. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f0172a5da5a2611e3dc0fe9c6e9a7480-Abstract-Conference.html.

Yuhua Zhu. PhiBE: A PDE-based Bellman Equation for Continuous Time Policy Evaluation. arXiv preprint arXiv:2405.12535, 2024. URL https://arxiv.org/abs/2405.12535.

# A Assumptions and Proofs

## A.1 Population Conditions and Objective Identity

All spatial integrals are over $\mathbb{R}^d$. Uniform population estimates are on $\mathcal{Z} = \Omega \times [0, T]$, where $\Omega$ is compact and convex. Vector norms are Euclidean, matrix norms are operator norms unless indicated otherwise, and $A : B$ denotes the Frobenius inner product. We use $(\nabla \cdot A)_i = \sum_j \partial_{x_j} A_{ij}$ and $(\nabla u)_{ij} = \partial_{x_j} u_i$.

Using the same marginal density $\rho_t$ as in equation 7, define the population PhiBE field

$$\widehat{v}(x,t) = \widehat{b}(x) - \frac{1}{2}\widehat{\Sigma}(x)\nabla_x \log \rho_t(x) - \frac{1}{2}\nabla_x \cdot \widehat{\Sigma}(x). \tag{25}$$

**Population conditions.** The SDE is nonexplosive with locally smooth coefficients and is defined through $T + h_0$ for some $h_0 > 0$. Its density $\rho_t$ is positive, and $\rho, \nabla_x \rho, \nabla_x^2 \rho$ are continuous near $\mathcal{Z}$. The fields $v$ and $\widehat{v}$ are locally Lipschitz in $x$, uniformly in $t \in [0, T]$ for each fixed $h$. The second-order semigroup expansions used in Appendix A.2 hold uniformly on $\Omega$, with $o(\Delta t^2)$ remainders, including one spatial derivative of the increment second moment. The coefficient functions appearing in these expansions and their stated derivatives are continuous near $\Omega$. These are assumptions on the short-time expansions; local smoothness alone is not used to claim uniform remainder bounds.

**Integrability and integration by parts.** For each relevant $t$, assume $\widehat{b}, \widehat{\Sigma}, \widehat{v} \in L^2(\rho_t)$ and $\rho_t, \widehat{\Sigma} \in C^1_{\mathrm{loc}}(\mathbb{R}^d)$. Admissible fields $u \in C^1_{\mathrm{loc}}$ satisfy $u \in L^2(\rho_t)$ and $\widehat{\Sigma} : \nabla u \in L^1(\rho_t)$. Assume the analogous integrability for $b, \Sigma, v$ when using the exact score-free objective, and time integrability when integrating these identities over $[0, T]$. Weak spatial derivatives are also allowed whenever the following integration-by-parts identity remains valid.

For a fixed $t$, define

$$\mathcal{J}_{h,t}(u) = \int_{\mathbb{R}^d} \big[\|u\|^2 - 2u \cdot \widehat{b} - \widehat{\Sigma} : \nabla u\big] \rho_t \, dx.$$

Integration by parts gives

$$-\int (\widehat{\Sigma} : \nabla u)\rho_t \, dx = \int u \cdot (\widehat{\Sigma}\nabla \log \rho_t + \nabla \cdot \widehat{\Sigma})\rho_t \, dx.$$

For noncompactly supported $u$, apply this first to $\chi_R u$, where $\chi_R$ is a smooth cutoff with $\|\nabla \chi_R\|_\infty \le C/R$. The extra term is at most

$$\frac{C}{R}\int \|u\| \, \|\widehat{\Sigma}\|_{\mathrm{F}} \rho_t \, dx \le \frac{C}{R}\|u\|_{L^2(\rho_t)}\|\widehat{\Sigma}\|_{L^2(\rho_t;\mathrm{F})} \longrightarrow 0.$$

The other terms converge by the stated integrability assumptions. Completing the square therefore yields

$$\mathcal{J}_{h,t}(u) = \|u - \widehat{v}\|^2_{L^2(\rho_t)} - \|\widehat{v}\|^2_{L^2(\rho_t)}. \tag{26}$$

The identity extends continuously from smooth compactly supported fields to $L^2(\rho_t)$, with unique minimizer $\widehat{v}$. Here $\widehat{v}$ uses the true density $\rho_t$ as in equation 25.

## A.2 Proof of the PhiBE Coefficient Approximation

*Proof of Lemma 1.* Let $\mathcal{L}f = b \cdot \nabla f + \frac{1}{2}\Sigma : \nabla^2 f$. For $f_j(y) = y_j$, the assumed semigroup expansion gives

$$\widehat{b}_j(x) = \frac{\mathbb{E}[f_j(X_{t+\Delta t}) \mid X_t = x] - f_j(x)}{\Delta t} = b_j(x) + \frac{\Delta t}{2}\mathcal{L}b_j(x) + o(\Delta t)$$

uniformly on $\Omega$. Boundedness of $\mathcal{L}b$ on $\Omega$ and a sufficiently large $L_b$ give the first inequality for small $h$.

For $g_{jk,x}(y) = (y_j - x_j)(y_k - x_k)$, with $x$ held fixed when applying $\mathcal{L}$ in $y$, we have $g_{jk,x}(x) = 0$ and $\mathcal{L}g_{jk,x}(x) = \Sigma_{jk}(x)$. Consequently,

$$\widehat{\Sigma}_{jk}(x) = \Sigma_{jk}(x) + \Delta t R_{jk}(x) + o(\Delta t), \qquad R_{jk}(x) := \frac{1}{2}\mathcal{L}^2 g_{jk,x}(x).$$

The expansion and its first spatial derivative hold uniformly on $\Omega$ by assumption. Thus

$$\|\Sigma - \widehat{\Sigma}\|_{\infty,\Omega} \le \Delta t \|R\|_{\infty,\Omega} + o(\Delta t), \qquad \|\nabla \cdot (\Sigma - \widehat{\Sigma})\|_{\infty,\Omega} \le \Delta t \|\nabla \cdot R\|_{\infty,\Omega} + o(\Delta t).$$

This proves the remaining inequalities. □

### A.3 Proof of the General Population Approximation

*Proof of Theorem 1.* Positivity and continuity on $\mathcal{Z}$ imply $\inf_{\mathcal{Z}} \rho_t > 0$. Hence

$$S_1 := \sup_{\mathcal{Z}} \|\nabla \log \rho_t\| < \infty, \qquad S_2 := \sup_{\mathcal{Z}} \|\nabla^2 \log \rho_t\| < \infty.$$

Subtracting equation 25 from equation 7 and applying Lemma 1 gives

$$\|v - \widehat{v}\|_{\infty,\mathcal{Z}} \le \left( L_b + \frac{1}{2} L_\Sigma S_1 + \frac{1}{2} L_{\mathrm{div}} \right) \Delta t + o(\Delta t).$$

Take $L_v = L_b + \frac{1}{2} L_\Sigma S_1 + \frac{1}{2} L_{\mathrm{div}}$. The spatial derivatives of $v$ are bounded on $\mathcal{Z}$; for example, a positive Lipschitz constant $K$ can be chosen at least as large as

$$\|\nabla b\|_{\infty,\Omega} + \frac{1}{2} \|\nabla \Sigma\|_{\infty,\Omega} S_1 + \frac{1}{2} \|\Sigma\|_{\infty,\Omega} S_2 + \frac{1}{2} \|\nabla(\nabla \cdot \Sigma)\|_{\infty,\Omega}.$$

Here the norm of $\nabla\Sigma$ is the induced tensor norm. Convexity of $\Omega$ gives $\|v(x,t) - v(y,t)\| \le K\|x - y\|$ for $x, y \in \Omega$. As long as both flows remain in $\Omega$, their common initial condition implies

$$\|x_t - \widehat{x}_t\| \le K \int_0^t \|x_s - \widehat{x}_s\| \, ds + t\big(L_v \Delta t + o(\Delta t)\big).$$

