LANGEVIN FLOW MAPS: EFFICIENT MOLECULAR DYNAMICS AND TRANSITION PATH SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular dynamics simulations proceed by integrating the Langevin equations over many small femtosecond timesteps. This poses a challenge for estimating ensemble properties and transition dynamics that occur on much longer timescales. We introduce Langevin Flow Maps, which extend machine-learned force-fields to additionally learn the stochastic Langevin integrator. We show that Langevin Flow Maps enable large-timestep molecular dynamics and recover accurate dynamical properties of the system, while running an order of magnitude faster than current machine-learned force fields. Further, by training on a diverse molecular dataset, we demonstrate a path towards *transferable* Langevin Flow Maps.

1 INTRODUCTION

Molecular dynamics (MD) aims to compute the kinetic and equilibrium properties of a molecular system by numerically integrating equations of motion (Hollingsworth & Dror, 2018). These equations of motion are defined by a potential energy landscape, which in turn defines the forces acting on each atom in the system (Goldstein et al., 2011).

In order to obtain accurate simulations we require that the potential energy landscape accurately describes the system's physics. Quantum Mechanical (QM) force calculations represent the most accurate computational method available for describing this landscape (Tuckerman & Martyna, 2000). However, QM methods are significantly more expensive than classical approximations (Frenkel & Smit, 1996). This computational bottleneck has led to the development of machine-learned force fields (MLFFs) that attempt to learn the QM energy landscape with neural networks (Behler & Parrinello, 2007; Noé et al., 2020; Keith et al., 2021; Unke et al., 2021b). This field has shown great success at improving the efficiency of QM-accurate force calculations.

Nevertheless, force calculation represents one half of the puzzle; we must also integrate these forces over time (Fu et al., 2023). This is commonly achieved by numerically integrating Langevin Dynamics (Leimkuhler & Matthews, 2013). To resolve the rapid oscillatory dynamics, timesteps of the order of 1 femtosecond (fs) are required. However, the transition and equilibrium dynamics of interest are typically observed beyond nanosecond timescales. Integration therefore remains a significant bottleneck in running molecular dynamics calculations (Wang et al., 2025).

As such, we may define the core problem for efficient molecular dynamics as to learn not only the instantaneous forces from data but to learn the integral of these forces too. Recent work has proposed solutions to this problem by learning transfer operators from a dataset of trajectories (Klein et al., 2023a; Schreiner et al., 2023; Diez et al., 2026; Thiemann et al., 2026; Bigi et al., 2025). However, these methods require long molecular dynamics trajectories to train and their predictions are limited to the resolution of the saved trajectory.

Alternatively, flow map methods have emerged as a trajectory-free construction for learning the integral of a differential equation directly (Geng et al., 2025; Boffi et al., 2025a). Hamiltonian Flow Maps (HFMs) apply this machinery to learn the integral of Hamilton's equations of motion (Ripken et al., 2026). Hamiltonian updates can be iterated alongside a stochastic update step such as the Langevin thermostat to sample equilibrium properties in molecular dynamics. Unfortunately, the Hamiltonian and stochastic updates do not commute and as a result this splitting scheme introduces a timestep-dependent bias in the transition dynamics; see Section 2.

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.

In this work, we introduce Langevin Flow Maps (LFMs) that learn the stochastic solution map of Langevin Dynamics directly. LFMs enable large-timestep molecular dynamics to predict equilibrium properties *and* transition dynamics of the system, without access to explicit trajectories as training data.

Our contributions can be summarized as follows:

1. We construct a trajectory-free learning objective for the pathwise solution map of Langevin Dynamics.
2. We demonstrate that LFMs allow for large-timestep molecular dynamics, predicting accurate equilibrium and dynamical properties.
3. We train transferable LFMs on a diverse molecular dataset and show that the model is able to generalize to entirely unseen systems during inference.

2 Background & Related Work

Langevin Dynamics describes the dynamics of a particle subject to a potential energy landscape at constant temperature (Pastor et al., 1988). The system is described by the following coupled stochastic differential equation (SDE). For position x_t, velocity v_t and potential $U(x_t)$, the system evolves via

$$\begin{aligned} dx_t &= v_t dt, \\ dv_t &= \left[-\gamma v_t - m^{-1}\nabla U(x_t)\right] dt + \sigma dW_t, \end{aligned} \tag{1}$$

where m is the particle mass and γ is the friction coefficient and $\sigma = \sqrt{2\gamma k_B T/m}$.

