KOLMOGOROV-CHAPMAN MAPS: TRAJECTORY-FREE LEARNING FOR SIMULATING MOLECULAR DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic differential equations (SDEs) describe stochastic processes across a wide range of scientific domains, including physics, biology, and finance. Although individual numerical integration steps are inexpensive, observing meaningful events or transitions can require millions of sequential steps, resulting in substantial wall-clock simulation time and poor hardware utilization. Deep learning offers an alternative by amortizing many integration steps into a single neural network forward pass, potentially enabling much faster simulation. However, existing supervised approaches typically require long simulated trajectory data, which restricts their applicability. In this work, we introduce Kolmogorov-Chapman Maps (KC-maps), a method for learning the transition kernel of an SDE that relies only on the *trajectory-free* marginal samples together with the known drift and diffusion coefficients. Given an initial time and state of the process, KC-maps models its distribution integrated to any given time, corresponding to the weak solution of the SDE. To recover the correct dynamics, KC-maps satisfies two key properties: (i) it satisfies the boundary conditions, which we incorporate directly into the neural network architecture, and (ii) it respects the Chapman–Kolmogorov equation, which provides an efficient training objective. We empirically demonstrate that KC-maps learns to integrate the dynamics of peptide molecular systems and outperforms classical molecular dynamics in terms of GPU-hours. The implementation is available at `anonymous.4open.science/r/kcm-accel-md-anon-5E1D`.

1 INTRODUCTION

Stochastic differential equations (SDEs) provide a fundamental framework for describing stochastic processes, with applications across physics, finance, and biology. A prominent example is molecular dynamics (Leimkuhler & Matthews, 2015; Hollingsworth & Dror, 2018), where Langevin equations describe the evolution of the atomic coordinates and momenta under interatomic forces and random fluctuations. Classical approaches, such as Euler-Maruyama (Kloeden & Platen, 1992) and Langevin splitting schemes (Leimkuhler & Matthews, 2013), approximate the dynamics through a sequence of short-time updates and provide systematic control over discretization error. However, such methods are inherently sequential, with computational cost scaling with the number of integration steps. In molecular dynamics, femtosecond-scale time steps are typically required to maintain numerical stability and resolve fast motions such as bond vibrations, while conformational rearrangements and transitions between metastable states can occur over orders-of-magnitude longer timescales. Reaching these slow phenomena can therefore require an enormous number of sequential force evaluations, making long-horizon molecular simulation computationally expensive.

Learned surrogates can accelerate molecular simulation by predicting the evolution over larger time intervals. Generative modeling approaches (Klein et al., 2023a; Schreiner et al., 2023) learn conditional distributions of future molecular configurations from simulated trajectories. Other approaches (Bigi et al., 2026) learn direct deterministic updates of the positions and momenta. However, both formulations learn these transitions from pairs of states extracted from reference trajectories. Generating these pairs still requires numerical integration with time steps typically of the order of 10^{-15} s, incurring substantial computational cost to produce the training data.

The underlying SDE, however, provides a different source of supervision. Although the finite-time transition kernel is generally unavailable in closed form, the drift and diffusion already specify the

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.

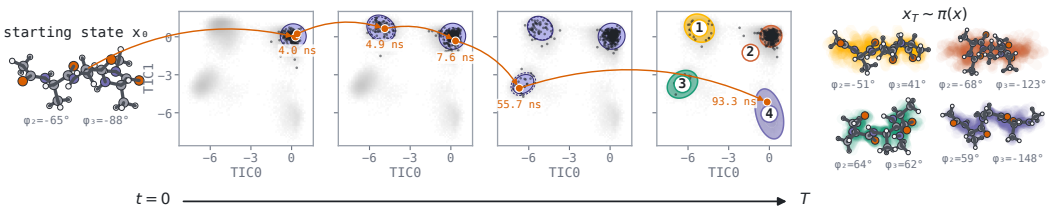


Figure 1: A single KC-maps rollout explores the landscape of metastable alanine tetrapeptide conformers. Starting from a single conformer X_0 (left), KC-maps is rolled out for 200 ns. Orange arrows trace the first visit to each newly discovered basin. Representative conformers from each basin are shown on the right, colored by basin identity, along with their backbone torsion angles (Φ_2, Φ_3).

dynamics locally. This raises the possibility of learning finite-time transitions from the governing SDE itself, without first generating reference trajectories. Recent approaches in flow-based generative modeling (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023) use consistency relations to connect the underlying vector fields with the finite-time maps (Song et al., 2023; Geng et al., 2025; Boffi et al., 2025), enabling accelerated generative sampling. Hamiltonian Flow Maps (Ripken et al., 2026) apply this consistency principle to molecular dynamics, learning Hamiltonian evolution from independent states and instantaneous force fields without trajectory supervision. For stochastic dynamics, we seek an analogous formulation at the level of conditional distributions, connecting their finite-time evolution to the known drift and diffusion of the SDE.

In this work, we present Kolmogorov-Chapman Maps (KC-maps), a trajectory-free method for learning finite-time transitions of SDEs. Our method uses *uncorrelated* samples of the initial state together with the known vector field and diffusion, without requiring trajectory supervision or a prescribed probability path. We derive a distributional consistency objective from the Kolmogorov–Chapman equation, enforcing agreement between transitions composed over different time scales; for example, five successive 2 fs transitions should agree in distribution with a single 10 fs transition. To enable tractable density evaluation and satisfy SDE boundary conditions, we parameterize transitions densities using a conditional normalizing flow with a Gaussian reference kernel. The parameterization is constrained to recover the appropriate short-time behavior of the underlying SDE, while additional stabilization mechanisms enable learning and sampling over substantially longer time intervals.

The formulation of KC-maps applies to general SDEs; here, we focus on molecular Langevin dynamics because of both its computational difficulty and its importance to applications such as drug discovery. We first conduct an empirical study on a small molecular system to identify important design choices required for stable and accurate learning, including the training objective, design of the normalizing flow, and stabilization techniques for sampling over long time intervals. We then apply KC-maps to larger scale molecular systems and highlight the following main results:

- Using only equilibrium samples and force-field evaluations, without trajectory supervision, we learn transition distributions over intervals of up to 100, fs for alanine dipeptide (AL2), tetrapeptide (AL4), and hexapeptide (AL6), replacing up to 100 sequential MD integration steps with a single learned transition.
- We demonstrate higher simulation throughput and faster exploration of conformational modes. On AL6, our method achieves approximately 2.7× aggregate simulation throughput (ns per GPU-hour) of MD on a single L40s GPU and reaches all three TICA modes within a 3 GPU-minutes whiles MD baseline still misses one mode after an hour.

2 Background

Stochastic Differential Equations. Stochastic differential equations (SDEs) describe the stochastic process X_t via the differential equation for their time-increments dX_t, i.e.

$$dX_t = v_t(X_t)\,dt + g_t(X_t)\,dW_t, \tag{1}$$

where $g_t(X_t) \in \mathbb{R}^{d\times d}$ is a state-independent diffusion matrix, and W_t is standard Brownian motion in $\mathbb{R}^d$. We will refer to the solution of the Cauchy problem of this SDE as a *transition kernel*, i.e.

given the initial condition $X_s = x$, we denote the density of the distribution of X_t as $p_{s,t}(y \mid x) \coloneqq p(X_t = y \mid X_s = x)$.

Molecular Dynamics. The formalism of SDEs is especially useful for describing the evolution of molecular systems. Their state is described by the position X_t and momentum P_t and its time-evolution is modeled by the second-order dynamics, which we will write in the matrix form

$$\begin{bmatrix} dX_t \\ dP_t \end{bmatrix} = \begin{bmatrix} P_t \\ v_t(X_t) - \gamma P_t \end{bmatrix} dt + \begin{bmatrix} 0 & 0 \\ 0 & g_t \end{bmatrix} dW_t \,, \tag{2}$$

where v_t is the force field driving the system, γ is the friction coefficient and g_t is the diffusion coefficient that depends on the temperature of the system. The stationary distribution of X_t is the Boltzmann distribution, i.e. the thermodynamical equilibrium of the molecular system. Thus, in practice, simulating this SDE for long enough time one can sample from the Boltzmann distribution describing different states — conformations — of the system.

Normalizing Flows. Generative modeling of the empirical data has two principal components: maximizing the model likelihood on the observed data and generating new samples under the learned density model. Normalizing flows satisfy both criteria by parameterizing bijective maps with tractable Jacobians. The building block of normalizing flows is the *affine coupling layer* Ψ_k, which transforms the state $x^{(k)} = [x_1^{(k)}, x_2^{(k)}]$ into $x^{(k+1)} = \Psi_k(x^{(k)})$ as

$$x_1^{(k+1)} = x_1^{(k)}, \tag{3}$$

$$x_2^{(k+1)} = x_2^{(k)} \odot \exp\left(a_k(x_1^{(k)}; \theta, c)\right) + b_k(x_1^{(k)}; \theta, c), \tag{4}$$

where $a_k(\cdot; \theta, c) : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and $b_k(\cdot; \theta, c) : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ are functions parameterized by the neural networks with parameters θ and, optionally, taking a condition c as an input. This layer is invertible, i.e. one can find $[x_1^{(k)}, x_2^{(k)}]$ from $[x_1^{(k+1)}, x_2^{(k+1)}]$, and the Jacobian of this transformation is available in a closed-form without additional computations

$$\log\det\left|\frac{\partial \Psi_k(x^{(k)})}{\partial x^{(k)}}\right| = \log\begin{bmatrix} I & 0 \\ \frac{\partial x_2^{(k+1)}}{\partial x_1^{(k)}} & I\exp\left(a_k(x_1^{(k)}; \theta, c)\right) \end{bmatrix} = \sum_i a_k(x_1^{(k)}; \theta, c)_i \,. \tag{5}$$

Finally, normalizing flows parameterize a density $p(x \mid \theta, c)$ as a push-forward map of some simple density $\pi(x)$ through the composition of affine layers $\Psi = \Psi_K \circ \ldots \circ \Psi_1$

$$\log p(x \mid \theta, c) = \log \pi(\Psi^{-1}(x; \theta, c)) - \sum_k \log\det\left|\frac{\partial \Psi_k(x^{(k)})}{\partial x^{(k)}}\right| . \tag{6}$$

3 Kolmogorov-Chapman Maps

Our method is designed around the classical definition of a diffusion process as a Markov process with continuous trajectories and infinitesimal first and second moments prescribed by the drift and diffusion coefficient (Kolmogoroff, 1931; Feller, 1937). The process defined this way is a weak solution of the corresponding SDE (Stroock & Varadhan, 1979). Therefore, our parametric model has to satisfy two conditions: (i) parameterize a valid Markov process, i.e. satisfy the Chapman-Kolmogorov equation, which we enforce through training (ii) the short-time behavior has to match the corresponding SDE, i.e. commit no jumps and have same first two moments, which we incorporate directly into our parameterization. We now proceed to formulating this statement formally.

For the SDE defined in Eq. (1), the infinitesimal change of the density $p_{s,t}(y \mid x)$ is defined by the following boundary conditions. $\forall \varepsilon > 0$, we have

$$\lim_{h\to 0} \frac{1}{h} \int_{\|y-x\|>\varepsilon} dy\, p_{s,s+h}(y \mid x) = 0 \,, \quad \texttt{(delta function with no jumps)} \tag{7}$$

$$\lim_{h\to 0} \frac{1}{h} \int_{\|y-x\|\le\varepsilon} dy\, p_{s,s+h}(y \mid x)(y-x) = v_s(x) \,, \quad \texttt{(matching first moment)} \tag{8}$$

$$\lim_{h\to 0} \frac{1}{h} \int_{\|y-x\|\le\varepsilon} dy\, p_{s,s+h}(y \mid x)(y-x)(y-x)^\top = g_s g_s^\top \,, \quad \texttt{(matching second moment)} \tag{9}$$

i.e. for $t \to s$, the density converges to the delta-function and makes no jumps, the change of the first moment equals the vector field and the change of the second moment equals the diffusion coefficient. Furthermore, any Markov transition kernel satisfies the Chapman-Kolmogorov equation

$$p_{s,t}(y \mid x) = \int dz\, p_{r,t}(y \mid z)\, p_{s,r}(z \mid x)\,. \tag{10}$$

This equation is the stochastic analog of the semi-group consistency condition in the ODEs, which is used in the Flow Maps literature (Boffi et al., 2025; Woo et al., 2026).

These two conditions define the SDE Eq. (1) and vice versa, the weak solution of an SDE satisfies both these conditions. The first statement we prove in Section A.1 using the classical result of probability theory (Stroock & Varadhan, 1979), the second statement is a text-book statement.

Proposition 1 (Identification of the transition kernel)**.** *Assume that g_s is continuous in s and that $v_s(x)$ is continuous, locally Lipschitz in x, and of at most linear growth, uniformly for s in compacts. Let $p_{s,t}(y \mid x)$ be a measurable family of nonnegative, normalized conditional densities. Suppose it satisfies the boundary conditions in Eqs. (7) to (9), with limits locally uniform in (s, x), and the Chapman–Kolmogorov equation in Eq. (10). Then $p_{s,t}(y \mid x)$ coincides (almost everywhere) with the transition density of the SDE in Eq. (1). Conversely, the SDE's transition kernel satisfies both the boundary conditions and Eq. (10).*

3.1 Kolmogorov–Chapman Constraints (Our Objective)

We parameterize our transition kernel $q_{s,t}(y \mid x; \theta)$ such that the boundary conditions are satisfies (see the following section), which leaves the Chapman-Kolmogorov equation as the sole optimization objective. The most straightforward way to train the model would be to minimize the discrepancy between both sides of the Chapman-Kolmogorov equation for the parameterized model, i.e.

$$\mathcal{L}_{\mathrm{KC}}(\theta) = D_{\mathrm{KL}}\left(\int dz\, q_{r,t}(y \mid z; \theta) q_{s,r}(z \mid x; \theta) \,\middle\|\, q_{s,t}(y \mid x; \theta) \right), \forall\, x, t, s, r\,. \tag{11}$$

However, as written, this function does not have an unbiased estimator of the gradient. Therefore, we propose to use the "stop-gradient trick" ubiquitously employed in the generative modeling literature (Geng et al., 2025; Boffi et al., 2025). Setting $\eta = \mathrm{stopgrad}(\theta)$ as the parameters in the first argument of the KL-divergence, we get

$$\mathcal{L}_{\mathrm{KC}}(\theta) = -\mathbb{E}_y[\log q_{s,t}(y \mid x; \theta)]\,, \text{ where } y \sim q_{r,t}(y \mid z; \eta)\,, \text{ and } z \sim q_{s,r}(z \mid x; \eta)\,. \tag{12}$$

This objective, along with the parameterization that satisfies the boundary conditions, has its global optimum in the transition kernel of the SDE Eq. (1), which we formalize in the following proposition relying on the classical result in stochastic processes.