Grönwall's inequality yields

$$\|x_t - \widehat{x}_t\| \le \frac{e^{Kt} - 1}{K} \big(L_v \Delta t + o(\Delta t)\big),$$

uniformly for the stated times $t \le T$. □

### A.4 Constant Isotropic Diffusion

If $\Sigma = \sigma^2 I_d$ is known, then $\Sigma : \nabla u = \sigma^2 \nabla \cdot u$ in the score-free objective. The same simplification applies to a covariance approximation constrained to be constant and isotropic, $\widetilde{\Sigma} = \widehat{\Sigma}^{\,2} I_d$. If

$$|\sigma^2 - \widehat{\Sigma}^{\,2}| \le L_{\sigma^2} \Delta t + o(\Delta t),$$

then, with $\widetilde{v} = \widehat{b} - \frac{1}{2} \widehat{\Sigma}^{\,2} \nabla \log \rho_t$, the proof above gives

$$\|v - \widetilde{v}\|_{\infty,\mathcal{Z}} \le \left( L_b + \frac{1}{2} L_{\sigma^2} S_1 \right) \Delta t + o(\Delta t).$$

The same flow bound follows under the same containment condition, using any positive $K$ at least $\|\nabla b\|_{\infty,\Omega} + \frac{1}{2}\sigma^2 S_2$. The raw increment second moment $\widehat{\Sigma}(x)$ in equation 11 need not be constant even when $\Sigma$ is constant, so this simplification does not automatically apply to the empirical objective equation 14.

### A.5 Ornstein–Uhlenbeck Population Approximation

**Corollary 2** (OU population approximation)**.** *Consider*

$$dX_t = AX_t \, dt + \sigma \, dB_t, \qquad \Sigma := \sigma\sigma^\top \succ 0, \qquad X_0 \sim \mathcal{N}(m_0, C_0), \quad C_0 \succ 0,$$

*with $X_0$ independent of $B$. Let $\Omega \subset \mathbb{R}^d$ be compact and $h = \Delta t$. The first-order PhiBE velocity is*

$$\widehat{v}(x,t) = \widehat{A}x - \frac{1}{2} \nabla \cdot \widehat{\Sigma}(x) + \frac{1}{2} \widehat{\Sigma}(x) C_t^{-1} (x - m_t),$$

*where*

$$M_h = e^{Ah} - I, \qquad \widehat{A} = M_h/h,$$
$$Q_h = \int_0^h e^{As}\Sigma e^{A^\top s}\, ds, \qquad \widehat{\Sigma}(x) = \frac{Q_h + M_h x x^\top M_h^\top}{h},$$

*and $m_t, C_t$ are the marginal mean and covariance. There are constants $L_{\mathrm{OU}}, K_{\mathrm{OU}} > 0$, independent of $h$, such that, uniformly on $\Omega \times [0,T]$ as $h \downarrow 0$,*

$$\|v(x,t) - \widehat{v}(x,t)\| \le L_{\mathrm{OU}} h + O(h^2).$$

*For solutions of $\dot{x}_t = v(x_t,t)$ and $\dot{\widehat{x}}_t = \widehat{v}(\widehat{x}_t,t)$ with the same initial state, while both remain in $\Omega$,*

$$\|x_t - \widehat{x}_t\| \le \frac{e^{K_{\mathrm{OU}} t} - 1}{K_{\mathrm{OU}}} L_{\mathrm{OU}} h + O(h^2), \qquad 0 \le t \le T.$$

*Proof.* The Gaussian marginals have

$$m_t = e^{At} m_0, \qquad C_t = e^{At} C_0 e^{A^\top t} + \int_0^t e^{As}\Sigma e^{A^\top s}\, ds.$$

Thus $\nabla \log \rho_t(x) = -C_t^{-1}(x - m_t)$ and $v(x,t) = Ax + \frac{1}{2}\Sigma C_t^{-1}(x - m_t)$. Conditionally on $X_t = x$, the increment is $M_h x + \xi_h$, where $\xi_h \sim \mathcal{N}(0, Q_h)$. Its first and raw second moments give $\widehat{b}(x) = \widehat{A}x$ and $\widehat{\Sigma}$ above. Direct differentiation yields

$$\nabla \cdot \widehat{\Sigma}(x) = \frac{M_h^2 x + \mathrm{tr}(M_h) M_h x}{h},$$

which also gives the stated expression for $\widehat{v}$.

Matrix exponential expansion gives, uniformly on $\Omega$ and also after the spatial derivative used below,

$$\widehat{A} = A + \tfrac{1}{2}A^2 h + O(h^2), \qquad \widehat{\Sigma}(x) = \Sigma + hH(x) + O(h^2),$$

where

$$H(x) = \tfrac{1}{2}(A\Sigma + \Sigma A^\top) + Axx^\top A^\top, \qquad \nabla \cdot H(x) = (A^2 + \mathrm{tr}(A)A)x.$$

Consequently,

$$v(x,t) - \widehat{v}(x,t) = \frac{h}{2}\left[-A^2 x + \nabla \cdot H(x) - H(x) C_t^{-1}(x - m_t)\right] + O(h^2).$$

Since $C_0 \succ 0$, $C_t^{-1}$ is uniformly bounded on $[0,T]$. With

$$R = \sup_{x \in \Omega} \|x\|, \qquad S_{\mathrm{OU}} = \sup_{(x,t) \in \Omega \times [0,T]} \|C_t^{-1}(x - m_t)\|,$$

the velocity bound follows by taking $L_{\mathrm{OU}}$ at least

$$\tfrac{1}{2}R\|A^2\| + \tfrac{1}{2}\|\nabla \cdot H\|_{\infty,\Omega} + \tfrac{1}{2}\|H\|_{\infty,\Omega} S_{\mathrm{OU}}.$$

Finally, $v$ is spatially Lipschitz with any positive constant

$$K_{\mathrm{OU}} \ge \sup_{0 \le t \le T} \|A + \tfrac{1}{2}\Sigma C_t^{-1}\|.$$

Subtracting the two ODEs and applying Grönwall's inequality gives the trajectory bound, with a remainder uniform for $t \in [0,T]$ while the trajectories remain in $\Omega$. □

### A.6 Linear-Basis Finite-Sample Setting and Proof

**Independent trajectories and the sampling measure.** Let $\Xi_k = (X_{t_0}^{(k)}, \ldots, X_{t_L}^{(k)})$, $k = 1, \ldots, N$, be independent copies of an SDE trajectory observed at $t_j = jh$, where $L = N_t \ge 1$, $0 < h = \Delta t \le 1$, and $Lh \le T$. Transitions within a trajectory are not assumed independent. Write $W_j^{(k)} = X_{t_{j+1}}^{(k)} - X_{t_j}^{(k)}$ and

$$\mu_L(dx, dt) = \frac{1}{L}\sum_{j=0}^{L-1} \rho_{t_j}(x)\, dx\, \delta_{t_j}(dt).$$

For one trajectory $\Xi$, define

$$\ell_\Xi(u) = \frac{1}{L}\sum_{j=0}^{L-1}\left[\|u(X_{t_j},t_j)\|^2 - 2u(X_{t_j},t_j)\cdot\frac{W_j}{h} - \frac{W_jW_j^\top}{h} : \nabla_x u(X_{t_j},t_j)\right].$$

Set $\mathcal{J}_N(u) = N^{-1}\sum_k \ell_{\Xi_k}(u)$ and $\mathcal{J}(u) = \mathbb{E}\ell_\Xi(u)$. This is the uniform-grid empirical objective equation 14. Conditional expectation and the integration-by-parts identity in Appendix A.1 give

$$\mathcal{J}(u) = \|u - \widehat{v}\|^2_{L^2(\mu_L)} - \|\widehat{v}\|^2_{L^2(\mu_L)}. \tag{27}$$

This objective averages over the observation times; no time quadrature claim is needed for this identity.