Langevin Dynamics admits a stationary distribution defined by the Hamiltonian $H(x,v) = U(x) + \frac{1}{2}m\|v\|^2$ with density $p \propto \exp(-\beta H(x,v))$, where $\beta = 1/k_B T$. In molecular dynamics this density corresponds to the canonical ensemble (NVT), denoted the Boltzmann distribution.

Numerical schemes. To compute the solution to Equation (1), numerical integration is used. The most widely used method in molecular dynamics is BAOAB (Leimkuhler & Matthews, 2013) that splits Langevin Dynamics into three pieces,

$$\underbrace{dx_t = v_t dt}_{A}, \qquad \underbrace{dv_t = -m^{-1}\nabla U(x_t)dt}_{B}, \qquad \underbrace{dv_t = -\gamma v_t dt + \sigma dW_t}_{O}. \tag{2}$$

The numerical approximation to each piece is then computed in the order in which the name suggests. Specifically, let $\mathcal{L}$ denote the generator of Equation (1), then the solver approximates this transition via

$$e^{\Delta t\mathcal{L}} = e^{\frac{\Delta t}{2}\mathcal{L}_B} e^{\frac{\Delta t}{2}\mathcal{L}_A} e^{\Delta t\mathcal{L}_O} e^{\frac{\Delta t}{2}\mathcal{L}_A} e^{\frac{\Delta t}{2}\mathcal{L}_B} + O(\Delta t^3). \tag{3}$$

where $\mathcal{L}_N$ is the generator for step N. This splitting scheme incurs a local weak (distributional) error $O(\Delta t^3)$ and therefore a global weak error $O(\Delta t^2)$. Many small steps must therefore be taken to ensure accurate dynamics.

Hamiltonian Flow Maps. Recent work by Ripken et al. (2026) demonstrated how to learn the solution (or, flow) map of Hamilton's equations of motion. This corresponds to learning the solution map of the AB split from Equation (2). At inference, Hamiltonian Flow Maps iterate the learned AB map and the O step,

$$e^{\Delta t\mathcal{L}} = e^{\frac{\Delta t}{2}\mathcal{L}_O} e^{\Delta t\mathcal{L}_{AB}} e^{\frac{\Delta t}{2}\mathcal{L}_O} + O(\Delta t^3). \tag{4}$$

This method suffers from the same fundamental non-commutativity of the chosen split as BAOAB and therefore the transition dynamics remain biased even for a perfectly learned Hamiltonian flow $e^{\Delta t\mathcal{L}_{AB}}$.

Transfer Operators An alternative approach to simulating Langevin Dynamics has focused on learning the distributional transfer operator (TO), given by $p_{\Delta t}(x_{t+\Delta t} \mid x_t)$. These methods typically parameterize a generative model that approximates this conditional distribution (Klein et al., 2023a; Schreiner et al., 2023; Diez et al., 2026; Thiemann et al., 2026; Bigi et al., 2025).

The limitation of transfer operators is that they consume molecular dynamics trajectories as training data, requiring expensive classical integration to generate. Further, the learned timescales of the transfer operator are limited to the saved resolutions of the training data.

Table 1: Comparison of large-timestep molecular dynamics methods.

Method	Equilibrium properties	Trajectory free	Transition dynamics
TOs	✓	✗	✓
HFMs	✓	✓	✗
LFMs	✓	✓	✓

3 METHOD

The aim of this work is to learn the solution map of Langevin Dynamics directly, with transition operator $e^{\Delta t \mathcal{L}}$, that requires no explicit molecular dynamics trajectories as training data. To do this, we extend the method presented by McCallum et al. (2026) that demonstrates how to learn the pathwise solution map of an additive-noise SDE. The core idea is to note that the solution map of an SDE is a deterministic function given a realization of the Brownian path $W_{[s,t]} = \{W_u\}_{s \leq u \leq t}$. This enables an efficient regression objective without explicit trajectory supervision.