Probability-flow objective. An alternative objective follows from the forward Kolmogorov equation written as a probability-flow ODE. Using the model's transition score, we define the velocity

$$v_{\mathrm{PF}}(y) = v_t(y) - \frac{1}{2} g_t g_t^\top \nabla_y \log q_{s,t}(y \mid x; \theta). \tag{13}$$

For a generated sample y_θ, define the average velocity $u_\theta = (y_\theta - x)/\tau$. Since $\frac{dy_\theta}{d\tau} = u_\theta + \tau \frac{\partial u_\theta}{\partial \tau}$, using the same "stop-gradient trick" and matching the sample velocity to v_{PF} gives

$$\mathcal{L}_{\mathrm{PF}}(\theta) = \mathbb{E}_y\left[\left\| u_\theta - \mathrm{stopgrad}\left(v_{\mathrm{PF}}(y_\theta) - \tau \frac{du_\theta}{d\tau} \right) \right\|^2 \right], \qquad y \sim q_{s,t}(y \mid x; \theta) \;\text{ and }\; \forall x,\; s < t. \tag{14}$$

However, we use the KC-KL objective in Eq. (12) as our primary training objective and compare it with the probability-flow objective in our experiments.

3.2 Boundary Constraints (Our Parameterization)

We parameterize our transition kernel $q_{s,t}(y \mid x; \theta)$ as a normalizing flow with time-scaled prior and time-scaled affine coupling layers. This guarantees the boundary constraints for $t \to s$ and puts no restrictions on larger time scales. For simplicity, we consider the autonomous SDE with the drift $v(x)$ and diffusion coefficient $gg^\top$, then our model is

$$y = x + \tau v(x) + \Psi(\sqrt{\tau} \cdot g z; x, \theta)\,, \quad z \sim \mathcal{N}(z \mid 0, I)\,, \tag{15}$$

where $\tau = t - s$ is the new time variable for the autonomous system and $\Psi(\cdot; x, \theta)$ is the normalizing flow forward map. Importantly, we add *time-gates* to the definition of the affine coupling layers, which then become defined as

$$x_1^{(k+1)} = x_1^{(k)}, \tag{16}$$

$$x_2^{(k+1)} = x_2^{(k)} \odot \exp\left(\frac{(\omega\tau)^{n_a}}{1+(\omega\tau)^{n_a}} \cdot a_k(x_1^{(k)}; x, \theta)\right) + \frac{(\omega\tau)^{n_b}}{1+(\omega\tau)^{n_b}} \cdot b_k(x_1^{(k)}; x, \theta)\,, \tag{17}$$

where ω is a learnable parameter, n_a and n_b are hyperparameters of the gates. The key property of the gates is that they behave as $(\omega\tau)^n$ for small τ, which controls the boundary conditions and converge to 1 for large τ, which implies full expressivity of the model at large time-scales.

Proposition 2. (Time-gates satisfy the boundary constraints) *Let $q_{s,t}(y \mid x; \theta)$ be the density of the random variable y defined in Eq. (15), and the map $\Psi = \Psi_K \circ \ldots \circ \Psi_1$ consists of the composition of affine coupling layers with time-gates of powers n_a and n_b. Assuming that a_k and b_k are uniformly bounded over their inputs and sufficiently small $\tau > 0$, for*

$$n_a > \frac{1}{2}\,, \text{ and } n_b > 1\,, \tag{18}$$

$q_{s,t}(\cdot \mid x; \theta)$ satisfies the boundary conditions in Eqs. (7) to (9). See proof in Section A.2.

Second-order dynamics. For second-order Langevin dynamics, we extend the parameterization to the joint source state $x = (X, P)$,

$$y = x + m(x, \tau) + \Psi(L(\tau)z; x, \theta), \quad z \sim \mathcal{N}(z \,|\, 0, I)\,, \tag{19}$$

where $m(x, \tau)$ and $L(\tau)L(\tau)^\top$ are mean displacement and covariance of the Gaussian reference obtained holding the vector field at $v(X)$. The expressions for m and L are derived in Section B.3.2.

The coupling layers in Eq. (17) alternate between position and momentum updates. We use $n_a = 2$ for both updates and $n_b = 3$ for position and $n_b = 2$ for momentum. This choice of gate orders preserves the boundary constraints for the second-order SDE.

3.3 Practical Design Choices

We use the design choices listed below to improve training stability and accuracy of learned transitions.

Stabilizing the Gaussian reference. To stabilize sampling over longer intervals, we limit the transport and vector-field contributions to the reference mean using $\lambda(\tau) = h(\tau)/\tau$. We set $h(\tau) = \tau$ for $\tau \leq \tau_0$ and $h(\tau) = \tau_0 + \delta \tanh((\tau - \tau_0)/\delta)$ otherwise, where $\tau_0 > 0$ sets the interval over which the reference is unchanged and $\delta > 0$ controls the transition to the maximum duration $\tau_0 + \delta$. Since $\lambda(\tau) = 1$ near zero, the reference retains its boundary conditions. Momentum damping and covariance are evaluated at the full elapsed time τ.

Furthermore, two-step targets in Eq. (11) are formed from shorter transitions, motivating a more accurate short-time reference. For source position X and momentum P, we evaluate the vector field at $v(X_{\text{mid}})$, where $X_{\text{mid}} = X + \frac{1}{2}h(\tau)P$. The schedule and midpoint derivation are detailed in Section B.3. We ablate duration capping and midpoint evaluation separately in Table 1a.

Time prior. During training, we sample elapsed times from $\tau \sim w\,\texttt{LogUniform}[T_{\min}, T_{\max}] + (1 - w)\,\texttt{Uniform}[T_{\min}, T_{\max}]$, where $T_{\min}$ and $T_{\max}$ are the minimum and maximum sampled elapsed times, with $0 < T_{\min} < T_{\max}$ and $w \in [0, 1]$. The uniform component upweights longer intervals, while $T_{\max}$ is gradually increased early in training so the model first learns shorter transitions. We compare this mixture against either distribution alone in Table 1b.

Input features. Alongside the source state and elapsed time τ, we condition the flow on the Gaussian reference mean displacement and covariance. Flow inputs are normalized by the reference marginal standard deviations for predicting affine scale and shift, while transformations act on the original inputs. We separately ablate mean and covariance features and input normalization in Table 1c.

Normalizing flow design. The learned rate ω controls the elapsed time at which the flow corrections become active, without changing their prescribed short-time orders. We compare learning this rate with fixing it at its initialization, $\omega = 400\,\text{ps}^{-1}$. For second-order dynamics, we also compare our gate choices with a lower-order alternative, labeled "Lower-order gates," which uses $n_a = 1$ for both log scales and $n_b = 2$ for both shifts, as shown in Table 1d. Finally, we compare the displacement

Table 1: **Ablation study on AL2.** NLL on samples from MD simulation and energy $\mathcal{E}$-$\mathcal{W}_2$ ($k_\mathrm{B}T$) over 64 source states with 256 samples per state compared to 256 independent MD trajectories with a 0.1 fs timestep. Results are averaged over $\tau = 10, 20, \ldots, 100$ fs, with lower values being better. Each subtable compares the full model with variants that change only the indicated component, keeping all other settings fixed. All models were trained with a reduced compute budget of 250K steps. Best value in each column within each subtable is **bold**.

(a) **Gaussian reference.**

Variant	NLL ↓	$\mathcal{E}$-$\mathcal{W}_2$ ↓
Full model	$\mathbf{-0.122}$	$\mathbf{1.33}$
w/o duration capping	0.350	9.7×10^3
w/o midpoint evaluation	−0.053	4.71

(b) **Time prior.**

Variant	NLL ↓	$\mathcal{E}$-$\mathcal{W}_2$ ↓
Full model	$\mathbf{-0.122}$	$\mathbf{1.33}$
Log-uniform only	−0.086	1.37
Uniform only	−0.087	2.64

(c) **Input features.**

Variant	NLL ↓	$\mathcal{E}$-$\mathcal{W}_2$ ↓
Full model	$\mathbf{-0.122}$	1.33
w/o mean features	−0.056	1.34
w/o covariance features	−0.094	1.45
w/o residual normalization	−0.106	$\mathbf{1.28}$

(d) **Normalizing flow design.**

Variant	NLL ↓	$\mathcal{E}$-$\mathcal{W}_2$ ↓
Full model	$\mathbf{-0.122}$	1.33
w/o source-frame alignment	0.028	1.77
Lower-order gates	−0.107	$\mathbf{1.25}$
Fixed rate ω	−0.115	1.56

parameterization used here with standard and whitened flows on AL2. We briefly describe these alternatives and their corresponding gates in Section E.2, with results in Fig. A2.

Equivariance. To account for global translation and rotation symmetries, we center positions at the center of mass and apply the flow in an orthonormal frame determined by the source configuration (Kaba et al., 2023; Jumper et al., 2021). We rotate the output back to the original orientation. The frame construction and treatment of the center-of-mass constraint are detailed in Section C.

Stabilizing training. To stabilize learning from model-generated targets, we use $\eta = \mathrm{EMA}(\theta)$ in $\mathcal{L}_\mathrm{KC}$. We compare different EMA decay rates, including $\eta = \mathrm{stopgrad}(\theta)$, in Fig. A1. Algorithm 1 summarizes the training procedure.

Inference-time stabilization. Following the use of inference-time corrections in HFM (Ripken et al., 2026), we apply additional corrections to limit accumulated energy errors during long simulations. We describe these corrections in Section C.2 and summarize one simulation step in Algorithm 2.

4 Trajectory-Free Simulation of Molecular Dynamics

4.1 Baselines

Throughout our experiments, we primarily consider trajectory-free methods at the target distribution in our comparison. We also compare against numerical Langevin integration as the standard simulation baseline. Reference simulations provide the transition samples and equilibrium statistics used for evaluation. For all experiments, we assume we have access to the initial states $x \sim \pi(x)$. We match training FLOPs between KC-KL and PF-ODE on Lennard-Jones systems and between our method and HFM on AL4 and AL6. For AL2, we use the released HFM checkpoint.

4.2 Lennard-Jones

To demonstrate our method to learn the map for first-order SDE, we consider the Lennard-Jones 13-particle (LJ-13) and 55-particle (LJ-55) systems, building on the setup in (Akhound-Sadegh et al., 2025). We train our flow to model transitions of particle positions at the target temperature $T = 1$ and diffusion coefficient $\frac{\sigma^2}{2} = 0.02$. We compare on two different training settings. The first uses configurations sampled at the target temperature for the initial state during training. The

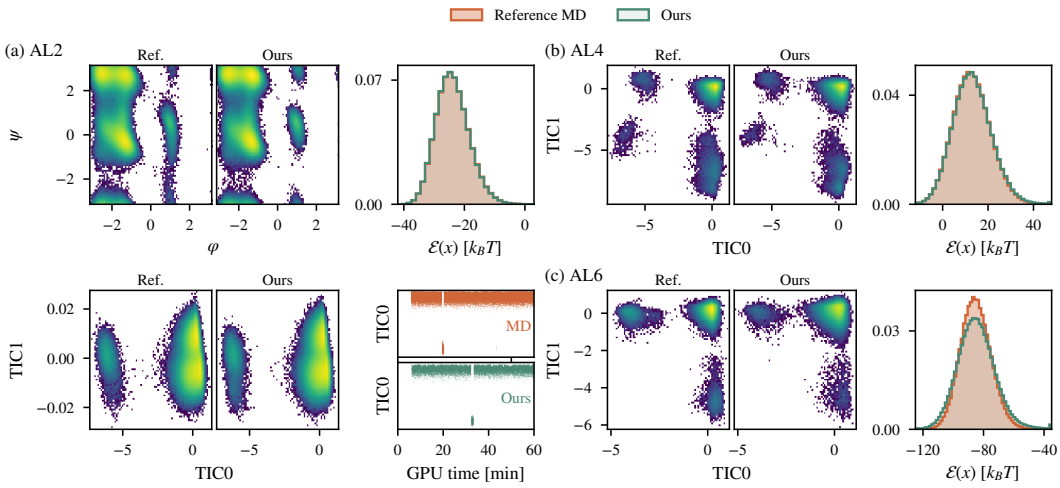


Figure 2: **Equilibrium sampling of alanine peptides.** Density panels show the reference MD (Ref.) beside KC-maps (Ours). Potential energy histograms of KC-maps against each system's MD reference. (a) AL2: Ramachandran and TICA densities, energy histogram, and TIC0 against simulated MD trajectory (top) and ours (bottom). (b) AL4: TICA densities and energy histogram. (c) AL6: TICA densities and energy histogram. For each molecule, samples were generated from KC-maps under the one GPU hour compute budget.

Table 2: Performance of KC-maps on LJ-13 and LJ-55. Published baseline results are taken from Potaptchik et al. (2025). Metrics are the 2-Wasserstein distance between energy distributions ($\mathcal{E}$-$\mathcal{W}_2$) and the geometric 2-Wasserstein distance (Geo-$\mathcal{W}_2$). In all settings, the target temperature is $T = 1$; best values are **bolded** and the second best values are underlined. Dashes indicate unavailable results.

	LJ-13		LJ-55	
Method	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	Geo-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	Geo-$\mathcal{W}_2 \downarrow$
MALA	0.325	**1.51**	3.40	**3.80**
Source temperature $T_S = 4$				
PITA (Akhound-Sadegh et al., 2025)	2.260	1.65	–	–
ITM (Potaptchik et al., 2025)	0.879	<u>1.54</u>	26.37	3.98
KC-maps ($\mathcal{L}_{\text{KC}}$)	**0.212**	**1.51**	<u>1.51</u>	<u>3.82</u>
Source temperature $T_S = 1$				
KC-maps ($\mathcal{L}_{\text{KC}}$)	0.266	**1.51**	**1.19**	<u>3.82</u>
KC-maps ($\mathcal{L}_{\text{PF}}$)	<u>0.254</u>	**1.51**	140.37	4.29

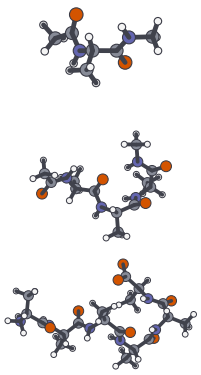
second uses configurations sampled from a higher temperature $T = 4$, motivated by what is done in (Akhound-Sadegh et al., 2025; Potaptchik et al., 2025). In both settings, the vector field and diffusion correspond to the SDE at the target temperature. This comparison demonstrates that our method can learn the target dynamics without requiring training configurations from the target distribution.

We evaluate 2-Wasserstein distances on 1D energy histograms ($\mathcal{E}$-$\mathcal{W}_2$) and 3D coordinates (Geo-$\mathcal{W}_2$). We run Metropolis-adjusted MCMC using our learned kernel (Hastings, 1970) and compare against MALA (Roberts & Tweedie, 1996), selecting its step size by sweep ($h = 3 \times 10^{-4}$ for LJ-13 and $h = 10^{-4}$ for LJ-55). Our method uses $\tau = 0.05$ for LJ-13 and $\tau = 0.02$ for LJ-55. All experiments use 5000 walkers for 5000 steps. Results are reported in Table 2.

4.3 Molecular Systems

We evaluate our second-order parameterization on three molecular systems of increasing size: alanine dipeptide (AL2), tetrapeptide (AL4), and hexapeptide (AL6). We train the flow to model transitions of the joint state of positions and momenta under the second-order Langevin dyanmics, using *uncorrelated* equilibrium configurations and the molecular force field without requiring paired

Table 3: Equilibrium sampling of alanine peptides per method with a fixed budget of one GPU-hour on single L40S, scored against each molecule's long MD. $\mathcal{W}_2$ metrics are computed between 10^4 reference frames and 10^4 frames drawn from the run after a 10 % burn-in, and report the mean $\pm$ std. over 3 draws. Best of HFM and KC-maps in **bold**, second underlined.