**Fixed linear-basis models.** Let $\psi_1,\dots,\psi_p : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ be deterministic basis fields and $h = \Delta t$, fixed independently of the training sample, for which the derivatives and expectations below are well defined. Write

$$\Psi(x,t) = (\psi_1(x,t) \quad \cdots \quad \psi_p(x,t)), \qquad \mathcal{V}_p = \{v_\theta = \Psi\theta : \theta \in \mathbb{R}^p\}.$$

The basis fields may be nonlinear in $(x,t)$; only their coefficients are fitted. For one trajectory, define

$$H_\Xi = \frac{1}{L}\sum_{j=0}^{L-1}\Psi(X_{t_j},t_j)^\top\Psi(X_{t_j},t_j),$$

and, for $\ell = 1,\dots,p$,

$$(c_\Xi)_\ell = \frac{1}{L}\sum_{j=0}^{L-1}\left[\psi_\ell(X_{t_j},t_j)\cdot\frac{W_j}{h} + \frac{W_jW_j^\top}{2h} : \nabla_x\psi_\ell(X_{t_j},t_j)\right].$$

Set

$$H = \mathbb{E}H_\Xi, \qquad c = \mathbb{E}c_\Xi, \qquad H_N = \frac{1}{N}\sum_{k=1}^N H_{\Xi_k}, \qquad c_N = \frac{1}{N}\sum_{k=1}^N c_{\Xi_k}.$$

The empirical and population objectives satisfy

$$\mathcal{J}_N(v_\theta) = \theta^\top H_N\theta - 2c_N^\top\theta, \qquad \mathcal{J}(v_\theta) = \theta^\top H\theta - 2c^\top\theta.$$

When $H \succ 0$, let $\theta_\star = H^{-1}c$. Then $\Psi\theta_\star$ is the population minimizer in $\mathcal{V}_p$. By equation 27, it equals $\widehat{v}$ under the realizability condition $\widehat{v} \in \mathcal{V}_p$. Define

$$\theta_N = \begin{cases} H_N^{-1}c_N, & H_N \succ 0, \\ 0, & \text{otherwise}, \end{cases} \qquad v_N = \Psi\theta_N.$$

On the event $H_N \succ 0$, this is the unique global empirical minimizer in $\mathcal{V}_p$. For a scalar random variable $U$, write

$$\|U\|_{\psi_1} := \inf\{a > 0 : \mathbb{E}\exp(|U|/a) \le 2\}.$$

**Assumption 1** (Linear-basis sampling conditions)**.** *In the preceding independent-trajectory setting, the identity equation 27 holds on $\mathcal{V}_p$, $H \succ 0$, and $\widehat{v} \in \mathcal{V}_p$. Let $h = \Delta t$. Define*

$$G_\Xi = H^{-1/2}H_\Xi H^{-1/2}, \qquad \eta_\Xi = H^{-1/2}(c_\Xi - H_\Xi\theta_\star).$$

*There exist constants $K_G, K_R > 0$, independent of $N$ and uniform over the observation grids under consideration, such that*

$$\sup_{\|a\|_2=1}\|a^\top(G_\Xi - I_p)a\|_{\psi_1} \le K_G, \qquad \max_{1\le\ell\le p}\|(\eta_\Xi)_\ell\|_{\psi_1} \le \frac{K_R}{\sqrt{h}}.$$

The moment conditions apply to the chosen basis and trajectory distribution; linearity alone does not imply them. Their constants may depend on the fixed basis and its conditioning. No uniform claim over growing basis families is made without uniform bounds.

*Proof of Theorem 2.* The definitions give

$$\mathbb{E}G_\Xi = I_p, \qquad \mathbb{E}\eta_\Xi = H^{-1/2}(c - H\theta_\star) = 0.$$

Put

$$G_N = H^{-1/2} H_N H^{-1/2}, \qquad \overline{\eta}_N = \frac{1}{N}\sum_{k=1}^{N} \eta_{\Xi_k}, \qquad \ell_\delta = \log(4p/\delta).$$

All concentration arguments concern the $N$ independent trajectory contributions; observations within a trajectory need not be independent.

Choose a $1/4$-net $\mathcal{A}$ of the Euclidean unit sphere in $\mathbb{R}^p$ with $|\mathcal{A}| \le 9^p$. Scalar Bernstein concentration and a union bound give, with probability at least $1 - \delta/2$,

$$\max_{a\in\mathcal{A}} |a^\top (G_N - I_p) a| \le C K_G \left[ \sqrt{\frac{p + \log(4/\delta)}{N}} + \frac{p + \log(4/\delta)}{N} \right].$$

For any symmetric matrix $B$, the net inequality gives

$$\|B\|_{\mathrm{op}} \le 2 \max_{a\in\mathcal{A}} |a^\top B a|.$$

Thus, if

$$N \ge C(1 + K_G + K_G^2)\big[p + \log(4/\delta)\big],$$

then $\|G_N - I_p\|_{\mathrm{op}} \le 1/2$. On this event, $H_N \succ 0$ and $\|G_N^{-1}\|_{\mathrm{op}} \le 2$.

Applying scalar Bernstein concentration to each coordinate of $\overline{\eta}_N$ and taking a union bound gives, with probability at least $1 - \delta/2$,

$$\|\overline{\eta}_N\|_2 \le C K_R \sqrt{\frac{p}{h}} \left[ \sqrt{\frac{\ell_\delta}{N}} + \frac{\ell_\delta}{N} \right].$$

On the intersection of these events, the empirical normal equations imply

$$H_N(\theta_N - \theta_\star) = c_N - H_N \theta_\star,$$

and therefore

$$G_N H^{1/2}(\theta_N - \theta_\star) = \overline{\eta}_N.$$

By realizability and the definition of $H$,

$$\|v_N - \widehat{v}\|^2_{L^2(\mu_L)} = (\theta_N - \theta_\star)^\top H (\theta_N - \theta_\star).$$

Consequently,

$$\|v_N - \widehat{v}\|_{L^2(\mu_L)} \le 2\|\overline{\eta}_N\|_2 \le C K_R \sqrt{\frac{p}{h}} \left[ \sqrt{\frac{\ell_\delta}{N}} + \frac{\ell_\delta}{N} \right].$$

For sufficiently large $c_0$ depending only on $K_G$, $N \ge c_0 p \log(2p/\delta)$ ensures both the Gram-matrix condition and $N \ge \ell_\delta$. The latter gives $\ell_\delta/N \le \sqrt{\ell_\delta/N}$. Since $\ell_\delta \le 2\log(2p/\delta)$, this proves

$$\|v_N - \widehat{v}\|_{L^2(\mu_L)} \le C_{\mathcal{V}} \sqrt{\frac{p \log(2p/\delta)}{Nh}}$$

with probability at least $1 - \delta$. □

**When the basis is not realizable.** If $\widehat{v} \notin \mathcal{V}_p$, the population minimizer $v_p = \Psi\theta_\star$ is its $L^2(\mu_L)$ projection onto $\mathcal{V}_p$. Under the other conditions of Assumption 1, with the residual $\eta_\Xi$ defined using this $\theta_\star$, the same proof bounds $\|v_N - v_p\|_{L^2(\mu_L)}$. Consequently, with the same probability,

$$\|v_N - \widehat{v}\|_{L^2(\mu_L)} \le \inf_{u\in\mathcal{V}_p} \|u - \widehat{v}\|_{L^2(\mu_L)} + C_{\mathcal{V}} \sqrt{\frac{p \log(2p/\delta)}{Nh}}.$$

Thus freezing neural features gives the required linear structure, but does not by itself establish realizability or the moment bounds.

### A.7 ORNSTEIN–UHLENBECK FINITE-SAMPLE CONVERGENCE

Consider

$$dX_t = A_{\mathrm{OU}}(X_t - m)\,dt + S\,dB_t, \qquad X_0 \sim \mathcal{N}(m_0, C_0), \quad C_0 \succ 0,$$

with $X_0$ independent of $B$. Observe $N$ independent trajectories at $t_j = jh$, $j = 0, \dots, L$, where $0 < h = \Delta t \le 1$ and $Lh \le T$. Write $W_j^{(k)} = X_{t_{j+1}}^{(k)} - X_{t_j}^{(k)}$. At each $j < L$, fit $u_M(x) = M(x^\top, 1)^\top$ by minimizing

$$\mathcal{J}_{N,j}^{\mathrm{aff}}(M) = \frac{1}{N}\sum_{k=1}^{N}\left[\|u_M(X_{t_j}^{(k)})\|^2 - 2u_M(X_{t_j}^{(k)}) \cdot \frac{W_j^{(k)}}{h} - \frac{W_j^{(k)} W_j^{(k)\top}}{h} : \nabla_x u_M\right].$$

Write $v_{N,\mathrm{aff}}(\cdot, t_j)$ and $\widehat{v}_{\mathrm{aff},h}(\cdot, t_j)$ for the empirical and population affine minimizers. Gaussian integration by parts shows that the latter is the $L^2(\rho_{t_j})$ projection of $\widehat{v}$ onto the affine class. It need not equal $\widehat{v}$: the raw increment second moment is quadratic in $x$, so the unrestricted PhiBE field can contain cubic terms. The compact set below is an evaluation region; the Gaussian samples are not truncated to it.