3.1 LANGEVIN FLOW MAPS

Solution Map. In the case of Langevin Dynamics, we can write the pathwise solution map as

$$z_t = \Psi_{s,t}(z_s, W_{[s,t]}), \quad z_s = \begin{pmatrix} x_s \\ v_s \end{pmatrix}. \tag{5}$$

That is, given an initial state (x_s, v_s) and a Brownian path $W_{[s,t]}$, the final state (x_t, v_t) is entirely determined.

This solution map satisfies a number of properties that can be exploited to parameterize the map for any molecular system. Notably, these properties can be satisfied given a dataset of molecular configurations and forces (x^i, f^i) without access to molecular dynamics trajectories. It is therefore possible to recover the kinetic properties of a system without observing the dynamics directly.

Parameterization. The solution map $\Psi_{s,t}$ can be expanded and written in terms of an integral equation,

$$\begin{aligned} x_t &= x_s + \int_s^t v_u du, \\ v_t &= v_s - \int_s^t \left(\gamma v_u - \frac{1}{m} f(x_u) \right) du + \sigma(W_t - W_s), \end{aligned} \tag{6}$$

where $f(x_t) = -\nabla U(x_t)$ and $\sigma = \sqrt{2\gamma k_B T / m}$. We therefore see that computing the solution map is equivalent to computing the velocity integral and the force integral. Additionally, since the stochastic process (x_t, v_t) necessarily depends on the Brownian path, and we integrate directly over this process, the integrals themselves must depend on the Brownian path. We can therefore write the solution map as,

$$\begin{aligned} x_t &= x_s + (t-s) v_{s,t}(z_s, W_{[s,t]}) \\ v_t &= v_s - (t-s) \left[\gamma v_{s,t}(z_s, W_{[s,t]}) - \frac{1}{m} f_{s,t}(z_s, W_{[s,t]}) \right] + \sigma(W_t - W_s), \end{aligned} \tag{7}$$

where the $(t-s)$ pre-factor is a modelling choice that forces $v_{s,t}$ and $f_{s,t}$ to represent the time-averaged integrals,

$$v_{s,t}(z_s, W_{[s,t]}) = \frac{1}{t-s} \int_s^t v_u du, \quad f_{s,t}(z_s, W_{[s,t]}) = \frac{1}{t-s} \int_s^t f(x_u) du. \tag{8}$$

This guarantees that as $s \to t$, the integrals $v_{s,t}$ and $f_{s,t}$ recover the instantaneous velocity v_t and force $f(x_t)$, respectively.

Consistency conditions. It is possible to show that the solution map to an SDE satisfies a set of consistency conditions (McCallum et al., 2026). Specifically, the solution map $\Psi_{s,t}$ is uniquely determined if (1) it satisfies the correct instantaneous $s = t$ quantities and (2) it satisfies the semigroup property $\Psi_{s,t} = \Psi_{u,t} \circ \Psi_{s,u}$, where $u = (s + t)/2$.

We derive these conditions for Langevin Dynamics and demonstrate sufficiency in Proposition 3.1. These consistency conditions will ultimately allow us to write an objective function which can be minimized to obtain the solution map (see Section 3.3).

Proposition 3.1 (Consistency condition)**.** Let $\Psi_{s,t}$ denote a candidate map to Equation (6) parameterized by Equation (7). Then, for all $s, t \in \mathbb{R}$ and $u = (s + t)/2$, $\Psi_{s,t}$ is the solution map iff,

$$v_{t,t}(z_t, W_{[t,t]}) = v_t, \quad f_{t,t}(z_t, W_{[t,t]}) = f(x_t),$$

and,

$$\Psi_{s,t}(z_s, W_{[s,t]}) = \Psi_{u,t}(\Psi_{s,u}(z_s, W_{[s,u]}), W_{[u,t]}).$$

The proof of Proposition 3.1 is given in Appendix A.2.

3.2 Constructing the Brownian path

The solution map $\Psi_{s,t}$ depends on the Brownian path $W_{[s,t]}$ throughout the $[s, t]$ interval. We must therefore pass this information to the function in a way that captures the rich stochastic path features. McCallum et al. (2026) solve this by constructing a polynomial expansion to Brownian motion. The degree-4 polynomial expansion is shown in Figure 1, where it can be seen to capture the path features over large time intervals.