	Method	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathcal{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$
AL2 ($d = 66$)	MD	0.13 ± 0.01	0.12 ± 0.01	0.27 ± 0.02
	HFM	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.11 \pm 0.02}$	0.22 ± 0.09
	KC-maps	0.16 ± 0.03	0.35 ± 0.01	$\mathbf{0.16 \pm 0.13}$
AL4 ($d = 126$)	MD	0.24 ± 0.10	0.53 ± 0.01	0.69 ± 0.03
	HFM	$\mathbf{0.26 \pm 0.03}$	$\mathbf{0.74 \pm 0.01}$	0.83 ± 0.02
	KC-maps	0.60 ± 0.08	1.06 ± 0.02	$\mathbf{0.58 \pm 0.04}$
AL6 ($d = 189$)	MD	0.27 ± 0.05	1.03 ± 0.01	0.82 ± 0.00
	HFM	$\mathbf{0.76 \pm 0.06}$	$\mathbf{1.54 \pm 0.02}$	0.82 ± 0.04
	KC-maps	2.19 ± 0.24	2.29 ± 0.01	$\mathbf{0.56 \pm 0.04}$

transition states. We generate the trajectories by sequentially applying the learned transition kernel at a large timestep τ, and we compare the resulting equilibrium distribution against HFM (Ripken et al., 2026) and molecular dynamics simulated using BAOAB scheme (Leimkuhler & Matthews, 2013).

We use $\mathcal{L}_{\mathrm{KC}}$ for all peptide experiments, as training with $\mathcal{L}_{\mathrm{PF}}$ remained unstable even with adaptive loss weighting (Geng et al., 2025; Woo et al., 2026). The poor performance on LJ-55 for $\mathcal{L}_{\mathrm{PF}}$ in Table 2 further supports this choice.

We evaluate all methods under an equal budget of one GPU hour. For each method, we vary the number of parallel chains and, for learned transitions, the elapsed time τ, and report the best-performing configuration on average across all three metrics. We compare the sampled distributions against long MD reference trajectories. The evaluated settings and reference simulations are detailed in Section D.4.

We measure the 2-Wasserstein distance between distributions of potential energy ($\mathcal{E}$-$\mathcal{W}_2$), torsion angles ($\mathcal{T}$-$\mathcal{W}_2$), and first two component projections from time-lagged independent component analysis (TICA-$\mathcal{W}_2$), as done in (Tan et al., 2025; 2026). The metrics capture complementary aspects of the quality of the sampled distribution. For example, $\mathcal{E}$-$\mathcal{W}_2$ assesses the accuracy of local geometry while $\mathcal{T}$-$\mathcal{W}_2$ and TICA-$\mathcal{W}_2$ assess coverage of metastable states. We report quantitative results Table 3 and compare the sampled distributions with the reference simulations in Fig. 2.

While MD and HFM perform well on the energy and torus metrics in Table 3, these metrics can mask incomplete mode coverage. Because most AL4 and AL6 reference samples lie in the dominant mode, failing to explore less populated modes incurs only a small penalty. We therefore also measure how efficiently each method explores conformational space under a fixed compute budget. As shown in Fig. 3, KC-maps achieves both higher simulation throughput and faster exploration, reaching all reference modes in fewer GPU-minutes than MD and HFM.

5 Related Work

Boltzmann generators (Noé et al., 2019; Klein & Noé, 2024; Tan et al., 2025; 2026) directly sample uncorrelated equilibrium configurations without integrating MCMC. However, equilibrium distributions alone do not capture dynamical properties such as relaxation times and transition rates (Dixit & Dill, 2014), motivating surrogates that model time-coarsened MD transitions.

Hamiltonian Flow Maps (Ripken et al., 2026) learn deterministic updates from Hamiltonian equations using uncorrelated states and forces, and apply a separate thermostat update during simulation as done in the OBABO scheme (Bussi & Parrinello, 2007). Similarly, we leverage local physical information, but learn the stochastic transition kernel of Langevin dynamics.

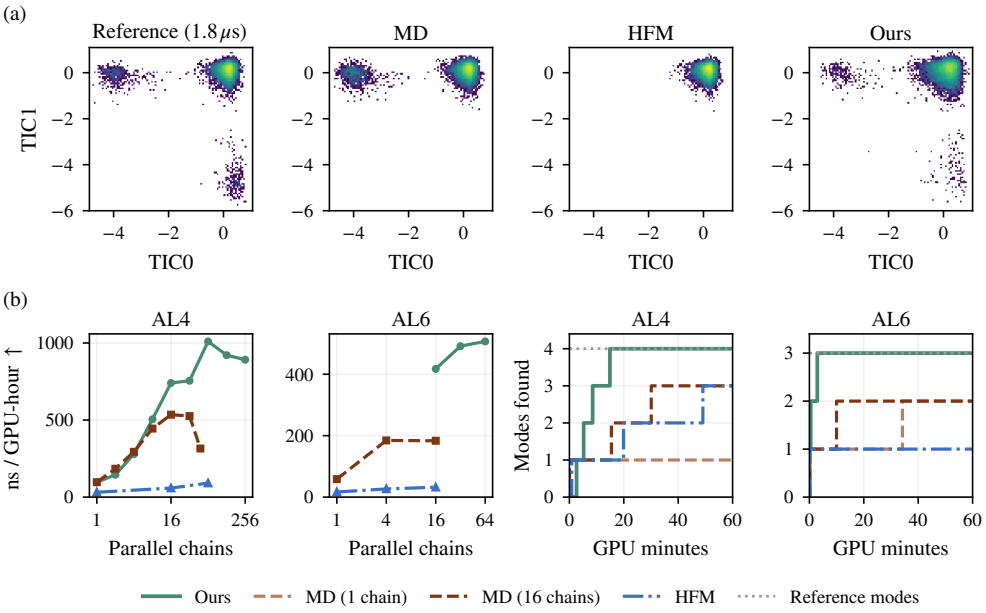


Figure 3: **One GPU-hour on one L40S.** (a) TICA densities of the reference (1.8 μs test split), MD (16 parallel chains), HFM (5 fs) and KC-maps ($\tau = 40$ fs, 64 chains). Only KC-maps visits both minor modes. (b) Left: simulated time per GPU-hour against the number of parallel chains for KC-maps ($\tau = 50$ fs for AL4, $\tau = 40$ fs for AL6), MD and HFM (7 fs for AL4, 5 fs for AL6). Right: metastable TICA modes discovered against GPU time. Dotted: modes of the reference MD.

Consistency models (Song et al., 2023; Lu & Song, 2025) and Flow Map methods (Boffi et al., 2024; Geng et al., 2025; Boffi et al., 2025) learn finite-time maps of generative ODEs. Stochastic Flow Maps (SFMs) (Potaptchik et al., 2026; Holderrieth et al., 2026) extend this framework to conditional distributions, while subsequent works learn pathwise SDE solution maps conditioned on Brownian motion (McCallum et al., 2026; Pan et al., 2026). Our method falls within the SFM framework, directly parameterizing the transition kernel and enforcing its composition through the Kolmogorov–Chapman equation.

Lastly, similar works like Neural Stochastic Flows (NSF) (Kiyohara et al., 2025) also parameterize the SDE transition kernel with a conditional normalizing flows. However, their objective combines likelihood on *observed* transition pairs with a variational upper bound on bidirectional KL regularization evaluated through a learned bridge model. KC-maps learns from unpaired states and the known SDE drift and diffusion. Unlike NSF, our parameterization preserves the prescribed infinitesimal dynamics, while the consistency objective uses transitions generated by composing the model itself. This allows us to learn finite-time transition distributions without observed future states.

6 Conclusion

Our work introduces KC-maps, which provides a trajectory-free approach to learning transition kernels of SDEs. Our method combines a parameterization that preserves the infinitesimal dynamics with a training objective based on the Kolmogorov–Chapman consistency, using *uncorrelated* initial states and the known drift and diffusion. On peptide molecular systems, we learn transitions of up to 100 fs and demonstrate improved equilibrium sampling efficiency compared with classical molecular dynamics, combining higher simulation throughput with broader conformational coverage at matched GPU-hour budgets.

For future work, a natural next step is to evaluate whether KC-maps reproduce dynamical observables beyond the equilibrium distribution, such as transition rates and relaxation times. Further work could extend the approach to larger molecular systems and larger step sizes, and amortize over systems to enable transfer to unseen molecules. Moreover, our Lennard-Jones results also motivate extending training from higher-temperature data to molecular systems, without requiring equilibrium samples at the target temperature.

AI USE STATEMENT

Large language models supported code development and debugging, manuscript formatting, and literature searches. The authors developed the method, designed the experiments, and reviewed the code, results, and cited sources. We take full responsibility for the paper's content.

REPRODUCIBILITY STATEMENT

We provide code at anonymous.4open.science/r/kcm-accel-md-anon-5E1D. We also fully detail the training implementation and evaluation setup with details on hyperparameters in Section C. Proofs of theoretical results are in Section A and derivations in Section B.

REFERENCES

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated Denoising Energy Matching for Sampling from Boltzmann Densities, February 2024. URL http://arxiv.org/abs/2402.06121. arXiv:2402.06121 [cs, stat].

Tara Akhound-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M. Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities, 2025. URL https://arxiv.org/abs/2506.16471.

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants, March 2023. URL http://arxiv.org/abs/2209.15571. arXiv:2209.15571 [cs].

Filippo Bigi, Sanggyu Chong, Agustinus Kristiadi, and Michele Ceriotti. Flashmd: long-stride, universal prediction of molecular dynamics, 2026. URL https://arxiv.org/abs/2505.19350.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow Map Matching, June 2024. URL http://arxiv.org/abs/2406.07507. arXiv:2406.07507 [cs, math].

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, October 2025. URL http://arxiv.org/abs/2505.18825. arXiv:2505.18825 [cs].

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

Giovanni Bussi and Michele Parrinello. Accurate sampling using langevin dynamics. *Physical Review E*, 75(5):056707, 2007. doi: 10.1103/PhysRevE.75.056707.

Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *The Journal of Chemical Physics*, 126(1):014101, 2007.

Purushottam D. Dixit and Ken A. Dill. Inferring microscopic kinetic rates from stationary state distributions. *Journal of Chemical Theory and Computation*, 10(8):3002–3005, 2014. doi: 10.1021/ct5001389.

Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, Kyle A. Beauchamp, Lee-Ping Wang, Andrew C. Simmonett, Matthew P. Harrigan, Chaya D. Stern, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Computational Biology*, 13(7):e1005659, 2017.

Willy Feller. Zur theorie der stochastischen prozesse: Existenz-und eindeutigkeitssätze. *Mathematische Annalen*, 113(1):113–160, 1937.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling, May 2025. URL http://arxiv.org/abs/2505.13447. arXiv:2505.13447 [cs].

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv:2006.11239 [cs, stat].

Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Moritz Hoffmann, Martin Konrad Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke Elena Husic, Stefan Klus, Hao Wu, J Nathan Kutz, Steven Brunton, and Frank Noé. Deeptime: a Python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 3(1):015009, 2022.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond Maps: Efficient Reward Alignment via Stochastic Flow Maps, April 2026. URL http://arxiv.org/abs/2602.05993. arXiv:2602.05993 [cs] version: 2.

Scott A. Hollingsworth and Ron O. Dror. Molecular Dynamics Simulation for All. *Neuron*, 99(6): 1129–1143, September 2018. doi: 10.1016/j.neuron.2018.08.011. URL https://doi.org/10.1016/j.neuron.2018.08.011. PMID: 30236283; PMCID: PMC6209097.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, July 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions, 2023. URL https://arxiv.org/abs/2211.06489.

Samuel Karlin and Howard Taylor. *A second course in stochastic processes*. Elsevier, 1981.

Naoki Kiyohara, Edward Johns, and Yingzhen Li. Neural stochastic flows: Solver-free modelling and inference for sde solutions, 2025. URL https://arxiv.org/abs/2510.25769.

Leon Klein and Frank Noé. Transferable Boltzmann Generators, June 2024. URL http://arxiv.org/abs/2406.14426. arXiv:2406.14426 [physics, stat].

Leon Klein, Andrew Y. K. Foong, Tor Erlend Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics, February 2023a. URL http://arxiv.org/abs/2302.01170. arXiv:2302.01170 [cond-mat, physics:physics, stat].

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching, June 2023b. URL http://arxiv.org/abs/2306.15030. arXiv:2306.15030 [physics, stat].

Peter E. Kloeden and Eckhard Platen. Numerical solution of stochastic differential equations. 1992. URL https://api.semanticscholar.org/CorpusID:116903781.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pp. 5361–5370, 2020.

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Andrei Kolmogoroff. Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104(1):415–458, 1931.

Benedict Leimkuhler and Charles Matthews. Robust and efficient configurational molecular sampling via langevin dynamics. *The Journal of chemical physics*, 138(17), 2013.

Benedict Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, volume 39 of *Interdisciplinary Applied Mathematics*. Springer International Publishing, 2015. ISBN 978-3-319-16375-8. doi: 10.1007/978-3-319-16375-8. URL https://doi.org/10.1007/978-3-319-16375-8.

Kresten Lindorff-Larsen, Stefano Piana, Kim Palmo, Paul Maragakis, John L. Klepeis, Ron O. Dror, and David E. Shaw. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Structure, Function, and Bioinformatics*, 78(8):1950–1958, 2010.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747. arXiv:2210.02747 [cs, stat].

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models, 2025. URL https://arxiv.org/abs/2410.11081.

James A. Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E. Hauser, and Carlos Simmerling. ff14SB: Improving the accuracy of protein side chain and backbone parameters from ff99SB. *Journal of Chemical Theory and Computation*, 11(8):3696–3713, 2015.

Sam McCallum, Zander W. Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong stochastic flow maps, 2026. URL https://arxiv.org/abs/2606.01086.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning, July 2019. URL http://arxiv.org/abs/1812.01729. arXiv:1812.01729 [stat].

Alexey Onufriev, Donald Bashford, and David A. Case. Exploring protein native states and large-scale conformational changes with a modified generalized Born model. *Proteins: Structure, Function, and Bioinformatics*, 55(2):383–394, 2004.

Zhengkai Pan, Peter Potaptchik, Wenxi Yao, Michael S. Albergo, and Jakiw Pidstrigach. Itô maps for any-step sdes, 2026. URL https://arxiv.org/abs/2606.11156.

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics*, 139(1):015102, 2013.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning, 2025. URL https://arxiv.org/abs/2512.21829.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S. Albergo, and Yee Whye Teh. Meta Flow Maps enable scalable reward alignment, January 2026. URL http://arxiv.org/abs/2601.14430. arXiv:2601.14430 [stat].

Winfried Ripken, Michael Plainer, Gregor Lied, Thorben Frank, Oliver T. Unke, Stefan Chmiela, Frank Noé, and Klaus-Robert Müller. Learning hamiltonian flow maps: Mean flow consistency for large-timestep molecular dynamics, 2026. URL https://arxiv.org/abs/2601.22123.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics, May 2023. URL http://arxiv.org/abs/2305.18046. arXiv:2305.18046 [physics, stat].