**Corollary 3** (Affine OU convergence at observation times)**.** *In the preceding setting, fix a compact set $\Omega \subset \mathbb{R}^d$ and put $p_{\mathrm{aff}} = d(d+1)$ and $\ell_\delta = \log(2p_{\mathrm{aff}}L/\delta)$. There exist constants $c_{\mathrm{OU}}, C_{\Omega,\mathrm{OU}} > 0$, depending only on the OU parameters, $T, d, \Omega$, such that, if $\delta \in (0,1)$ and $N \ge c_{\mathrm{OU}} p_{\mathrm{aff}} \ell_\delta$, then, with probability at least $1 - \delta$,*

$$\max_{0 \le j < L} \sup_{x \in \Omega} \|v_{N,\mathrm{aff}}(x, t_j) - \widehat{v}_{\mathrm{aff},h}(x, t_j)\| \le C_{\Omega,\mathrm{OU}} \sqrt{\frac{p_{\mathrm{aff}}\ell_\delta}{Nh}}.$$

*Proof.* Put $q = d + 1$, $z_j = (X_{t_j}^\top, 1)^\top$, and $w_j = W_j$. Define

$$H_{N,j} = \frac{1}{N}\sum_{k=1}^{N} z_j^{(k)} z_j^{(k)\top},$$
$$D_{N,j} = \frac{1}{N}\sum_{k=1}^{N}\left[\frac{w_j^{(k)} z_j^{(k)\top}}{h} + \left(\frac{w_j^{(k)} w_j^{(k)\top}}{2h}, 0\right)\right],$$

where $(B, 0)$ appends a zero column. Let $H_j = \mathbb{E}H_{N,j}$ and $D_j = \mathbb{E}D_{N,j}$. The objective is $\mathrm{tr}(MH_{N,j}M^\top) - 2\langle M, D_{N,j}\rangle_F$, so its normal equations and their population counterparts are

$$M_{N,j}H_{N,j} = D_{N,j}, \qquad \widehat{M}_j H_j = D_j.$$

The Gaussian marginal means and covariances are bounded on $[0, T]$. Since $C_0 \succ 0$, their covariances are also uniformly positive definite, and hence $\inf_{j,h} \lambda_{\min}(H_j) =: \lambda_* > 0$. For $N \ge q$, the augmented Gaussian design has full row rank almost surely at all observation times; thus the empirical minimizers exist and are unique. Choose $c_{\mathrm{OU}}$ to ensure $N \ge q$.

The exact transition representation is

$$w_j = F_h(X_{t_j} - m) + \xi_j, \qquad F_h = e^{A_{\mathrm{OU}}h} - I,$$

$$\xi_j \sim \mathcal{N}\left(0, \int_0^h e^{A_{\mathrm{OU}}s} S S^\top e^{A_{\mathrm{OU}}^\top s}\,ds\right),$$

where $\xi_j$ is independent of $X_{t_j}$. Consequently, $\|F_h\| \le Ch$, the coordinates of $z_j$ have uniformly bounded sub-Gaussian norms, and those of $w_j$ have sub-Gaussian norms at most $C\sqrt{h}$. Products of sub-Gaussian variables are sub-exponential even when the factors are dependent. Thus the centered entries of $z_j z_j^\top$ have $\psi_1$-norm at most $C$, and those of $w_j z_j^\top/h + (w_j w_j^\top/(2h), 0)$ have $\psi_1$-norm at most $C/\sqrt{h}$. Here constants are independent of $j, N, h, L, \delta$.

Apply scalar Bernstein inequalities across the $N$ independent trajectories and take a union bound over entries and times. With probability at least $1-\delta$, simultaneously,

$$\max_{j<L} \|H_{N,j} - H_j\|_{\mathrm{op}} \le Cq\left(\sqrt{\frac{\ell_\delta}{N}} + \frac{\ell_\delta}{N}\right),$$

$$\max_{j<L} \|D_{N,j} - D_j\|_F \le C\sqrt{\frac{p_{\mathrm{aff}}}{h}}\left(\sqrt{\frac{\ell_\delta}{N}} + \frac{\ell_\delta}{N}\right).$$

Since $q^2 \le 2p_{\mathrm{aff}}$, the sample-size condition ensures $H_{N,j} \succeq (\lambda_*/2)I_q$ for every $j < L$. The transition representation also gives $\sup_{j,h} \|D_j\|_F < \infty$: the term involving $\mathbb{E}[\xi_j z_j^\top]$ vanishes, $F_h/h$ is bounded, and $\mathbb{E}[w_j w_j^\top]/h$ is bounded. Therefore $\widehat{M}_j = D_j H_j^{-1}$ is uniformly bounded. Subtracting the normal equations yields

$$M_{N,j} - \widehat{M}_j = \big[D_{N,j} - D_j + \widehat{M}_j(H_j - H_{N,j})\big] H_{N,j}^{-1}.$$

The preceding bounds, $h \le 1$, and $q \le \sqrt{2p_{\mathrm{aff}}}$ give

$$\max_{j<L} \|M_{N,j} - \widehat{M}_j\|_F \le C\sqrt{\frac{p_{\mathrm{aff}}\ell_\delta}{Nh}}.$$

Multiplication by $\sup_{x\in\Omega} \|(x^\top, 1)^\top\| < \infty$ proves the claim. No independence between transitions within a trajectory is used. □

### A.8 Proof of the Overall Convergence

*Proof of Corollary 1.* Restriction to $\Omega$ and the triangle inequality give

$$\|v_N - v\|_{L^2(\mu_L;\Omega)} \le \|v_N - \widehat{v}\|_{L^2(\mu_L)} + \|\widehat{v} - v\|_{L^2(\mu_L;\Omega)}.$$

Theorem 2 bounds the first term. Since $L^{-1}\sum_{j=0}^{L-1} \int_\Omega \rho_{t_j}(x)\,dx \le 1$, the second is at most $\|\widehat{v} - v\|_{\infty,\Omega\times[0,T]}$ and is bounded by Theorem 1. This proves the result. □

## B Experimental Settings and More Results

### B.1 Ornstein–Uhlenbeck Process

**Dynamics and analytical solution.** We consider the two-dimensional Ornstein–Uhlenbeck process

$$dX_t = \Gamma(X_t - \mu)\,dt + \sigma\,dB_t, \qquad X_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \tag{28}$$

where $\Gamma \in \mathbb{R}^{2\times 2}$, $\sigma \in \mathbb{R}^{2\times 2}$, $\mu \in \mathbb{R}^2$. We first establish a general result for the marginal distributions of such OU process with Gaussian initial.

**Theorem 3** (Gaussian marginals of the OU process)**.** *Consider the Ornstein–Uhlenbeck process*

$$dX_t = \Gamma(X_t - \mu)\,dt + \sigma\,dB_t, \qquad X_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \tag{29}$$

*where $\Gamma \in \mathbb{R}^{d\times d}$ has eigenvalues with strictly negative real parts, $\sigma \in \mathbb{R}^{d\times r}$ is constant, $\sigma\sigma^\top \succ 0$, and $\Sigma_0 \succ 0$. Assume that $X_0$ is independent of the Brownian motion $W$.*

*Then, for every $t \ge 0$, the marginal distribution of $X_t$ is Gaussian, $\mathcal{N}(\mu_t, \Sigma_t)$, with*

$$\mu_t = \mu + \exp(\Gamma t)(\mu_0 - \mu), \tag{30}$$

$$\Sigma_t = \exp(\Gamma t)(\Sigma_0 - B)\exp(\Gamma^\top t) + B, \tag{31}$$

*where $B$ is the unique symmetric positive-definite solution of the Lyapunov equation*

$$\Gamma B + B\Gamma^\top = -\sigma\sigma^\top. \tag{32}$$

*In particular,*

$$\mu_t \longrightarrow \mu, \qquad \Sigma_t \longrightarrow B, \qquad t \longrightarrow \infty,$$

*and the process has the unique stationary distribution $\mathcal{N}(\mu, B)$.*

We defer the proof to the end of this section.

**Reversible case:** We set

$$\Gamma = \mathrm{diag}(-1, -4), \qquad \mu = (4, 4)^\top, \qquad \sigma = \frac{1}{\sqrt{2}} I_2, \qquad \mu_0 = (0, 0)^\top, \qquad \Sigma_0 = I_2.$$

By Theorem 3, the marginal distribution remains Gaussian,

$$\rho_t = \mathcal{N}(\mu_t, \Sigma_t),$$

where

$$\mu_t = \begin{bmatrix} 4(1 - e^{-t}) \\ 4(1 - e^{-4t}) \end{bmatrix}, \qquad \Sigma_t = \begin{bmatrix} e^{-2t} + \frac{1-e^{-2t}}{4} & 0 \\ 0 & e^{-8t} + \frac{1-e^{-8t}}{16} \end{bmatrix}. \tag{33}$$

Since

$$\nabla_x \log \rho_t(x) = -\Sigma_t^{-1}(x - \mu_t),$$

the exact probability velocity under the SDE convention of equation 28 is

$$v^\star(x, t) = \Gamma(x - \mu) + \frac{1}{4}\Sigma_t^{-1}(x - \mu_t). \tag{34}$$

We refer readers to Pavliotis (2014) for detailed derivation of the above formulation. Equation 34 is used only for evaluation and is not provided to the neural PhiBE-Flow model during training.

**Non-reversible case:** We retain the values of $\mu$, $\mu_0$, and $\Sigma_0$ from the reversible case and choose

$$\Gamma = \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}, \qquad \sigma = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

The eigenvalues of $\Gamma$ are $-1$ and $-3$, so the stability condition in Theorem 3 is satisfied.

By Theorem 3, the marginal density is Gaussian, and its score is

$$\nabla_x \log \rho(x, t) = -\Sigma_t^{-1}(x - \mu_t).$$

The probability velocity associated with equation 29 is therefore

$$\begin{aligned} v(x, t) &= \Gamma(x - \mu) - \frac{1}{2}\sigma\sigma^\top \nabla_x \log \rho(x, t) \\ &= \Gamma(x - \mu) + \frac{1}{2}\sigma\sigma^\top \Sigma_t^{-1}(x - \mu_t), \end{aligned} \tag{35}$$

where $\mu_t$ and $\Sigma_t$ are given by equation 30 and equation 31, respectively. The corresponding deterministic probability flow satisfies

$$\dot{x}_t = v(x_t, t).$$

Substituting the expression for $\mu_t$ into equation 35, we obtain

$$\begin{aligned} v(x, t) &= \left(\Gamma + \frac{1}{2}\sigma\sigma^\top \Sigma_t^{-1}\right)(x - \mu) + \frac{1}{2}\sigma\sigma^\top \Sigma_t^{-1}(\mu - \mu_t) \\ &= \Gamma(x - \mu) + \frac{1}{2}\sigma\sigma^\top \Sigma_t^{-1}\left(x - \mu + \exp(\Gamma t)(\mu - \mu_0)\right). \end{aligned}$$

As $t \to \infty$, the marginal density converges to

$$\rho_\infty(x) = \frac{1}{\sqrt{(2\pi)^d \det(B)}} \exp\left(-\frac{1}{2}(x - \mu)^\top B^{-1}(x - \mu)\right),$$

and the probability velocity converges to

$$v_\infty(x) = \mathscr{B}(x - \mu),$$

where

$$\mathscr{B} = \Gamma + \frac{1}{2}\sigma\sigma^\top B^{-1} = \frac{1}{2}\left(\Gamma - B\Gamma^\top B^{-1}\right). \tag{36}$$

The second equality follows from equation 32.

The limiting velocity matrix satisfies

$$\mathscr{B}B + B\mathscr{B}^\top = 0,$$

or, equivalently,

$$\mathscr{B}^\top = -B^{-1}\mathscr{B}B.$$

Hence, the transformed matrix

$$\widetilde{\mathscr{B}} = B^{-1/2}\mathscr{B}B^{1/2}$$

is antisymmetric. The limiting probability flow therefore preserves the quadratic form

$$\frac{\mathrm{d}}{\mathrm{d}t}\left((x_t-\mu)^\top B^{-1}(x_t-\mu)\right) = 0,$$

and leaves the stationary Gaussian distribution invariant. Thus, a stationary marginal distribution can coexist with persistent probability circulation.

For the matrices $\Gamma$ and $\sigma$ specified above, solving equation 32 yields

$$B = \frac{1}{16}\begin{bmatrix} 5 & -2 \\ -2 & 3 \end{bmatrix}, \qquad B^{-1} = \frac{16}{11}\begin{bmatrix} 3 & 2 \\ 2 & 5 \end{bmatrix}.$$

Substituting these expressions into equation 36 gives

$$\begin{aligned}\mathscr{B} &= \frac{1}{2}\left(\Gamma - B\Gamma^\top B^{-1}\right) \\ &= \frac{1}{11}\begin{bmatrix} 2 & 5 \\ -3 & -2 \end{bmatrix}.\end{aligned}$$

Consequently, the limiting probability-flow dynamics are

$$\dot{x}_t = \mathscr{B}(x_t - \mu).$$

The nonconstant trajectories follow closed elliptic orbits centered at $\mu$, with orthogonal principal-axis directions $(\sqrt{5}-1, 2)^\top$ and $(-(\sqrt{5}+1), 2)^\top$. Moreover,

$$\mathscr{B}^2 = -\frac{1}{11}I,$$

so the trajectories are periodic with angular frequency $\omega = 1/\sqrt{11}$ and period $2\pi\sqrt{11}$.

**Trajectory generation.** We generate independent sample trajectories from equation 28 and record the states at discrete physical-time nodes

$$0 = t_0 < t_1 < \cdots < t_{N_t-1} = T.$$

The resulting training set consists of adjacent transition pairs

$$\left\{\left(X_{t_i}^{(k)}, X_{t_{i+1}}^{(k)}\right)\right\}_{k=1,\ldots,N_x;\ i=0,\ldots,N_t-2}.$$

For the neural velocity-field experiment, we use

$$T = 1, \qquad N_x = 50{,}000, \qquad N_t = 2{,}000.$$

**Affine velocity field and optimization.** As discussed in Sec. 5.1, for the non-reversible Ornstein–Uhlenbeck (OU) process, we seek an affine vector field $v_{\theta_i} : \mathbb{R}^d \to \mathbb{R}^d$ at each time step $t_i$ that approximates

$$\mathbf{b}(\cdot) - \frac{1}{2}\big(\nabla\cdot\mathbf{\Sigma}(\cdot)\big)^\top - \frac{1}{2}\mathbf{\Sigma}(\cdot)\nabla\log\rho_{t_i}(\cdot).$$

For convenience, let $N = N_{\mathbf{x}}$, $x_k = x_{t_i}^{(k)}$, $\Delta x_k = x_{t_{i+1}}^{(k)} - x_{t_i}^{(k)}$, and $\Delta t_i = t_{i+1} - t_i$. The corresponding empirical least-squares problem is

$$\min_{\theta_i\in\mathbb{R}^m} \frac{1}{N}\sum_{k=1}^{N}\left[\|v_{\theta_i}(x_k)\|^2 - 2v_{\theta_i}(x_k)\cdot\frac{\Delta x_k}{\Delta t_i} - 2\nabla_x v_{\theta_i}(x_k) : \frac{\Delta x_k \Delta x_k^\top}{2\Delta t_i}\right]. \tag{37}$$

For a fixed time step, we suppress the time index in the parameters and write

$$v_\theta(x) = Ax + \mu, \qquad A \in \mathbb{R}^{d\times d}, \quad \mu \in \mathbb{R}^d, \quad \theta = (A, \mu).$$