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. URL http://arxiv.org/abs/2011.13456. arXiv:2011.13456 [cs].

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models, May 2023. URL http://arxiv.org/abs/2303.01469. arXiv:2303.01469 [cs].

Daniel W. Stroock and Srinivasa Varadhan. Multidimensional diffusion processes. 1979. URL https://api.semanticscholar.org/CorpusID:118022077.

Charlie B. Tan, Avishek Joey Bose, Chen Lin, Leon Klein, Michael M. Bronstein, and Alexander Tong. Scalable Equilibrium Sampling with Sequential Boltzmann Generators, February 2025. URL http://arxiv.org/abs/2502.18462. arXiv:2502.18462 [cs].

Charlie B. Tan, Majdi Hassan, Leon Klein, Saifuddin Syed, Dominique Beaini, Michael M. Bronstein, Alexander Tong, and Kirill Neklyudov. Amortized sampling with transferable normalizing flows, 2026. URL https://arxiv.org/abs/2508.18175.

Dongyeop Woo, Marta Skreta, Seonghyun Park, Kirill Neklyudov, and Sungsoo Ahn. Riemannian meanflow. *arXiv preprint arXiv:2602.07744*, 2026.

A PROOFS

A.1 IDENTIFICATION OF THE TRANSITION KERNEL

Here we show the proof of a classical result for the identification of the kernel (Stroock & Varadhan, 1979).

Proof of Proposition 1. Fix $s < t$ and an initial state x. Let $\tau = t - s$, set $h = \tau/n$ and $t_j = s + jh$, and define the Markov chain

$$X_s^{(n)} = x, \qquad X_{t_{j+1}}^{(n)} \mid X_{t_j}^{(n)} \sim p_{t_j,t_{j+1}}(\cdot \mid X_{t_j}^{(n)}), \qquad j = 0, \dots, n-1.$$

By repeated application of Chapman–Kolmogorov, $X_t^{(n)}$ has density $p_{s,t}(\cdot \mid x)$ for every n.

The boundary conditions in Eqs. (7) to (9) are exactly the assumption of the diffusion approximation (Stroock & Varadhan, 1979). Its remaining hypothesis, well-posedness of the martingale problem for $(v, gg^\top)$, follows from the assumptions on v and g in Proposition 1: they give a pathwise unique, non-explosive solution of Eq. (1), hence uniqueness in law, which is equivalent to well-posedness (Stroock & Varadhan, 1979). Therefore, this gives weak convergence of the linearly interpolated chains $(X_r^{(n)})_{r\in[s,t]}$ to $(X_r)_{r\in[s,t]}$, where $(X_r)_{r\in[s,t]}$ solves the SDE in Eq. (1) with initial state $X_s = x$. Note that to apply the diffusion approximation to time-dependent systems, we can simply apply the result to (r, X_r), treating time as part of the state. We first establish convergence within some bounded region where the boundary condition errors vanish uniformly as $h \to 0$. Second, since the SDE has continuous trajectories defined through $[s, t]$, the probability of leaving the bounded region tends to zero as the region extends to the full state space. Therefore, the localization argument gives convergence over the full interval.

Weak convergence thus gives, for every bounded continuous f,

$$\int dy\, f(y)\, p_{s,t}(y \mid x) = \mathbb{E}\Big[f\Big(X_t^{(n)}\Big)\Big] \longrightarrow \mathbb{E}[f(X_t) \mid X_s = x].$$

The left-hand side is independent of n, so $p_{s,t}(\cdot \mid x)$ is the SDE's transition density, up to equality almost everywhere. Since s, t, and x were arbitrary, this identifies the entire transition kernel.

Conversely, the SDE's transition kernel satisfies the boundary conditions Eqs. (7) to (9), and its Markov property gives Chapman–Kolmogorov consistency. For the proof, we refer the reader to a textbook (Karlin & Taylor, 1981). □

A.2 PROOF OF PROPOSITION 2

We use the following notation:

- $u := \sqrt{\tau}\, gz$ with $z \sim \mathcal{N}(0, I)$;
- $\gamma_n(\tau) := \frac{(\omega\tau)^n}{1+(\omega\tau)^n}$;
- $\bar{a} := \max_k \sup \|a_k\|_\infty$ and $\bar{b} := \max_k \sup \|b_k\|$ denote the uniform bounds on the conditioners (over $x_1^{(k)}$ and x);
- C denote constants depending only on K, ω, $\bar{a}$, $\bar{b}$.

Lemma 1 (Gated layers are small perturbations of the identity)**.** *For all $\tau \in (0, 1]$ and all u,*

$$\|\Psi(u) - u\| \le C(\tau^{n_a}\|u\| + \tau^{n_b})\,. \tag{20}$$

Proof. Consider one layer Eq. (17) with input $x = (x_1, x_2)$

$$\|\Psi_k(x) - x\| = \big\|x_2 \odot \big(e^{\gamma_{n_a}(\tau)a_k} - 1\big) + \gamma_{n_b}(\tau)\, b_k\big\| \le \|x_2\|\, \gamma_{n_a}(\tau)\, \bar{a}e^{\bar{a}} + \gamma_{n_b}(\tau)\bar{b}\,, \tag{21}$$

where we use $|e^c - 1| \le |c|e^{|c|}$. Then we have

$$\|\Psi_k(x) - x\| \le \|x_2\|\, \gamma_{n_a}(\tau)\, \bar{a}e^{\bar{a}} + \gamma_{n_b}(\tau)\bar{b} \le \|x\|\, (\omega\tau)^{n_a}\, \bar{a}e^{\bar{a}} + (\omega\tau)^{n_b}\bar{b} \tag{22}$$

For $x^{(0)} = u$ and $x^{(k)} = \Psi_k(x^{(k-1)})$, we have

$$\left\|x^{(k)}\right\| = \left\|\Psi_k(x^{(k-1)}) - x^{(k-1)} + x^{(k-1)}\right\| \le \left\|\Psi_k(x^{(k-1)}) - x^{(k-1)}\right\| + \left\|x^{(k-1)}\right\| \tag{23}$$

$$\le \left\|x^{(k-1)}\right\|\left((\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + 1\right) + (\omega\tau)^{n_b}\bar{b}\,. \tag{24}$$

So the norm is multiplied by $((\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + 1)$ and added to $(\omega\tau)^{n_b}\bar{b}$. Thus, we have

$$\left\|x^{(k)}\right\| \le \left((\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + 1\right)^k\left[\left\|x^{(0)}\right\| + k(\omega\tau)^{n_b}\bar{b}\right] \tag{25}$$

$$\le \left((\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + 1\right)^K\left[\left\|x^{(0)}\right\| + K(\omega\tau)^{n_b}\bar{b}\right] \tag{26}$$

Therefore

$$\|\Psi(u) - u\| \le \sum_{k=1}^{K}\left\|\Psi_k(x^{(k-1)}) - x^{(k-1)}\right\| \le \sum_{k=1}^{K}\left(\|x^{(k-1)}\|\,(\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + (\omega\tau)^{n_b}\bar{b}\right) \tag{27}$$

$$\le K\left(\left((\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + 1\right)^K\left[\|u\| + K(\omega\tau)^{n_b}\bar{b}\right](\omega\tau)^{n_a}\,\bar{a}e^{\bar{a}} + K(\omega\tau)^{n_b}\bar{b}\right) \tag{28}$$

$$\le K\,\bar{a}e^{\bar{a}}\omega^{n_a}\left(\omega^{n_a}\,\bar{a}e^{\bar{a}} + 1\right)^K\left[\tau^{n_a}\|u\| + K(\omega\tau)^{n_b}\bar{b}\right] + K^2(\omega\tau)^{n_b}\bar{b} \tag{29}$$

$$\le C[\tau^{n_a}\|u\| + \tau^{n_b}]\,, \tag{30}$$

for some constant C. □

Proof of Proposition 2. From our parameterization Eq. (15) we have

$$y - x = \tau v(x) + u + \Psi(u) - u\,. \tag{31}$$

From Lemma 1, we have

$$\|\Psi(u) - u\| \le C(\tau^{n_a}\|u\| + \tau^{n_b})\,, \tag{32}$$

and since u is Gaussian, we can write $\mathbb{E}\|u\|^m = c_m\,\tau^{m/2}$ with c_m being a constant depending only on g, m and number of dimensions, so we have

$$\mathbb{E}\|\Psi(u) - u\|^m \le C\left(\tau^{mn_a}\,\mathbb{E}\|u\|^m + \tau^{mn_b}\right) = C\left(\tau^{m(n_a+1/2)} + \tau^{mn_b}\right) = O(\tau^{m\beta})\,, \tag{33}$$

where we denote $\beta \coloneqq \min\{n_a + \frac{1}{2},\ n_b\}$, and note that by the condition $n_a > \frac{1}{2}$ and $n_b > 1$, we have $\beta > 1$.

The first moment is then

$$\mathbb{E}[y - x] = \tau v(x) + \mathbb{E}u + \mathbb{E}[\Psi(u) - u] = \tau v(x) + O(\tau^\beta)\,. \tag{34}$$

The second moment we write down using $r \coloneqq \Psi(u) - u$. That is

$$\mathbb{E}[(y-x)(y-x)^\top] = \tau^2 vv^\top + \tau gg^\top + \mathbb{E}[rr^\top] + \tau\left(v\,\mathbb{E}[u+r]^\top + \mathbb{E}[u+r]\,v^\top\right) + \tag{35}$$

$$+ \mathbb{E}[ur^\top] + \mathbb{E}[ru^\top]\,. \tag{36}$$

Furthermore, using $\mathbb{E}u = 0$ and Cauchy-Schwarz, we have

$$\left\|\mathbb{E}[ur^\top]\right\| \le \sqrt{\mathbb{E}\|u\|^2\mathbb{E}\|r\|^2} = O(\tau^{1/2+\beta}) \tag{37}$$

$$\left\|\mathbb{E}[rr^\top]\right\| \le \mathbb{E}\|r\|^2 = O(\tau^{2\beta}) \tag{38}$$

$$\left\|v\,\mathbb{E}[u+r]^\top\right\| = \left\|v\,\mathbb{E}r^\top\right\| \le \sqrt{\mathbb{E}\|v\|^2\mathbb{E}\|r\|^2} = O(\tau^\beta)\,. \tag{39}$$

Since $\beta > 1$, we, finally, have

$$\mathbb{E}[(y-x)(y-x)^\top] = \tau gg^\top + o(\tau)\,. \tag{40}$$

To prove the no jump condition in Eq. (7), we can use Markov's inequality i.e., for every $\varepsilon > 0$,

$$\mathbb{P}\left(\|y - x\| > \varepsilon\right) \le \frac{\mathbb{E}\|y-x\|^4}{\varepsilon^4} \le \frac{1}{\varepsilon^4}27\left(\tau^4\|v(x)\|^4 + \mathbb{E}\|u\|^4 + \mathbb{E}\|r\|^4\right) = O(\tau^2)\,, \tag{41}$$

where we use $(a+b+c)^4 \le 27(a^4+b^4+c^4)$. Thus, we show $\mathbb{P}\left(\|y-x\| > \varepsilon\right)/\tau \to 0$. Finally, we show that the truncation does not matter for the moments $k \in \{1, 2\}$, by Cauchy–Schwarz,

$$\mathbb{E}\left[\|y-x\|^k\,\mathbf{1}\{\|y-x\| > \varepsilon\}\right] \le \sqrt{\mathbb{E}\|y-x\|^{2k}}\,\sqrt{\mathbb{P}(\|y-x\| > \varepsilon)} = O(\tau^{k/2})\,O(\tau) = o(\tau)\,.$$

□

A.3 Global minimizers of the KC divergence

Proof of global minimizer. Here, $\mathcal{L}_{\mathrm{KC}}$ denotes the KL divergence in Eq. (11), averaged over the sampled source states and time triples. KL divergence is nonnegative, so $\mathcal{L}_{\mathrm{KC}}(\theta) \geq 0$. The SDE's transition kernel p satisfies Chapman–Kolmogorov consistency and belongs to the model class. When $q_{s,t}(y \mid x; \theta) = \int dz\, q_{r,t}(y \mid z; \theta) q_{s,r}(z \mid x; \theta)$, the parameterized kernel achieves $\mathcal{L}_{\mathrm{KC}} = 0$, giving $\min_\theta \mathcal{L}_{\mathrm{KC}}(\theta) = 0$.

Now suppose $\mathcal{L}_{\mathrm{KC}}(\theta) = 0$. The nonnegative KL integrand in Eq. (11) must vanish almost everywhere under the joint sampling distribution. Continuity and full support imply that it vanishes for every initial state and time triple $s < r < t$.

Since the target and learned transition kernels match, zero KL divergence gives

$$q_{s,t}(y \mid x; \theta) = \int dz\, q_{r,t}(y \mid z; \theta) q_{s,r}(z \mid x; \theta) \tag{42}$$

for every x and $s < r < t$, with equality almost everywhere in y. Thus, the learned kernel satisfies Chapman–Kolmogorov consistency. By assumption, it also satisfies the boundary conditions with limits locally uniform in (s, x). Applying Proposition 1 yields

$$q_{s,t}(\cdot \mid x; \theta) = p_{s,t}(\cdot \mid x), \qquad \forall x,\ s < t. \tag{43}$$

□

B Derivations

B.1 Chapman–Kolmogorov KL loss derivation

Fix a source state x and times $s < r < t$, and hold η fixed when optimizing θ. The two-step target density is

$$q_{s,r,t}(y \mid x; \eta) = \int dz\, q_{r,t}(y \mid z; \eta) q_{s,r}(z \mid x; \eta). \tag{44}$$

We expand the KL divergence as

$$\begin{aligned}
&D_{\mathrm{KL}}(q_{s,r,t}(\cdot \mid x; \eta) \,\|\, q_{s,t}(\cdot \mid x; \theta)) \\
&\quad = \int dy\, q_{s,r,t}(y \mid x; \eta) \log \frac{q_{s,r,t}(y \mid x; \eta)}{q_{s,t}(y \mid x; \theta)} \\
&\quad = -\int dy\, q_{s,r,t}(y \mid x; \eta) \log q_{s,t}(y \mid x; \theta) + C \\
&\quad = -\int dz\, dy\, q_{s,r}(z \mid x; \eta) q_{r,t}(y \mid z; \eta) \log q_{s,t}(y \mid x; \theta) + C \\
&\quad = \mathbb{E}_{\substack{z \sim q_{s,r}(\cdot \mid x; \eta) \\ y \sim q_{r,t}(\cdot \mid z; \eta)}} [-\log q_{s,t}(y \mid x; \theta)] + C,
\end{aligned} \tag{45}$$

where

$$C = \int dy\, q_{s,r,t}(y \mid x; \eta) \log q_{s,r,t}(y \mid x; \eta) \tag{46}$$

is the negative entropy of the fixed target and is independent of θ. The two objectives therefore have the same minimizers,

$$\begin{aligned}
&\arg\min_\theta D_{\mathrm{KL}}(q_{s,r,t}(\cdot \mid x; \eta) \,\|\, q_{s,t}(\cdot \mid x; \theta)) \\
&\quad = \arg\min_\theta \mathbb{E}_{\substack{z \sim q_{s,r}(\cdot \mid x; \eta) \\ y \sim q_{r,t}(\cdot \mid z; \eta)}} [-\log q_{s,t}(y \mid x; \theta)].
\end{aligned} \tag{47}$$

Thus, we sample two-step transitions and evaluate their likelihood under the direct transition model without evaluating the composed density. Taking the expectation over the sampled source states and times gives Eq. (12).