Introducing the augmented matrix and vector

$$\widetilde{A} = [A \mid \mu] \in \mathbb{R}^{d\times(d+1)}, \qquad \widetilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathbb{R}^{d+1},$$

we have $v_\theta(x) = \widetilde{A}\widetilde{x}$. Define the augmented sample matrix and the increment matrix as

$$\widetilde{X} = [\widetilde{x}_1, \ldots, \widetilde{x}_N] \in \mathbb{R}^{(d+1)\times N}, \qquad \widetilde{x}_k = \begin{bmatrix} x_k \\ 1 \end{bmatrix}, \qquad \Delta X = [\Delta x_1, \ldots, \Delta x_N] \in \mathbb{R}^{d\times N}.$$

Let

$$\mathbf{B} = \frac{\Delta X}{\Delta t_i} \in \mathbb{R}^{d\times N}, \qquad \mathbf{C} = \frac{\Delta X \Delta X^\top}{2\Delta t_i} \in \mathbb{R}^{d\times d}, \qquad \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} \\ 0^\top \end{bmatrix} \in \mathbb{R}^{(d+1)\times d}.$$

Since $\nabla_x v_\theta(x) = A$, the objective in equation 37 can be written as

$$\frac{1}{N}\left[\|\widetilde{A}\widetilde{X}\|_F^2 - 2\,\mathrm{Tr}\big(\mathbf{B}^\top \widetilde{A}\widetilde{X}\big) - 2\,\mathrm{Tr}\big(\widetilde{A}\widetilde{\mathbf{C}}\big)\right].$$

Thus, the minimization problem is equivalent to

$$\min_{\widetilde{A}} \mathrm{Tr}\big(\widetilde{X}\widetilde{X}^\top \widetilde{A}^\top \widetilde{A}\big) - 2\,\mathrm{Tr}\big(\widetilde{X}\mathbf{B}^\top \widetilde{A}\big) - 2\,\mathrm{Tr}\big(\widetilde{\mathbf{C}}\widetilde{A}\big).$$

Setting the gradient with respect to $\widetilde{A}$ to zero yields the normal equations

$$\widetilde{A}^*\widetilde{X}\widetilde{X}^\top = \mathbf{B}\widetilde{X}^\top + \widetilde{\mathbf{C}}^\top.$$

Provided that these equations are consistent, the minimizer with the smallest Frobenius norm is

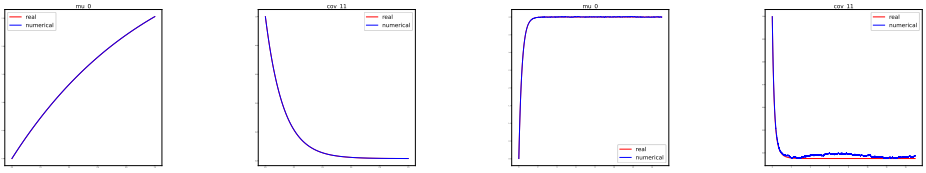
$$\widetilde{A}^* = \big(\mathbf{B}\widetilde{X}^\top + \widetilde{\mathbf{C}}^\top\big)\big(\widetilde{X}\widetilde{X}^\top\big)^\dagger,$$

where † denotes the Moore–Penrose pseudoinverse. If $\widetilde{X}$ has full row rank, the minimizer is unique, and the pseudoinverse reduces to the ordinary inverse. The resulting optimal vector field at time $t_i$ is therefore

$$v_{\theta_i^*}(x) = \widetilde{A}^*\widetilde{x},$$

with coefficients computed directly from the observed data pairs.

We apply this optimization method for affine vector fields to both reversible and non-reversible OU processes. Figure 8 compares the time evolution of selected mean and covariance components of the marginal distributions obtained from the computed probability flows with their exact counterparts. The numerical results show good agreement with the exact solutions in both cases.



(a) $\mu_{t,0}$ (Reversible) (b) $\mathbf{\Sigma}_{t,11}$ (Reversible) (c) $\mu_{t,0}$ (Non-reversible) (d) $\mathbf{\Sigma}_{t,11}$ (Non-reversible)

Figure 8: Time evolution of selected mean and covariance components of the marginal distributions for reversible (left two panels) and non-reversible (right two panels) OU processes. Results obtained from the computed probability flows are compared with the exact solutions.

**Velocity network and optimization.** The probability velocity $v_\theta(x, t)$ is represented by an MLP with five hidden layers and hidden width 50. The network takes the physical state $x \in \mathbb{R}^2$ and physical time $t$ as input and outputs a two-dimensional velocity vector. The model is optimized using the empirical PhiBE-Flow objective described in Section 3.

After training, we generate probability-flow trajectories by numerically integrating

$$\dot{X}_t = v_\theta(X_t, t).$$

**Velocity-field error.** Because the exact probability velocity is available, we evaluate the learned field using

$$\|v_\theta - v^\star\|_{L^2(\rho\otimes[0,T])},$$

where the expectation over $\rho_t$ is approximated using independent samples from the analytical OU marginal. We additionally report the time-dependent velocity error

$$\|v_\theta(\cdot,t) - v^\star(\cdot,t)\|_{L^2(\rho_t)}.$$

**Dependence on temporal resolution and sample size.** To study the effect of temporal resolution, we fix $N_x = 30{,}000$ and vary

$$N_t = 10 \times 2^k, \qquad k = 0, 1, \ldots, 8.$$

To study finite-sample behavior, we fix $N_t = 3{,}000$ and vary

$$N_x = 100 \times 2^k, \qquad k = 0, 1, \ldots, 9.$$

For each setting, we measure the global $L^2(\rho \otimes [0,T])$ error between the learned and exact probability velocities.

**Proof of Theorem 3.**

*Proof.* The explicit solution of equation 29 is

$$X_t = \mu + \exp(\Gamma t)(X_0 - \mu) + \int_0^t \exp(\Gamma(t-s))\sigma\, dB_s.$$

Since $X_0$ is Gaussian and independent of the Gaussian stochastic integral, $X_t$ is Gaussian. Taking expectations gives equation 30, while the Itô isometry yields

$$\Sigma_t = \exp(\Gamma t)\Sigma_0 \exp(\Gamma^\top t) + \int_0^t \exp(\Gamma s)\sigma\sigma^\top \exp(\Gamma^\top s)\, ds. \tag{38}$$

Because $\Gamma$ is stable, the integral

$$B = \int_0^\infty \exp(\Gamma s)\sigma\sigma^\top \exp(\Gamma^\top s)\, ds$$

converges and defines the unique symmetric positive-definite solution of equation 32. Moreover,

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\exp(\Gamma s)B\exp(\Gamma^\top s)\right) = -\exp(\Gamma s)\sigma\sigma^\top \exp(\Gamma^\top s).$$

Integrating this identity gives

$$\int_0^t \exp(\Gamma s)\sigma\sigma^\top \exp(\Gamma^\top s)\, ds = B - \exp(\Gamma t)B\exp(\Gamma^\top t).$$

Substitution into the covariance formula establishes equation 31. Finally, $\exp(\Gamma t) \to 0$ as $t \to \infty$, which gives the stated limits and the stationary distribution. □

### B.2 Double Pendulum System

We use the standard Acrobot configuration with $m_1 = m_2 = 1$, $l_1 = l_2 = 1$, $l_{c1} = l_{c2} = 0.5$, $I_1 = I_2 = 1$, and gravitational acceleration $g = 9.8$. Stochasticity is introduced by adding independent Brownian perturbations to the two angular accelerations.

For the state-space experiment, we generate 1000 trajectories, each containing 240 observations corresponding to 8 seconds of simulation at 30 frames per second. Initial states are sampled uniformly from

$$\Omega = \left\{(\theta_1, \theta_2, \dot\theta_1, \dot\theta_2) : -\pi \le \theta_1, \theta_2 \le \pi,\ -0.1 \le \dot\theta_1, \dot\theta_2 \le 0.1\right\}.$$

The probability-velocity model is implemented as a residual MLP with hidden dimension 256 and two residual blocks. We optimize the model using Adam with learning rate $10^{-3}$ for 50 epochs. Results are generated autoregressively using forward Euler integration.