B.2 Learning the boundary conditions

In Section 3.2, we enforce the boundary conditions through the parameterization. Alternatively, we can retain the locality and diffusion conditions in the parameterization and learn the infinitesimal drift through a training objective. We consider two approaches: direct regression to the known vector field and likelihood training on short-time samples generated from the SDE's drift and diffusion. We also describe a conditional regression formulation for generative modeling, where the marginal vector field is unavailable but conditional training targets can be constructed.

Drift regression. Instead of adding the known displacement $\tau v(x)$ as in Eq. (15), we apply the flow directly to the perturbed source state,

$$y_\theta = \Psi(x + \sqrt{\tau}\, gz; x, \theta), \qquad z \sim \mathcal{N}(0, I). \tag{48}$$

We use the coupling layers in Eq. (17) with $n_a = n_b = 1$, allowing the flow to introduce a displacement proportional to τ while preserving the locality and diffusion conditions.

Assume the conditioners are uniformly bounded over their inputs for sufficiently small τ, and jointly continuous in their inputs and τ at $\tau = 0$. The model's infinitesimal drift is then

$$v_\theta(x) := \frac{\partial}{\partial \tau}\mathbb{E}_z[y_\theta - x]\bigg|_{\tau=0} = \frac{\partial \Psi(x; x, \theta)}{\partial \tau}\bigg|_{\tau=0}, \tag{49}$$

where the derivative of Ψ holds its input and source state fixed.

Since every coupling layer is the identity at $\tau = 0$, differentiating their composition gives

$$v_\theta(x) = \sum_{k=1}^{K} \frac{\partial \Psi_k(x; x, \theta)}{\partial \tau}\bigg|_{\tau=0}. \tag{50}$$

For $n_a = n_b = 1$, both gates have derivative ω at $\tau = 0$. A layer that transforms the second block therefore contributes

$$\frac{\partial \Psi_k(x; x, \theta)}{\partial \tau}\bigg|_{\tau=0} = \omega \begin{pmatrix} 0 \\ x_2 \odot a_k(x_1; x, \theta) + b_k(x_1; x, \theta) \end{pmatrix}, \tag{51}$$

where $x = [x_1, x_2]$ and the conditioners are evaluated at $\tau = 0$. For layers transforming the first block, the block roles are exchanged. Thus, we can evaluate the infinitesimal drift directly from the scale and shift conditioners. Expanding the gated transformations gives $\mathbb{E}[y_\theta - x] = \tau v_\theta(x) + o(\tau)$. The flow correction has root mean square size $O(\tau)$, so the same moment and locality estimates used in Section A.2 establish preservation of the diffusion and locality conditions.

We learn this drift by minimizing

$$\mathcal{L}_{\text{drift}}(\theta) = \mathbb{E}_{x\sim\pi}\left[\|v_\theta(x) - v(x)\|^2\right], \tag{52}$$

where π is the distribution of source states used for training.

For generative modeling, we can sample states and vector-field targets from prescribed conditional probability paths, following conditional flow matching (Lipman et al., 2023). When the conditional targets average to the marginal vector field given the state and time, this regression objective differs from regression against the marginal field only by a constant independent of θ.

Short-time likelihood. Alternatively, we can supervise the model using short-time Euler–Maruyama transitions. For a small elapsed time $h > 0$, we sample

$$y = x + h\, v(x) + \sqrt{h}\, gz, \qquad z \sim \mathcal{N}(0, I). \tag{53}$$

We maximize the likelihood of these samples under the model's transition density at elapsed time h,

$$\mathcal{L}_{\text{EM}}(\theta) = \mathbb{E}_{x\sim\pi,\, z\sim\mathcal{N}(0,I)}[-\log q_h(y \mid x; \theta)]. \tag{54}$$

The target has mean displacement $h\, v(x)$ and covariance $h\, gg^\top$, giving the required infinitesimal moments as $h \to 0$. At a fixed h, this objective fits the Euler–Maruyama approximation and does not by itself guarantee the boundary conditions at $\tau = 0$.

B.3 Gaussian reference kernel

We derive the Gaussian reference kernels for the first- and second-order autonomous parameterizations in Eqs. (15) and (19). The source state x denotes the position in the first-order case and the joint

position–momentum state in the second-order case. By time homogeneity, we take the source time to be zero and write $p_\tau(y \mid x)$ and $q_\tau(y \mid x; \theta)$ for transitions over elapsed time τ.

Throughout these derivations, Ψ is set to the identity, so $q_\tau(y \mid x; \theta)$ is the Gaussian reference density with mean $\mu = x + m$ and covariance $\Sigma = LL^\top$. All expectations and covariances are conditional on the source state, held fixed in time derivatives. We assume a smooth vector field with bounded derivatives and at most linear growth, and constant isotropic noise with scalar coefficient g, to justify the expansions and integrations by parts below.

B.3.1 First-order Gaussian reference

Let $p_\tau(y \mid x)$ denote the conditional density of X_τ given $X_0 = x$, evaluated at y. Conditioning on $X_0 = x$, we have the SDE

$$dX_\tau = v(X_\tau)\, d\tau + g\, dW_\tau, \tag{55}$$

and its corresponding Fokker–Planck equation

$$\frac{\partial p_\tau(y \mid x)}{\partial \tau} = -\langle \nabla_y, p_\tau(y \mid x) v(y) \rangle + \frac{g^2}{2} \Delta_y p_\tau(y \mid x), \tag{56}$$

with $p_0(y \mid x) = \delta(y - x)$.

To obtain a Gaussian reference with a closed-form sampler, we approximate the vector field about the source position. Its Taylor expansion is

$$v(y) = v(x) + \frac{\partial v(x)}{\partial x}(y - x) + O(\|y - x\|^2). \tag{57}$$

Retaining only the zeroth-order term gives the reference SDE

$$dy_\tau = v(x)\, d\tau + g\, dW_\tau, \qquad y_0 = x. \tag{58}$$

Its transition density satisfies

$$\frac{\partial q_\tau(y \mid x; \theta)}{\partial \tau} = -\langle \nabla_y, q_\tau(y \mid x; \theta) v(x) \rangle + \frac{g^2}{2} \Delta_y q_\tau(y \mid x; \theta), \tag{59}$$

with $q_0(y \mid x; \theta) = \delta(y - x)$. The reference SDE has constant drift and diffusion, so its transition kernel is Gaussian. We derive its mean and covariance below.

Mean and covariance. Starting with the first moment, we have

$$\mu(x, \tau) = \int dy\, y\, q_\tau(y \mid x; \theta). \tag{60}$$

Differentiating with respect to τ and integrating by parts gives

$$\frac{\partial \mu}{\partial \tau} = \int dy\, y\, \frac{\partial q_\tau(y \mid x; \theta)}{\partial \tau} \tag{61}$$

$$= -\int dy\, y\, \langle \nabla_y, q_\tau(y \mid x; \theta) v(x) \rangle + \frac{g^2}{2} \int dy\, y\, \Delta_y q_\tau(y \mid x; \theta) \tag{62}$$

$$= \int dy\, q_\tau(y \mid x; \theta) v(x) = v(x). \tag{63}$$

Using the initial condition $\mu(x, 0) = x$, we obtain

$$\mu(x, \tau) = x + \tau v(x). \tag{64}$$

Similarly, differentiating the second moment and integrating by parts gives

$$\frac{\partial}{\partial \tau} \mathbb{E}[yy^\top] = \int dy\, yy^\top \frac{\partial q_\tau(y \mid x; \theta)}{\partial \tau} \tag{65}$$

$$= v(x)\mu^\top + \mu v(x)^\top + g^2 I_d. \tag{66}$$

Since $\Sigma = \mathbb{E}[yy^\top] - \mu\mu^\top$, its derivative is

$$\frac{\partial \Sigma}{\partial \tau} = \frac{\partial}{\partial \tau} \mathbb{E}[yy^\top] - \frac{\partial \mu}{\partial \tau} \mu^\top - \mu \left(\frac{\partial \mu}{\partial \tau} \right)^\top \tag{67}$$

$$= g^2 I_d. \tag{68}$$

Using the initial condition $\Sigma(0) = 0$, we obtain

$$\Sigma(\tau) = \tau g^2 I_d. \tag{69}$$

The reference density and sampler are

$$q_\tau(y \mid x; \theta) = \mathcal{N}\big(y \mid \mu(x, \tau), \Sigma(\tau)\big), \tag{70}$$

$$y = x + \tau v(x) + u, \qquad u = \sqrt{\tau}\, g z, \quad z \sim \mathcal{N}(0, I_d). \tag{71}$$

For nonzero g, the reference covariance is positive definite whenever $\tau > 0$.

B.3.2 Second-order Gaussian reference

We start from the second-order Langevin dynamics

$$\begin{aligned} dX_\tau &= P_\tau \, d\tau, \\ dP_\tau &= \big(v(X_\tau) - \gamma P_\tau\big)\, d\tau + g\, dW_\tau, \end{aligned} \tag{72}$$

conditioned on the source state $x = (X, P)$, with $X_0 = X$ and $P_0 = P$. As in the first-order case, we retain the zeroth-order approximation of the vector field about the source position while retaining momentum damping. Writing the reference state as $y(\tau) = (y_x(\tau), y_p(\tau))$, this gives

$$\begin{aligned} dy_x &= y_p \, d\tau, \\ dy_p &= \big(v(X) - \gamma y_p\big)\, d\tau + g\, dW_\tau, \end{aligned} \tag{73}$$

with $y_x(0) = X$ and $y_p(0) = P$. Here, g is scalar, with diffusion matrix gI_d. The reference density $q_\tau(y \mid x; \theta)$ satisfies

$$\frac{\partial q_\tau}{\partial \tau} = -\langle \nabla_{y_x}, y_p q_\tau \rangle - \langle \nabla_{y_p}, \big(v(X) - \gamma y_p\big) q_\tau \rangle + \frac{g^2}{2} \Delta_{y_p} q_\tau, \tag{74}$$

with $q_0(y \mid x; \theta) = \delta(y_x - X)\delta(y_p - P)$. The reference dynamics are linear in the joint state and driven by Gaussian noise, so their transition kernel is Gaussian. Now, we derive its mean and covariance.

Mean. Starting with the first moment, we have

$$\mu(x, \tau) = \begin{pmatrix} \mu_x(x, \tau) \\ \mu_p(x, \tau) \end{pmatrix} = \int dy\, y\, q_\tau(y \mid x; \theta). \tag{75}$$

We suppress the arguments of q_τ in the intermediate calculations. Differentiating with respect to τ, substituting Eq. (74), and integrating by parts gives

$$\begin{aligned} \frac{\partial \mu}{\partial \tau} &= \int dy\, y\, \frac{\partial q_\tau}{\partial \tau} \\ &= -\int dy\, y\, \langle \nabla_{y_x}, y_p q_\tau \rangle - \int dy\, y\, \langle \nabla_{y_p}, \big(v(X) - \gamma y_p\big) q_\tau \rangle \\ &\quad + \frac{g^2}{2} \int dy\, y\, \Delta_{y_p} q_\tau \\ &= \int dy \begin{pmatrix} y_p \\ v(X) - \gamma y_p \end{pmatrix} q_\tau = \begin{pmatrix} \mu_p \\ v(X) - \gamma \mu_p \end{pmatrix}. \end{aligned} \tag{76}$$

Thus, the position and momentum means satisfy

$$\begin{aligned} \frac{d\mu_x}{d\tau} &= \mu_p, \\ \frac{d\mu_p}{d\tau} + \gamma \mu_p &= v(X), \end{aligned} \tag{77}$$

with $\mu_x(0) = X$ and $\mu_p(0) = P$. Multiplying the momentum equation by $e^{\gamma\tau}$ and integrating from 0 to τ gives

$$\begin{aligned} \frac{d}{d\tau}(e^{\gamma\tau} \mu_p(\tau)) &= e^{\gamma\tau} v(X), \\ e^{\gamma\tau} \mu_p(\tau) - P &= v(X) \int_0^\tau dr\, e^{\gamma r}, \\ \mu_p(\tau) &= e^{-\gamma\tau} P + \frac{1 - e^{-\gamma\tau}}{\gamma} v(X). \end{aligned} \tag{78}$$

For position, integrating its mean equation gives

$$\begin{aligned}\mu_x(\tau) - X &= \int_0^\tau dr\, \mu_p(r) \\ &= P\int_0^\tau dr\, e^{-\gamma r} + \frac{v(X)}{\gamma}\int_0^\tau dr\,(1 - e^{-\gamma r}) \\ &= \frac{1-e^{-\gamma\tau}}{\gamma}P + \frac{\gamma\tau - 1 + e^{-\gamma\tau}}{\gamma^2}v(X). \end{aligned} \tag{79}$$

Using the functions

$$\phi_1(\tau) = \frac{1-e^{-\gamma\tau}}{\gamma}, \qquad \phi_2(\tau) = \frac{\tau - \phi_1(\tau)}{\gamma}, \tag{80}$$

we can write the mean as $\mu = x + m$, with

$$m = \begin{pmatrix} \phi_1(\tau)P + \phi_2(\tau)v(X) \\ \phi_1(\tau)\big(v(X) - \gamma P\big) \end{pmatrix}. \tag{81}$$

Covariance. Now, for the covariance, define the fluctuations centered around the mean

$$\widetilde{X} = y_x - \mu_x, \qquad \widetilde{P} = y_p - \mu_p. \tag{82}$$

Both start at zero because the source state is fixed. Subtracting the mean equations from the reference dynamics gives

$$\begin{aligned} d\widetilde{X} &= \widetilde{P}\, d\tau, \\ d\widetilde{P} &= -\gamma\widetilde{P}\, d\tau + g\, dW_\tau, \end{aligned} \tag{83}$$

where we note that the vector field cancels.

Now, for each spatial coordinate i, define the position variance, position-momentum covariance, and momentum variance as

$$S_{xx} = \mathbb{E}[\widetilde{X}_i^2], \qquad S_{xp} = \mathbb{E}[\widetilde{X}_i\,\widetilde{P}_i], \qquad S_{pp} = \mathbb{E}[\widetilde{P}_i^2]. \tag{84}$$

Each quantity is initially zero. We write $W_{\tau,i}$ for the ith component of the Brownian motion.