**Algorithm 1** State-space training and autoregressive generation with time input

**Part I: Training**

**Require:** Trajectories $\mathcal{D} = \{(x_i^{(\ell)}, t_i^{(\ell)})_{i=0}^{M-1}\}_{\ell=1}^{N_{\mathrm{traj}}}$ with spacing $\Delta t$; model $v_\omega(x,t)$; learning rate $\eta = 10^{-3}$; epochs $E = 50$

**for** $e = 1, \dots, E$ **do**
  **for** each consecutive pair $(x_i, x_{i+1})$ and its timestamp $t_i$ in $\mathcal{D}$ **do**
    $\delta x_i \leftarrow x_{i+1} - x_i$
    $v \leftarrow v_\omega(x_i, t_i)$
    $b \leftarrow \delta x_i / \Delta t, \quad A \leftarrow \delta x_i \delta x_i^\top / \Delta t$
    $J \leftarrow \nabla_x v_\omega(x, t_i)|_{x=x_i}$ ▷ hold time fixed
    $\mathcal{L} \leftarrow \|v\|_2^2 - 2\langle b, v\rangle - \mathrm{Tr}(AJ)$
    $\omega \leftarrow \mathrm{Adam}(\omega, \nabla_\omega \mathcal{L}, \eta)$
  **end for**
**end for**

**Part II: Autoregressive generation**

**Require:** Initial state $x_{\mathrm{in}}$ at time $t_{\mathrm{in}}$; trained $v_\omega$; number of forecast steps $H$; time step $\Delta t$

$\widehat{x}_0 \leftarrow x_{\mathrm{in}}, \quad \tau_0 \leftarrow t_{\mathrm{in}}$
$\mathcal{X}_{\mathrm{gen}} \leftarrow [(\widehat{x}_0, \tau_0)]$
**for** $h = 0, \dots, H-1$ **do**
  $v \leftarrow v_\omega(\widehat{x}_h, \tau_h)$
  $\widehat{x}_{h+1} \leftarrow \widehat{x}_h + \Delta t\, v$
  $\tau_{h+1} \leftarrow \tau_h + \Delta t$
  Append $(\widehat{x}_{h+1}, \tau_{h+1})$ to $\mathcal{X}_{\mathrm{gen}}$
**end for**
**return** $\mathcal{X}_{\mathrm{gen}}$

### B.2.1 State-Space Implementation Details

The state-space model operates directly on $x_t = (\theta_1, \theta_2, \dot\theta_1, \dot\theta_2)_t$. Consecutive observations $(x_t, x_{t+1})$ are used to construct the empirical drift and diffusion terms required by the training objective. Algorithm 1 summarizes training and autoregressive generation.

### B.2.2 Image Data Generation

For the image-based experiment, each physical trajectory is converted into a sequence of visual observations. Given the joint angles $(\theta_1, \theta_2)$, the Cartesian coordinates of the first and second joints are computed as

$$x_1 = \sin(\theta_1), \qquad y_1 = -\cos(\theta_1),$$

and

$$x_2 = x_1 + \sin(\theta_1 + \theta_2), \qquad y_2 = y_1 - \cos(\theta_1 + \theta_2).$$

Both links have unit length. The pendulum is rendered inside the Cartesian domain $[-2, 2] \times [-2, 2]$, with rods and joints shown in white against a black background. Each rendered frame is converted to grayscale and resized to the input resolution of the encoder.

For representation learning and latent-space generation, we generate 1000 image trajectories, each containing 240 consecutive frames.

### B.2.3 Geometry-Preserving Encoder and Decoder

We employ a geometry-preserving encoder $T_\phi$ (Lee et al., 2025) to map an image $x_t$ to a three-dimensional latent representation $z_t = T_\phi(x_t)$ and a decoder $S_\psi$ to reconstruct the image from the latent representation. During decoder training, Gaussian perturbations are added to the latent variables to improve robustness to errors accumulated during autoregressive prediction.

The encoder and decoder architectures used for the double-pendulum experiment are summarized in Table 2.

Table 2: Architectures of the geometry-preserving encoder $T_\phi$ and decoder $S_\psi$.

| Model | Layer | Output Shape |
|---|---|---|
| **Encoder** | Input image $x$ (resized) | $1 \times 32 \times 32$ |
| | Conv2d($1 \to 32$), LeakyReLU, BatchNorm | $32 \times 16 \times 16$ |
| | Conv2d($32 \to 64$), LeakyReLU | $64 \times 8 \times 8$ |
| | Conv2d($64 \to 128$), LeakyReLU, BatchNorm | $128 \times 4 \times 4$ |
| | Flatten, Linear($2048 \to 3$) | 3 |
| **Decoder** | Input latent $z$ | $3 \times 1 \times 1$ |
| | ConvTranspose2d($3 \to 64$), Conv2d($64 \to 128$) | $128 \times 4 \times 4$ |
| | ConvTranspose2d($128 \to 64$), Conv2d($64 \to 64$) | $64 \times 8 \times 8$ |
| | ConvTranspose2d($64 \to 32$), Conv2d($32 \to 32$) | $32 \times 16 \times 16$ |
| | ConvTranspose2d($32 \to 128$), Conv2d($128 \to 1$), Tanh | $1 \times 32 \times 32$ |

#### B.2.4 Encoder and Decoder Training

The encoder is first trained to preserve pairwise geometry in the latent space while encouraging temporally adjacent observations to remain close. After the encoder is fixed, the decoder is trained to reconstruct images from noisy latent representations. Algorithm 2 summarizes the procedure.

#### B.2.5 Latent-Space Dynamics

Once the image representation is learned, each encoded latent position $z_t \in \mathbb{R}^3$ is augmented with its finite-difference velocity,

$$\dot{z}_t = \frac{z_t - z_{t-1}}{\Delta t},$$

yielding the six-dimensional dynamical state

$$s_t = \begin{bmatrix} z_t \\ \dot{z}_t \end{bmatrix} \in \mathbb{R}^6.$$

We use a neural network $v_\omega$ to learn the probability velocity in this latent dynamical space. The specific neural architecture can be chosen according to the dataset and is not essential to the proposed training objective.

#### B.2.6 Latent-Space Training and Prediction

Algorithm 3 describes training and autoregressive prediction in the learned latent space. The same probability-velocity objective as in the explicit state-space experiment is applied to the latent dynamical states.

### B.3 2D Navier–Stokes Equations Generation

**Stochastic Navier–Stokes dataset.** We use the two-dimensional stochastic Navier–Stokes dataset from Chen et al. (2024), consisting of vorticity fields on the periodic domain $[0, 2\pi]^2$. The dynamics incorporate viscosity $\nu = 10^{-3}$, linear damping $\alpha = 0.1$, and white-in-time stochastic forcing with amplitude $\varepsilon = 1$ acting on selected Fourier modes. The data were generated using a pseudo-spectral solver with Euler–Maruyama time stepping on a $256 \times 256$ grid, with integration step $\delta t = 10^{-4}$. The dataset comprises 1,000 trajectories simulated over $t \in [0, 100]$, with the initial transient $t \in [0, 50]$ discarded. Vorticity snapshots were recorded every $\Delta t = 0.5$, yielding $2 \times 10^5$ snapshots, and subsequently downsampled to $128 \times 128$ resolution.