Starting with S_{pp}, the product rule gives

$$d(\widetilde{P}_i^2) = 2\widetilde{P}_i\, d\widetilde{P}_i + (d\widetilde{P}_i)^2. \tag{85}$$

Using the Itô identities $(dW_{\tau,i})^2 = d\tau$ and $d\tau\, dW_{\tau,i} = (d\tau)^2 = 0$, we obtain

$$d(\widetilde{P}_i^2) = -2\gamma\widetilde{P}_i^2\, d\tau + 2g\widetilde{P}_i\, dW_{\tau,i} + g^2\, d\tau. \tag{86}$$

Taking expectations eliminates the stochastic integral,

$$\frac{dS_{pp}}{d\tau} + 2\gamma S_{pp} = g^2. \tag{87}$$

Multiplying by $e^{2\gamma\tau}$ and using $S_{pp}(0) = 0$, integrating with respect to time gives

$$\begin{aligned} \frac{d}{d\tau}\big(e^{2\gamma\tau}S_{pp}(\tau)\big) &= g^2 e^{2\gamma\tau}, \\ S_{pp}(\tau) &= \frac{g^2}{2\gamma}\big(1 - e^{-2\gamma\tau}\big). \end{aligned} \tag{88}$$

We repeat for S_{xp},

$$d(\widetilde{X}_i\,\widetilde{P}_i) = \widetilde{X}_i\, d\widetilde{P}_i + \widetilde{P}_i\, d\widetilde{X}_i + d\widetilde{X}_i\, d\widetilde{P}_i. \tag{89}$$

Since $d\widetilde{X}_i = \widetilde{P}_i\, d\tau$, its quadratic covariation with $\widetilde{P}_i$ is zero. Substituting the fluctuation equations therefore gives

$$d(\widetilde{X}_i\,\widetilde{P}_i) = \big(-\gamma\widetilde{X}_i\,\widetilde{P}_i + \widetilde{P}_i^2\big)d\tau + g\widetilde{X}_i\, dW_{\tau,i}. \tag{90}$$

Taking expectations,

$$\frac{dS_{xp}}{d\tau} + \gamma S_{xp} = S_{pp}, \tag{91}$$

and substituting the expression for S_{pp}, multiplying by $e^{\gamma\tau}$, and integrating from 0 to τ, using $S_{xp}(0) = 0$, gives

$$\frac{d}{d\tau}(e^{\gamma\tau}S_{xp}(\tau)) = \frac{g^2}{2\gamma}\big(e^{\gamma\tau} - e^{-\gamma\tau}\big),$$
$$e^{\gamma\tau}S_{xp}(\tau) = \frac{g^2}{2\gamma^2}\big(e^{\gamma\tau} + e^{-\gamma\tau} - 2\big),$$
$$S_{xp}(\tau) = \frac{g^2}{2\gamma^2}\big(1 - e^{-\gamma\tau}\big)^2. \tag{92}$$

Position and momentum are correlated because position accumulates the momentum fluctuations.

Finally, for S_{xx},

$$d(\widetilde{X}_i^2) = 2\widetilde{X}_i\, d\widetilde{X}_i + (d\widetilde{X}_i)^2 = 2\widetilde{X}_i\, \widetilde{P}_i\, d\tau. \tag{93}$$

Taking expectations and integrating gives

$$\frac{dS_{xx}}{d\tau} = 2S_{xp},$$
$$S_{xx}(\tau) = \frac{g^2}{\gamma^2}\int_0^\tau dr\, \big(1 - e^{-\gamma r}\big)^2$$
$$= \frac{g^2}{\gamma^2}\left[\tau - \frac{2(1 - e^{-\gamma\tau})}{\gamma} + \frac{1 - e^{-2\gamma\tau}}{2\gamma}\right]. \tag{94}$$

Since the diffusion is isotropic and the fluctuation equations do not couple different spatial coordinates, the joint covariance is

$$\Sigma(\tau) = \begin{pmatrix} S_{xx}I_d & S_{xp}I_d \\ S_{xp}I_d & S_{pp}I_d \end{pmatrix}. \tag{95}$$

Reference kernel and sampler. We can now construct the covariance factor,

$$L(\tau) = \begin{pmatrix} \sigma_{xx}gI_d & 0 \\ \sigma_{px}gI_d & \sigma_{pp}gI_d \end{pmatrix}. \tag{96}$$

Matching the entries of $LL^\top$ with Σ gives

$$\sigma_{xx}(\tau) = \frac{1}{\gamma}\sqrt{\tau - 2\phi_1(\tau) + \frac{1 - e^{-2\gamma\tau}}{2\gamma}},$$
$$\sigma_{px}(\tau) = \frac{\phi_1(\tau)^2}{2\sigma_{xx}(\tau)},$$
$$\sigma_{pp}(\tau) = \sqrt{\frac{1 - e^{-2\gamma\tau}}{2\gamma} - \sigma_{px}(\tau)^2}. \tag{97}$$

For $g > 0$ and $\tau > 0$, this is the Cholesky factor with positive diagonal. At $\tau = 0$, we define $L(0) = 0$ by continuity.

The reference kernel and sampler are therefore

$$q_\tau(y \mid x; \theta) = \mathcal{N}\big(y \mid \mu(x, \tau), \Sigma(\tau)\big), \tag{98}$$
$$y_x = \mu_x + g\sigma_{xx}z_x,$$
$$y_p = \mu_p + g\sigma_{px}z_x + g\sigma_{pp}z_p, \tag{99}$$

where $z_x, z_p \sim \mathcal{N}(0, I_d)$. The shared z_x produces the position–momentum correlation derived above.

B.3.3 Midpoint reference and leading corrections

For the bounded time schedule $\lambda(\tau) = h(\tau)/\tau$, we have $h(\tau) = \tau$ near zero, so the schedule does not affect the expansions below. We retain the smoothness and moment assumptions stated above. Evaluating the vector field at $X_{\mathrm{mid}} = X + \frac{\tau}{2}P$ gives the reference mean

$$\mu_x = X + \phi_1(\tau)P + \phi_2(\tau)v(X_{\mathrm{mid}}),$$
$$\mu_p = e^{-\gamma\tau}P + \phi_1(\tau)v(X_{\mathrm{mid}}). \tag{100}$$

Its covariance remains that of Eq. (95). We denote its blocks by Σ_{xx}, Σ_{xp}, and Σ_{pp}. Throughout this subsection, μ denotes the midpoint reference mean.

Mean corrections. Let $\mu_x^\star = \mathbb{E}[X_\tau]$ and $\mu_p^\star = \mathbb{E}[P_\tau]$ denote the exact conditional means. Taking expectations of the SDE gives

$$\begin{aligned}\frac{\partial \mu_x^\star}{\partial \tau} &= \mu_p^\star,\\ \frac{\partial \mu_p^\star}{\partial \tau} &= \mathbb{E}[v(X_\tau)] - \gamma \mu_p^\star, \qquad \mu_x^\star(0) = X, \quad \mu_p^\star(0) = P.\end{aligned} \tag{101}$$

Since position has no direct diffusion term, differentiating the expected vector field at the source gives

$$\frac{\partial}{\partial \tau}\mathbb{E}[v(X_\tau)]\Big|_{\tau=0} = \frac{\partial v(X)}{\partial X} P, \tag{102}$$

$$\frac{\partial^2}{\partial \tau^2}\mathbb{E}[v(X_\tau)]\Big|_{\tau=0} = \frac{\partial^2 v(X)}{\partial X^2}[P,P] + \frac{\partial v(X)}{\partial X}\big(v(X) - \gamma P\big). \tag{103}$$

Here, $\frac{\partial^2 v(X)}{\partial X^2}[P,P]$ is the second spatial derivative applied to (P,P). The midpoint vector field has the expansion

$$v(X_{\mathrm{mid}}) = v(X) + \frac{\tau}{2}\frac{\partial v(X)}{\partial X} P + \frac{\tau^2}{8}\frac{\partial^2 v(X)}{\partial X^2}[P,P] + O(\tau^3). \tag{104}$$

Taylor expanding the exact mean equations and subtracting Eq. (100) yields

$$\mu_x^\star - \mu_x = -\frac{\tau^3}{12}\frac{\partial v(X)}{\partial X} P + O(\tau^4), \tag{105}$$

$$\begin{aligned}\mu_p^\star - \mu_p = \tau^3 \Bigg[&\frac{1}{24}\frac{\partial^2 v(X)}{\partial X^2}[P,P] + \frac{1}{6}\frac{\partial v(X)}{\partial X} v(X)\\ &- \frac{\gamma}{12}\frac{\partial v(X)}{\partial X} P\Bigg] + O(\tau^4).\end{aligned} \tag{106}$$

Thus, both mean corrections start at order τ^3. Evaluating the vector field at the source instead leaves a momentum mean correction $\frac{\tau^2}{2}\frac{\partial v(X)}{\partial X} P + O(\tau^3)$. The midpoint removes this term while preserving the reference's infinitesimal drift. Since the covariance is unchanged and the reference remains Gaussian, its diffusion and locality conditions are also preserved.

Covariance corrections. The Itô product rule gives the exact conditional covariance equations

$$\frac{\partial}{\partial \tau}\mathrm{Cov}(X_\tau) = \mathrm{Cov}(X_\tau, P_\tau) + \mathrm{Cov}(P_\tau, X_\tau), \tag{107}$$

$$\frac{\partial}{\partial \tau}\mathrm{Cov}(X_\tau, P_\tau) = \mathrm{Cov}(P_\tau) - \gamma \mathrm{Cov}(X_\tau, P_\tau) + \mathrm{Cov}(X_\tau, v(X_\tau)), \tag{108}$$

$$\frac{\partial}{\partial \tau}\mathrm{Cov}(P_\tau) = -2\gamma \mathrm{Cov}(P_\tau) + g^2 I_d + \mathrm{Cov}(v(X_\tau), P_\tau) + \mathrm{Cov}(P_\tau, v(X_\tau)). \tag{109}$$

All covariance blocks vanish at $\tau = 0$. The reference captures the leading terms

$$\begin{aligned}\Sigma_{pp} &= g^2 \tau I_d + O(\tau^2),\\ \Sigma_{xp} &= \frac{g^2\tau^2}{2} I_d + O(\tau^3),\\ \Sigma_{xx} &= \frac{g^2\tau^3}{3} I_d + O(\tau^4).\end{aligned} \tag{110}$$

Expanding the vector field about X gives

$$\mathrm{Cov}(v(X_\tau), P_\tau) = \frac{g^2\tau^2}{2}\frac{\partial v(X)}{\partial X} + O(\tau^3), \tag{111}$$

$$\mathrm{Cov}(X_\tau, v(X_\tau)) = \frac{g^2\tau^3}{3}\left(\frac{\partial v(X)}{\partial X}\right)^\top + O(\tau^4). \tag{112}$$

Subtracting the reference covariance equations and integrating the leading terms therefore yields

$$\mathrm{Cov}(P_\tau) - \Sigma_{pp} = \frac{g^2\tau^3}{6}\left[\frac{\partial v(X)}{\partial X} + \left(\frac{\partial v(X)}{\partial X}\right)^\top\right] + O(\tau^4), \tag{113}$$

$$\mathrm{Cov}(X_\tau, P_\tau) - \Sigma_{xp} = \frac{g^2\tau^4}{24}\left[\frac{\partial v(X)}{\partial X} + 3\left(\frac{\partial v(X)}{\partial X}\right)^\top\right] + O(\tau^5), \tag{114}$$

$$\mathrm{Cov}(X_\tau) - \Sigma_{xx} = \frac{g^2\tau^5}{30}\left[\frac{\partial v(X)}{\partial X} + \left(\frac{\partial v(X)}{\partial X}\right)^\top\right] + O(\tau^6). \tag{115}$$

These corrections are the same for the source and midpoint references because their covariances are identical.

Centered random corrections. Mean and covariance errors do not specify the full distributional correction. To examine the sample corrections, couple the exact dynamics and the reference using the same Brownian motion:

$$\begin{aligned} y_x &= \mu_x + g\int_0^\tau dW_r\,\phi_1(\tau - r), \\ y_p &= \mu_p + g\int_0^\tau dW_r\,e^{-\gamma(\tau-r)}. \end{aligned} \tag{116}$$

The leading centered position fluctuation at time r is $g\int_0^r dW_w\,(r-w)$. Substituting this term into the spatial expansion of the vector field and integrating the SDE gives

$$(X_\tau - \mu_x^\star) - (y_x - \mu_x) = \frac{g}{6}\frac{\partial v(X)}{\partial X}\int_0^\tau dW_r\,(\tau - r)^3 + O_{L^2}(\tau^4), \tag{117}$$

$$(P_\tau - \mu_p^\star) - (y_p - \mu_p) = \frac{g}{2}\frac{\partial v(X)}{\partial X}\int_0^\tau dW_r\,(\tau - r)^2 + O_{L^2}(\tau^{7/2}). \tag{118}$$

Here, $O_{L^2}(\tau^a)$ denotes a remainder with root mean square size $O(\tau^a)$. By Itô isometry, the leading centered corrections have orders $O(\tau^{7/2})$ for position and $O(\tau^{5/2})$ for momentum. In particular, the momentum correction starts earlier than its $O(\tau^3)$ mean correction. These leading random terms are Gaussian, while higher-order terms in the vector field can produce non-Gaussian corrections. The Brownian integrals provide a coupling for comparing correction sizes and need not be determined by y. The learned map is trained to match the conditional transition distribution, including its non-Gaussian corrections.

Gate orders. The reference position and momentum fluctuations have orders $\tau^{3/2}$ and $\tau^{1/2}$, respectively. A log-scale gate with $n_a = 2$ produces multiplicative corrections of orders $\tau^{7/2}$ and $\tau^{5/2}$, matching the centered correction orders in Eq. (118). This motivates using $n_a = 2$ for both blocks.

For the source-evaluated reference, the position and momentum mean corrections begin at orders $O(\tau^3)$ and $O(\tau^2)$, motivating shift gates with $n_b = 3$ and $n_b = 2$, respectively. Midpoint evaluation raises the momentum mean correction to order $O(\tau^3)$. We retain the momentum shift gate with $n_b = 2$ in this case as we found it empirically better.

Boundary preservation. Both second-order references satisfy the boundary conditions and have $\mathbb{E}\|L(\tau)z\|^2 = O(\tau)$. Under the boundedness assumptions of Proposition 2, define the flow correction as

$$\Delta_\tau = \Psi(L(\tau)z; x, \theta) - L(\tau)z. \tag{119}$$

With $n_a = 2$ for both blocks and shift powers of at least two, the perturbation bound in Eq. (20) gives

$$\left(\mathbb{E}\|\Delta_\tau\|^2\right)^{1/2} \le C\left[\tau^2\left(\mathbb{E}\|L(\tau)z\|^2\right)^{1/2} + \tau^2\right] = O(\tau^2) = o(\tau). \tag{120}$$

The correction therefore leaves the infinitesimal drift and diffusion unchanged. For every $\epsilon > 0$, locality follows from

$$\Pr(\|y - x\| > \epsilon) \le \Pr(\|m + L(\tau)z\| > \epsilon/2) + \frac{4\mathbb{E}\|\Delta_\tau\|^2}{\epsilon^2} = o(\tau). \tag{121}$$

C Implementation Details

C.1 Training Details

Standardization. After centering molecular positions at the center of mass, we standardize the state and elapsed time as

$$\widehat{X}_i = \frac{X_i}{X_{\mathrm{std}}}, \qquad \widehat{P}_i = \frac{P_i}{P_{\mathrm{std},i}}, \qquad \widehat{\tau} = \frac{\tau}{T_{\mathrm{std}}}. \tag{122}$$

The position scale is the root mean square over training configurations, atoms and Cartesian coordinates, $X_{\mathrm{std}}^2 = (3N)^{-1}\mathbb{E}_{\mathrm{train}}[\sum_i \|X_i\|^2]$, where N is the number of atoms. We use $P_{\mathrm{std},i} = \sqrt{m_i k_B T}$ at the system temperature and $T_{\mathrm{std}} = 1\,\mathrm{fs}$. These scales remain fixed during training and inference, including momentum augmentation. Generated states are converted back to physical units using the same scales.