### B.4 KTH Video Generation

**Pretrained video autoencoder.** Following Chen et al. (2024), we use a pretrained VQGAN (Esser et al., 2021) to represent KTH videos in a compact latent space. We download the KTH autoencoder

**Algorithm 2** Training the Geometry-Preserving Encoder and Decoder

**Require:** Dataset $\mathcal{D} = \{X_1^l, X_2^l, \ldots, X_{240}^l\}_{l=1}^{1000}$, encoder $T_\phi$, decoder $S_\psi$, $\lambda_{\text{smooth}} = 25$, noise scale $\sigma = 0.3$, learning rate $\eta = 10^{-4}$, batch size $B = 100$

**Stage I: Encoder Training**

**for** $k = 1, 2, \ldots, 50{,}000$ **do**
  Sample $\mathcal{B} = \{(x_t^{(b)}, x_{t+1}^{(b)})\}_{b=1}^{B}$
  **for** $b = 1, 2, \ldots, B$ **do**
    $z_t^{(b)} \leftarrow T_\phi(x_t^{(b)})$
    $z_{t+1}^{(b)} \leftarrow T_\phi(x_{t+1}^{(b)})$
  **end for**
  $\mathcal{L}_{\text{GME}}^{(t)} \leftarrow \frac{1}{B^2} \sum_{i,j=1}^{B} \left[\log\left(1 + \|z_t^{(i)} - z_t^{(j)}\|_2^2\right) - \log\left(1 + \|x_t^{(i)} - x_t^{(j)}\|_2^2\right)\right]^2$
  $\mathcal{L}_{\text{GME}}^{(t+1)} \leftarrow \frac{1}{B^2} \sum_{i,j=1}^{B} \left[\log\left(1 + \|z_{t+1}^{(i)} - z_{t+1}^{(j)}\|_2^2\right) - \log\left(1 + \|x_{t+1}^{(i)} - x_{t+1}^{(j)}\|_2^2\right)\right]^2$
  $\mathcal{L}_{\text{GPE}} \leftarrow \frac{1}{2}\left(\mathcal{L}_{\text{GME}}^{(t)} + \mathcal{L}_{\text{GME}}^{(t+1)}\right)$
  $\mathcal{L}_{\text{smooth}} \leftarrow \frac{1}{B} \sum_{b=1}^{B} \|z_{t+1}^{(b)} - z_t^{(b)}\|_2^2$
  $\mathcal{L}_{\text{enc}} \leftarrow \mathcal{L}_{\text{GPE}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}$
  $\phi \leftarrow \text{Adam}\big(\phi, \nabla_\phi \mathcal{L}_{\text{enc}}, \eta\big)$
**end for**

**Stage II: Decoder Training**

**for** $k = 1, 2, \ldots, 100{,}000$ **do**
  Sample $\mathcal{B} = \{(x_t^{(b)}, x_{t+1}^{(b)})\}_{b=1}^{B}$
  **for** $b = 1, 2, \ldots, B$ **do**
    $z_t^{(b)} \leftarrow T_\phi(x_t^{(b)})$
    $z_{t+1}^{(b)} \leftarrow T_\phi(x_{t+1}^{(b)})$
    $\epsilon_t^{(b)}, \epsilon_{t+1}^{(b)} \sim \mathcal{N}(0, \sigma^2 I)$
    $\hat{x}_t^{(b)} \leftarrow S_\psi(z_t^{(b)} + \epsilon_t^{(b)})$
    $\hat{x}_{t+1}^{(b)} \leftarrow S_\psi(z_{t+1}^{(b)} + \epsilon_{t+1}^{(b)})$
  **end for**
  $\mathcal{L}_{\text{recon}} \leftarrow \frac{1}{2B} \sum_{b=1}^{B} \left(\|x_t^{(b)} - \hat{x}_t^{(b)}\|_2^2 + \|x_{t+1}^{(b)} - \hat{x}_{t+1}^{(b)}\|_2^2\right)$
  $\psi \leftarrow \text{Adam}\big(\psi, \nabla_\psi \mathcal{L}_{\text{recon}}, \eta\big)$
**end for**

checkpoint released by RIVER (Davtyan et al., 2023), which is also used in their experiments. The encoder maps each $64 \times 64$ frame to a $4 \times 8 \times 8$ latent representation. PhiBE-Flow learns the probability velocity field in this latent space, and the generated latent trajectories are mapped back to video frames using the pretrained decoder.

**Evaluation metrics.** We evaluate video results using Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Fréchet Video Distance (FVD). MSE, PSNR, and SSIM evaluate frame-level reconstruction quality, while FVD measures the discrepancy between real and generated videos in a learned spatiotemporal feature space.

**MSE.** The mean squared error is

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2, \tag{39}$$

**Algorithm 3** Latent-space training and single-state autoregressive prediction with time input

**Part I: Training**

**Require:** Latent trajectories $\{(x_i^{(\ell)}, t_i^{(\ell)})_{i=0}^{M-1}\}_{\ell=1}^{N_{\mathrm{traj}}}$; model $v_\omega(x,t)$; learning rate $\eta = 10^{-4}$; time step $\Delta t$; epochs $E = 50$

**for** $e = 1, \dots, E$ **do**
  **for** each consecutive pair $(x_i, x_{i+1})$ **do**
    $\delta x_i \leftarrow x_{i+1} - x_i$
    $v \leftarrow v_\omega(x_i, t_i)$
    $b \leftarrow \delta x_i / \Delta t, \quad A \leftarrow \delta x_i \delta x_i^\top / \Delta t$
    $J \leftarrow \nabla_x v_\omega(x, t_i)|_{x=x_i}$
    $\mathcal{L} \leftarrow \|v\|_2^2 - 2\langle b, v\rangle - \mathrm{Tr}(AJ)$
    $\omega \leftarrow \mathrm{Adam}(\omega, \nabla_\omega \mathcal{L}, \eta)$
  **end for**
**end for**

**Part II: Single-state autoregressive prediction**

**Require:** Initial latent state $\widehat{x}_0$ at time $t_{\mathrm{in}}$; trained $v_\omega$; decoder $S_\psi$; number of forecast steps $H$; time step $\Delta t$

$\tau_0 \leftarrow t_{\mathrm{in}}, \quad \mathcal{X}_{\mathrm{pred}} \leftarrow [\,]$
**for** $h = 0, \dots, H-1$ **do**
  $v \leftarrow v_\omega(\widehat{x}_h, \tau_h)$
  $\widehat{x}_{h+1} \leftarrow \widehat{x}_h + \Delta t\, v$
  $\tau_{h+1} \leftarrow \tau_h + \Delta t$
  $\widehat{y}_{h+1} \leftarrow S_\psi(\widehat{x}_{h+1})$
  Append $(\widehat{y}_{h+1}, \tau_{h+1})$ to $\mathcal{X}_{\mathrm{pred}}$
**end for**
**return** $\mathcal{X}_{\mathrm{pred}}$

where $x_i$ and $\hat{x}_i$ denote the ground-truth and predicted pixel values, respectively.

**PSNR.** Peak signal-to-noise ratio is computed as

$$\mathrm{PSNR} = 10 \log_{10} \left( \frac{\mathrm{MAX}_I^2}{\mathrm{MSE}} \right), \tag{40}$$

where $\mathrm{MAX}_I$ denotes the maximum possible pixel value.

**SSIM.** Structural similarity is computed as

$$\mathrm{SSIM}(x, \hat{x}) = \frac{(2\mu_x \mu_{\hat{x}} + C_1)(2\sigma_{x\hat{x}} + C_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2)}, \tag{41}$$

where $\mu$, $\sigma^2$, and $\sigma_{x\hat{x}}$ denote the mean, variance, and covariance, respectively, and $C_1$ and $C_2$ are constants for numerical stability.

**FVD.** Fréchet Video Distance compares the distributions of real and generated videos in a spatiotemporal feature space:

$$\mathrm{FVD} = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \mathrm{Tr}\left( \boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2\left(\boldsymbol{\Sigma}_r \boldsymbol{\Sigma}_g\right)^{1/2} \right), \tag{42}$$

where $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ and $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ denote the mean and covariance of the corresponding video feature distributions.

We follow the public Google Research FVD evaluation protocol using an I3D feature extractor pretrained on Kinetics-400. The same pretrained feature extractor and evaluation implementation are used for all compared methods.

The FVD implementation is taken from the public Google Research evaluation code[1], with an I3D feature extractor pretrained on Kinetics-400[2]. The corresponding pretrained I3D weights are obtained from the public evaluation code[3] and are kept fixed for all methods.

[1] https://github.com/google-research/google-research/tree/master/frechet_video_distance

[2] https://github.com/google-deepmind/kinetics-i3d

[3] https://onedrive.live.com/download?cid=78EEF3EB6AE7DBCB&resid=78EEF3EB6AE7DBCB%21199&authkey=AApKdFHPXzWLNyI