Table A1: **Standardization scales.** Position scales are fitted to the training split. Momentum scales are listed by element at 300 K for AL2/AL4 and 310 K for AL6.

Scale	Unit	AL2	AL4	AL6
X_{std}	nm	0.166	0.227	0.300
T_{std}	fs	1	1	1
$P_{\mathrm{std},i}$ (H)	amu nm ps^{-1}	1.585	1.585	1.611
$P_{\mathrm{std},i}$ (C)	amu nm ps^{-1}	5.473	5.473	5.564
$P_{\mathrm{std},i}$ (N)	amu nm ps^{-1}	5.910	5.910	6.008
$P_{\mathrm{std},i}$ (O)	amu nm ps^{-1}	6.317	6.317	6.421

Momentum Augmentation. To expose the model to a wider range of source momenta, we sample $P_i \sim \mathcal{N}(0, m_i k_B T)$ for each atom i, then enforce zero total momentum. For 75% of training examples, T is drawn from a Gaussian with mean $500\,\mathrm{K}$ and standard deviation $150\,\mathrm{K}$, clamped below at $50\,\mathrm{K}$. The remaining examples use the system temperature. This augmentation changes the source momentum distribution while the target Langevin dynamics remain at the system temperature.

Algorithm 1 Training with $\mathcal{L}_{\mathrm{KC}}$

Require: Kernel $q_{s,t}(y \mid x; \theta)$, target parameters η, configurations $\mathcal{D}$, time prior $p(\tau)$

1: **repeat**
2: $\quad X \sim \mathcal{D}, \quad T \sim \mathcal{N}(500\,\mathrm{K}, (150\,\mathrm{K})^2), \quad P \sim \mathcal{N}(0, m\,k_B T), \quad x \leftarrow (X, P)$
3: $\quad \tau \sim p(\tau), \quad r \sim T_{\min}/2 + (\tau - T_{\min})\,\mathrm{Beta}(2,2)$
4: $\quad z \sim q_{0,r}(z \mid x; \eta), \quad y \sim q_{r,\tau}(y \mid z; \eta)$
5: $\quad \mathcal{L}_{\mathrm{KC}} \leftarrow -\log q_{0,\tau}(y \mid x; \theta)$
6: $\quad \theta \leftarrow \textsc{AdamW}(\theta, \nabla_\theta \mathcal{L}_{\mathrm{KC}}), \quad \eta \leftarrow \mathrm{EMA}(\theta)$
7: **until** converged

Equivariance. We center positions at the center of mass and keep positions and momenta in the center-of-mass-free subspace, where densities are evaluated. For peptides, the source configuration defines a rotation $R = [e_1, e_2, e_3]$ from the N, C_α and C atoms of the first complete residue,

$$e_1 = \frac{X_{\mathrm{N}} - X_{\mathrm{C}_\alpha}}{\|X_{\mathrm{N}} - X_{\mathrm{C}_\alpha}\|}, \qquad e_2 \propto (X_{\mathrm{C}} - X_{\mathrm{C}_\alpha}) - \langle X_{\mathrm{C}} - X_{\mathrm{C}_\alpha}, e_1\rangle e_1, \qquad e_3 = e_1 \times e_2. \tag{123}$$

The flow acts on inputs rotated by $R^\top$, and its outputs are rotated back by R. Since R depends only on the source and is orthogonal, the log-determinant is unchanged and the kernel is rotation equivariant. For LJ, we instead apply a random rotation to each training example and at each inference step.

Conditioning features. Besides the source state and τ embedding, each particle i receives features of the Gaussian reference. For first-order dynamics, the reference has mean displacement $m(x,\tau) = h(\tau)\,v(x)$ and width $\sigma(\tau) = g\sqrt{\tau}$,

$$f_{\mathrm{ref},i} = \big(m_i(x,\tau),\, \log \sigma(\tau)\big). \tag{124}$$

For second-order dynamics, with $m(x,\tau)$ from Eq. (81) and the covariance blocks of Eq. (95) for atom i,

$$f_{\mathrm{ref},i} = \big(m_i(x,\tau),\, \log\sigma_{x,i},\, \log\sigma_{p,i},\, \rho_{xp,i}\big), \tag{125}$$

where

$$\sigma_{x,i}^2 = S_{xx,i}, \quad \sigma_{p,i}^2 = S_{pp,i}, \quad \rho_{xp,i} = \frac{S_{xp,i}}{\sigma_{x,i}\,\sigma_{p,i}}. \tag{126}$$

C.2 Inference-time stabilization

We apply the following corrections around each flow proposal to prevent repeated transitions from accumulating errors during long molecular simulations. Algorithm 2 summarizes one step.

Force capping. To limit large force-induced momentum increments, and ultimately prevent non-finite values, we measure each atom's reference impulse in thermal-momentum units as $\lambda(\tau)\phi_1(\tau)\|v_i(X_{\mathrm{mid}})\|/\sqrt{m_i k_B T}$, where $\lambda(\tau) = h(\tau)/\tau$ and $\phi_1(\tau) = (1-e^{-\gamma\tau})/\gamma$. We rescale the midpoint forces according to this dimensionless impulse magnitude and the specified cap. The rescaled forces enter the reference mean and conditioning features.

Rigid and internal momenta. For positions X_i centered at the center of mass and momenta P_i, let $L = \sum_i X_i \times P_i$ be the angular momentum and $I = \sum_i m_i\big(\|X_i\|^2\mathrm{Id} - X_i X_i^\top\big)$ the inertia tensor. We split the momenta into a rigid rotation and an internal part,

$$P_i = m_i\big(I^{-1}L \times X_i\big) + P_i^{\mathrm{int}}. \tag{127}$$

The internal momenta carry no angular momentum, $\sum_i X_i \times P_i^{\mathrm{int}} = 0$, and are therefore orthogonal to every rigid rotation in the mass-weighted inner product $\sum_i P_i \cdot Q_i/m_i$. In units of k_BT, the kinetic energy splits as

$$K^{\mathrm{tot}} = K_{\mathrm{rot}}(L,I) + K^{\mathrm{int}}, \qquad K_{\mathrm{rot}}(L,I) = \frac{L^\top I^{-1} L}{2k_BT}, \qquad K^{\mathrm{int}} = \sum_i \frac{\|P_i^{\mathrm{int}}\|^2}{2m_i k_B T}. \tag{128}$$

For the source (X_t, P_t) and the proposal $(X_{t+\tau}, P_{t+\tau})$, we write U_t, $U_{t+\tau}$ for the potential energies in units of k_BT, K_t, $K_{t+\tau}$ for the internal kinetic energies, L_t for the source angular momentum, and I_t, $I_{t+\tau}$ for the inertia tensors.

Momentum correction. Following Ripken et al. (2026), we keep the proposed positions and minimally adjust the proposed momenta to satisfy an angular-momentum and an energy constraint,

$$\min_{P'} \sum_i \frac{\|P_i' - P_{t+\tau,i}\|^2}{2m_i} \quad \text{s.t.} \quad \sum_i X_{t+\tau,i} \times P_i' = L_{\mathrm{target}}, \qquad K^{\mathrm{int}}(P') = K_{\mathrm{target}}. \tag{129}$$

Because the rigid and internal parts are orthogonal, the two constraints decouple (the cross term in the closed-form solution of Ripken et al. (2026) vanishes): the angular-momentum constraint sets the rigid part and the energy constraint rescales the internal part,

$$P_i' = m_i\big(I_{t+\tau}^{-1} L_{\mathrm{target}} \times X_{t+\tau,i}\big) + \sqrt{\frac{K_{\mathrm{target}}}{K_{t+\tau}}}\, P_{t+\tau,i}^{\mathrm{int}}. \tag{130}$$

The positions and the total momentum are unchanged. For numerical safety, both kinetic energies in the ratio are floored at 0.1. Ripken et al. (2026) conserve the angular momentum and the total energy and apply a thermostat separately; we instead choose targets that follow the Langevin dynamics.

Angular-momentum target. The flow leaves the angular momentum about the source positions, $\sum_i X_{t,i} \times P_{t+\tau,i}$, on the law of the Gaussian reference. When the molecule turns during the step, this value no longer describes the momenta about the proposed positions, so we set

$$L_{\mathrm{target}} = \sum_i X_{t,i} \times P_{t+\tau,i}. \tag{131}$$

This moves the flow's angular momentum onto the proposed positions: the rigid part changes by $m_i\big[(I_{t+\tau}^{-1}\Delta L) \times X_{t+\tau,i}\big]$ with $\Delta L = \sum_i (X_{t,i} - X_{t+\tau,i}) \times P_{t+\tau,i}$, and the internal momenta are untouched.

Energy target. We split each step into an Ornstein–Uhlenbeck update of the momenta at fixed positions, which carries the friction and noise, and Hamiltonian motion, which conserves $U + K^{\mathrm{tot}}$.

The Ornstein–Uhlenbeck update leaves the internal kinetic energy with the closed-form law of the stochastic velocity-rescaling thermostat (Bussi et al., 2007),

$$K_{\mathrm{bath}} = \frac{1}{2}\big(a\sqrt{2K_t} + \sqrt{1-a^2}\,Z\big)^2 + \frac{1}{2}(1-a^2)\,Y, \qquad a = e^{-\gamma\tau}, \tag{132}$$

where γ is the friction coefficient, $n = 3N - 6$ is the number of internal degrees of freedom, and $Z \sim \mathcal{N}(0,1)$ and $Y \sim \chi^2_{n-1}$ are independent. Energy conservation of the Hamiltonian part then gives

$$K_{\mathrm{target}} = U_t - U_{t+\tau} + K_{\mathrm{bath}} + \Delta K_{\mathrm{rot}}, \qquad \Delta K_{\mathrm{rot}} = K_{\mathrm{rot}}(L_t, I_t) - K_{\mathrm{rot}}(L_t, I_{t+\tau}), \tag{133}$$

where ΔK_{rot} is the kinetic energy the rigid rotation releases into the internal motion when the inertia tensor changes at fixed angular momentum. With $\gamma = 0$, where $K_{\mathrm{bath}} = K_t$, and conserved angular momentum, this is conservation of the total energy, the constraint of Ripken et al. (2026). If $K_{\mathrm{target}} < 0.1$, no rescaling of the momenta can absorb the increase in potential energy, and we use $\max(K_{\mathrm{bath}} + \Delta K_{\mathrm{rot}}, 0.1)$ instead.

Adaptive proposal retries. We redraw proposals when $|(U_{t+\tau} + K_{t+\tau}) - (U_t + K_t)| > H_{\mathrm{threshold}}$ or $K_{\mathrm{target}} < 0.1$. After N_{retry} unsuccessful attempts at a given step size, we halve τ and retry until a proposal passes these checks or the attempt budget is exhausted. After each update, we restore the original step size.

Algorithm 2 One inference step

Require: Kernel parameters θ, current state $x_t = (X_t, P_t)$, step size τ, $z \sim \mathcal{N}(0, I)$

$v(X_{\mathrm{mid}}) \leftarrow \textsc{ForceCapFilter}\big(v(X_{\mathrm{mid}})\big)$

$X_{t+\tau}, P_{t+\tau} \leftarrow x_t + m(x_t, \tau) + \Psi\big(L(\tau)z; x_t, \theta\big)$

$X_{t+\tau}, P_{t+\tau} \leftarrow \textsc{RedrawFilter}(x_t, X_{t+\tau}, P_{t+\tau}, \tau)$

$P_{t+\tau} \leftarrow \textsc{MomentumCorrectionFilter}(x_t, X_{t+\tau}, P_{t+\tau}, \tau)$

D Experimental Setup

D.1 Lennard-Jones systems

Potential. We use the confined Lennard-Jones systems studied by Köhler et al. (2020); Akhound-Sadegh et al. (2025). For N particles, let $d_{ij} = \|x_i - x_j\|$ and $\bar{x} = N^{-1}\sum_i x_i$. The potential is

$$E(x) = \epsilon \sum_{i<j} \left[\left(\frac{r_m}{d_{ij}}\right)^{12} - 2\left(\frac{r_m}{d_{ij}}\right)^{6} \right] + \frac{c}{2}\sum_i \|x_i - \bar{x}\|^2, \tag{134}$$

with $\epsilon = 2$, $r_m = 1$ and $c = 1$. We learn overdamped Langevin transitions with drift $D\nabla \log \pi(x)$ and diffusion $\sqrt{2D}$, where $\pi(x) \propto \exp(-E(x)/k_BT)$, $k_BT = 1$ and $D = 0.02$. Positions and noise are restricted to the zero-center-of-mass subspace, with $3(N-1)$ degrees of freedom.

Data. At source temperature $T_S = 1$, we use the public splits of Akhound-Sadegh et al. (2024), with 100k/10k/10k training/validation/test configurations for LJ-13 and 10k/10k/10k for LJ-55. At $T_S = 4$, we generate 200k configurations per system using the NUTS (Hoffman & Gelman, 2014) implementation as done in Akhound-Sadegh et al. (2025), with energy $E/4$. In both settings, the target dynamics and evaluation data remain at $T = 1$. Training draws source configurations independently and uniformly with replacement, with a random rotation for each draw.

Model. The Gaussian reference has mean $x + D\,h(\tau)\nabla \log \pi(x)$ and covariance $2D\tau I$ on the center-of-mass-free subspace, using the duration cap in Section 3.3 with $\tau_0 = \delta = 0.01$. The flow has 9 coupling blocks, each updating one Cartesian axis across all particles given the other two. Each coupling network uses one transformer layer with AdaLN-Zero conditioning and a SwiGLU feed-forward layer. The context includes the source configuration, time embedding of τ, the reference width and source-force magnitudes.

Table A2: **Peptide data and physical settings.** Training uses random configurations from these datasets. The learned dynamics use the listed force fields, temperatures and friction coefficients. The source trajectories use a 1 fs integration step with flexible bonds.

	AL2	AL4	AL6
Sequence	Ace-A-Nme	Ace-AAA-Nme	AAAAAA
Atoms (dimension d)	22 (66)	42 (126)	63 (189)
Force field	Amber ff99SB-ILDN	Amber ff99SB-ILDN	Amber ff14SB
Implicit solvent	OBC2	OBC2	OBC1
Nonbonded cutoff	none	none	2 nm
Temperature	300 K	300 K	310 K
Friction γ	1 ps^{-1}	0.3 ps^{-1}	0.3 ps^{-1}
Trajectory (frame stride)	1 μs (1 ps)	7 μs (10 ps)	3 μs (10 ps)
Random split: train / val / test	800k / 100k / 100k	560k / 70k / 70k	240k / 30k / 30k

D.2 Alanine peptides

Data. AL2 uses the 1 μs trajectory of Köhler et al. (2021). AL4 and AL6 use the datasets of Tan et al. (2025). Table A2 lists the force fields (Lindorff-Larsen et al., 2010; Maier et al., 2015) and OBC solvent models (Onufriev et al., 2004). We randomly split the available configurations 80/10/10 with seed 42, following the split procedure of Ripken et al. (2026) for AL2. For AL4 and AL6, we pool the original SBG splits before reshuffling. During training, we draw positions independently and uniformly with replacement from the training split, and do not use trajectory order or paired transitions. Source momenta are sampled independently using the augmentation in Section 3.3. Energies and forces are implemented in JAX (Bradbury et al., 2018) and checked against OpenMM (Eastman et al., 2017).

D.3 Training

Objective. Our main models use the Chapman–Kolmogorov objective in Eq. (12) with two transitions from the EMA target kernel (Algorithm 1). We split each sampled elapsed time τ into two durations using a rescaled $\mathrm{Beta}(2,2)$ distribution, with each step lasting at least $T_{\min}/2$. The minimum elapsed time $T_{\min}$ is listed in Table A3. For numerical stability, we remove the parameter-independent Gaussian normalization terms from the NLL, divide by the number of unconstrained degrees of freedom and cap each sample's NLL at 10.

Optimization. We use AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 10^{-8} and a global gradient-norm limit of 1. The learning rate warms up linearly to 3×10^{-4} and then decays to 3×10^{-6} with a cosine schedule. Elapsed times follow the mixture prior in Section 3.3, with $T_{\max}$ increased log-linearly during early training. Table A3 lists the settings for each system. The LJ PF-ODE comparison uses the same architecture, with $T_{\max} = 0.05$ annealed from 0.008. Its training budget matches the FLOPs of KC-KL at $T_S = 1$, giving 275k updates for LJ-13 and 545k for LJ-55. We train on a heterogenous cluster consisting of L40s and H100s.

Training compute. We set the HFM training steps to match the estimated FLOPs of our method on each system (Table A4). We estimate FLOPs using JAX 0.9.2 and analyze the loss and its parameter gradients, multiplied by the number of training updates. This includes model evaluations, teacher evaluations, Jacobian–vector products and force computations within the loss. Optimizer and EMA updates, data preparation, validation, communication and compilation are excluded.

D.4 Evaluation

Lennard-Jones. We run 5000 parallel Metropolis–Hastings chains (Hastings, 1970) for 5000 steps, using learned proposals at $\tau = 0.05$ for LJ-13 and $\tau = 0.02$ for LJ-55. Each step uses a fresh random rotation shared by the forward and reverse density evaluations. All chains start from the same energy-minimized configuration, and we evaluate their final states. MALA uses the same initialization and number of chains. We sweep physical step sizes $h \in \{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times$

Table A3: **Training hyperparameters.** LJ times are in reduced units. For LJ, $T_{\max}$ is given for $T_S = 1$ / $T_S = 4$.

	LJ-13	LJ-55	AL2	AL4	AL6
Blocks × attention layers	9×1	9×1	8×2	8×2	8×2
Channels / heads	256 / 4	256 / 4	384 / 6	384 / 6	384 / 6
Parameters	9.4M	9.4M	45.2M	45.2M	45.2M
Gate orders (n_a, n_b)	$(1, 2)$	$(1, 2)$	x: $(2, 3)$, p: $(2, 2)$		
Initial ω (range)	40 ($[4, 400]$)	40 ($[4, 400]$)	400 ps^{-1} ($[10, 2000]$ ps^{-1})		
Reference cap $\tau_0 = \delta$	0.01	0.01	1 fs		
Minimum elapsed time $T_{\min}$	2×10^{-5}	2×10^{-5}	0.02 fs		
$T_{\max}$	0.5 / 0.1	0.5 / 0.1	100 fs	85 fs	85 fs
$T_{\max}$ ramp (start, duration)	0.05, 10%	0.05, 10%	1 fs, 15%		
Log-uniform weight w	0.4	0.4	0.4	0.25	0.25
Batch size	256	256	512	512	512
Training steps	300k	600k	500K	1M	1.5M
Learning-rate warm-up	3k steps	3k steps	2.5K	2.5K	2.5K
Two-step EMA decay	0.995	0.995	0.99975	0.995	0.995
Evaluation EMA decay	0.995	0.995	0.9995	0.995	0.995

Table A4: **Training compute on AL4 and AL6.** Parameter counts, completed training updates and estimated FLOPs for the loss and parameter gradients. FLOP totals follow the accounting described above.

System	Method	Parameters (M)	Batch size	Updates	FLOPs (10^{18})
AL4	HFM	9.09	512	1 288 647	9.699
	KC-maps	45.23	512	1 000 000	9.698
AL6	HFM	9.09	512	1 374 048	21.966
	KC-maps	45.23	512	1 500 000	21.964

$10^{-3}, 10^{-2}, 0.02, 0.05\}$, with 120 s of sampling on one L40S per setting. We select h using validation Geo-$\mathcal{W}_2$, giving 3×10^{-4} for LJ-13 and 10^{-4} for LJ-55.

Energy $\mathcal{W}_2$ compares the 5000 endpoint energies with the 10k test energies. Geo-$\mathcal{W}_2$ compares the endpoints with 5000 test configurations as done in (Klein et al., 2023b). Published baseline values are taken from Potaptchik et al. (2025), without recomputing their geometric metric.

Molecular reference trajectories. Training uses independent draws from the configuration data, whereas evaluation compares generated trajectories with MD reference trajectories. The original AL2 data were generated with the legacy `LangevinIntegrator` in an older OpenMM release. To reduce discretization bias in evaluation, we generate new AL2 and AL4 references with OpenMM 8.5.1 using `LangevinMiddleIntegrator`, a leapfrog form of BAOAB with improved configurational accuracy (Leimkuhler & Matthews, 2013). We retain the physical settings in Table A2 and a 1 fs integration step. The AL2 and AL4 references span 1 μs and 1.5 μs, respectively, with frames saved every 1 ps and the first 1 ns discarded.

The AL4 trajectory does not visit the basin with Ala3 and Ala4 both in α_L. We supplement it with ≈ 3000 frames from simulations initialized in that basin, giving it the 0.21% occupancy observed in the original SBG test segment. For AL6, we retain the 1.8 μs SBG test segment (Tan et al., 2025) as the equilibrium reference.

Molecular sampling. Each setting receives one hour on one L40S with 4 CPU cores, excluding compilation and initialization. All methods start from the same configuration for each molecule, with independently sampled momenta at the target temperature.

- **MD:** OpenMM LangevinMiddle integrator with a 1 fs step, running 1, 4 or 16 independent processes that share the GPU through CUDA MPS.

- **HFM:** the sampler of Ripken et al. (2026), including its energy and momentum corrections. We use $\tau \in \{5, 7\}$ fs with $\{1, 16, 64\}$ chains for AL2 and AL4, and $\tau \in \{3, 5\}$ fs with $\{1, 4, 16\}$ chains for AL6. AL2 uses the released checkpoint. We train AL4 and AL6 with the public code and match their training FLOPs to ours.

- **KC-maps:** $\tau \in \{30, 50, 70, 90\}$ fs with $\{16, 32, 128\}$ chains for AL2, $\tau \in \{30, 50, 70, 85\}$ fs with $\{16, 32, 64\}$ chains for AL4, and $\tau \in 40, 50, 60$ fs with $\{16, 32, 64\}$ chains for AL6. We apply the corrections in Section C.2, with a dimensionless force cap of 7, $N_{\text{retry}} = 5$ attempts per step size and $H_{\text{threshold}} = 40$ for energies in units of $k_B T$. Failed proposals trigger step-size halving down to 1 fs.

Metrics. For the molecular equilibrium metrics, we discard the first 10% of each chain's simulated time and any adaptation frames. We pool the remaining frames and draw 10^4 without replacement, weighted by the simulated interval each frame represents. We compare these with 10^4 uniformly drawn reference frames using exact optimal transport, and report the mean and standard deviation over three subsampling draws. Energy $\mathcal{W}_2$ uses potential energies in units of $k_B T$. Torus $\mathcal{W}_2$ uses the squared wrapped distance over all backbone (ϕ, ψ) angles, with 2, 6 and 10 angles for AL2, AL4 and AL6, respectively.

TICA-$\mathcal{W}_2$ uses squared distances in the first two TICA components (Pérez-Hernández et al., 2013). We fit one fixed projection per molecule using deeptime (Hoffmann et al., 2022) on the original SBG test trajectories, with a lag of 100 frames, Koopman reweighting and kinetic-map scaling. Features comprise pairwise distances among carbon and nitrogen atoms and the sine and cosine of backbone angles. In Fig. 3, modes are fixed regions of the TICA plane, with discovery requiring a continuous 20 ps stay.

E Additional Experiments

E.1 Target EMA decay

In Fig. A1, the stop-gradient target (`ema_decay` $= 0$) has the highest NLL at every elapsed time, decays between 0.99 and 0.9995 perform similarly, and the slowest decay, 0.99975, degarding in performance.

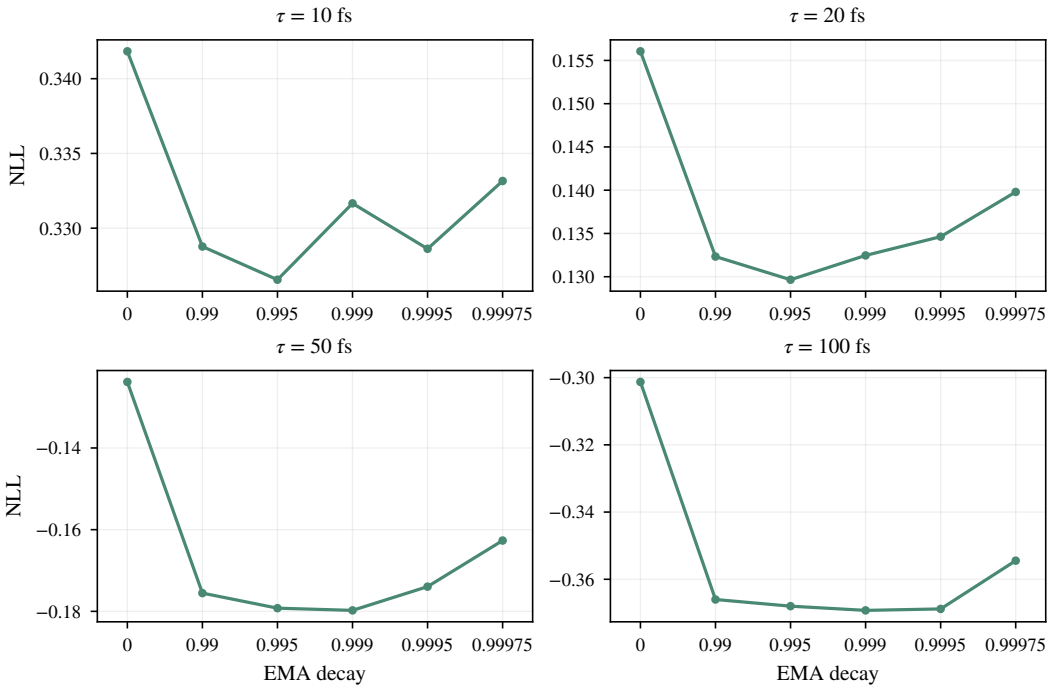


Figure A1: **Target EMA decay on AL2.** Models differ only in `ema_decay` for the two-step target $\eta = \mathrm{EMA}(\theta)$ in Eq. (12). All models are trained for 500K steps with batch size 512. We report NLL on MD transition samples at $\tau \in \{10, 20, 50, 100\}$ fs. A decay of zero corresponds to $\eta = \mathrm{stopgrad}(\theta)$.

E.2 Flow parameterizations

We compare three coordinate choices for the affine coupling flow, using the same Gaussian reference $\mu + Lz$, where $z \sim \mathcal{N}(0, I)$, $\mu = x + m$, and $\Sigma = LL^\top$. Writing Ψ for the coupling map in the chosen coordinates, the three parameterizations are

$$\begin{aligned} \text{Standard:} \quad & y = \Psi(x + m + Lz; x, \theta), \\ \text{Displacement:} \quad & y = x + m + \Psi(Lz; x, \theta), \\ \text{Whitened:} \quad & y = x + m + L\Psi(z; x, \theta). \end{aligned} \tag{135}$$

The standard flow transforms the full joint state, the displacement flow transforms residuals about the reference mean, and the whitened flow transforms standard Gaussian coordinates. All three recover the Gaussian reference when Ψ is the identity. The whitened parameterization uses the full joint covariance, including position–momentum correlations. In the displacement flow, residual input normalization only rescales the conditioner inputs. The affine transformations still act on the original residuals.

We adapt the gates to each coordinate choice. Using $\gamma_n(\tau) = (\omega\tau)^n/[1 + (\omega\tau)^n]$, the gates are

	A_x	A_p	B_x	B_p
Standard	γ_3	γ_2	γ_3	γ_2
Displacement	γ_2	γ_2	γ_3	γ_2
Whitened	γ_2	γ_2	$\sqrt{\gamma_3}$	$\sqrt{\gamma_3}$,

(136)

where the A gates the log scale and B gates the shift in the corresponding coordinates. For bounded coupling outputs, these choices give physical corrections of order $O_p(\tau^3)$ in position and $O_p(\tau^2)$ in momentum, resulting in the same short-time correction orders across the three representations.

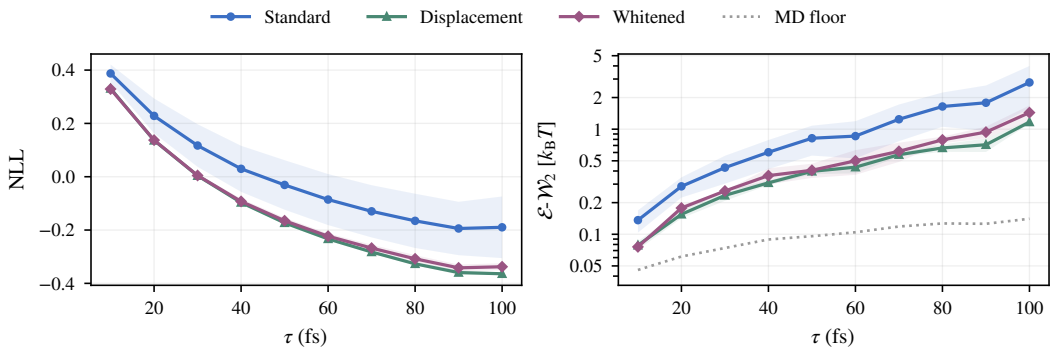


Figure A2: **Flow parameterization on AL2.** Comparison of standard, displacement, and whitened flows, each trained for 500K steps with batch size 512 and evaluated using EMA parameters with decay 0.999. Lines and shaded bands show the mean and standard deviation across training seeds: 43 and 44 for the standard flow, and 42, 43, and 44 for the displacement and whitened flows. Left: NLL per degree of freedom (DoF) on molecular dynamics samples. Right: $\mathcal{E}$-$\mathcal{W}_2$ in units of $k_B T$, averaged over source configurations.

In Fig. A2, displacement and whitened flows perform similarly at short elapsed times, while the displacement flow has lower mean NLL and $\mathcal{E}$-$\mathcal{W}_2$ at longer elapsed times. Both improve on the standard flow across the evaluated elapsed times.