This polynomial expansion has two essential properties: (1) its coefficients can be sampled exactly without first building the fine Brownian path and (2) polynomials can be composed over sub-intervals $[s, u], [u, t]$ into one polynomial over $[s, t]$. This second property enables an implementation of the semigroup condition. We discuss both of these properties below.

Firstly, the coefficients of this polynomial representation can be sampled independently as

$$I_{s,t}^{(n)} \sim \mathcal{N}\left(0, \frac{t - s}{2n + 1}\right). \tag{9}$$

The zeroth coefficient, $I_{s,t}^{(0)} = W_t - W_s$, specifies the total increment; higher-order coefficients describe how the shape of the path departs from the straight line defined by this increment.

Secondly, the composition over sub-intervals is obtained via the Chen relations, denoted by

$$I_{s,t}^{(n)} = I_{s,u}^{(n)} \oplus I_{u,t}^{(n)}, \tag{10}$$

where, for example, the zeroth coefficient combines as $I_{s,t}^{(0)} = I_{s,u}^{(0)} + I_{u,t}^{(0)}$. Higher order coefficients combine according to similar addition rules. Please see Appendix A.1 for the general expression.

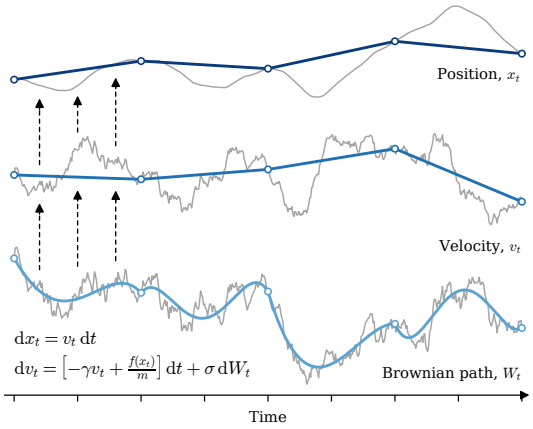

Figure 1: Langevin Flow Maps (*blues*) learn the pathwise solution to Langevin Dynamics (*gray*). Given a polynomial expansion of $W_{[s,t]}$, the flow map takes large timesteps to accurately predict the velocity and position trajectories.

3.3 Training algorithm

The consistency condition demonstrated by Proposition 3.1 enables an objective function which, when minimized, returns the solution map. We therefore look to parameterize the solution map as a neural network $\Psi_{s,t}^{\theta}$ and minimize this objective.

Neural parameterization. Specifically, we implement the Langevin Flow Map as follows. For neural networks $v^\theta_{s,t}, f^\theta_{s,t}$ and polynomial coefficients $I^N_{s,t} = \{I^{(n)}_{s,t}\}_{0\le n<N}$, we write

$$\begin{aligned} x_t &= x_s + (t-s)v^\theta_{s,t}(z_s, I^N_{s,t}) \\ v_t &= v_s - (t-s)\left[\gamma v^\theta_{s,t}(z_s, I^N_{s,t}) - \frac{1}{m} f^\theta_{s,t}(z_s, I^N_{s,t})\right] + \sigma(W_t - W_s), \end{aligned} \tag{11}$$

where $W_t - W_s = I^{(0)}_{s,t}$ by definition of the zeroth coefficient. We define $\Psi^\theta_{s,t} := \Psi_{s,t}[v^\theta, f^\theta]$ to be the application of Equation (11) for a given v^θ, f^θ tuple.

Objective function. Translating the consistency condition into a training objective, we obtain

$$\mathcal{L}(\theta) = \mathcal{L}_{v,f}(\theta) + \mathcal{L}_{\mathrm{D}}(\theta), \qquad \text{where} \tag{12}$$

$$\mathcal{L}_{v,f}(\theta) = \mathbb{E}_{t,v_t,x_t}\left[\|v^\theta_{t,t}(z_t,0) - v_t\|^2 + \|f^\theta_{t,t}(z_t,0) - f(x_t)\|^2\right], \tag{13}$$

$$\mathcal{L}_{\mathrm{D}}(\theta) = \mathbb{E}_{s,u,t,z_s,I}\left[\|\Psi^\theta_{s,t}(z_s, I^N_{s,t}) - \Psi^\theta_{u,t}(\Psi^\theta_{s,u}(z_s, I^N_{s,u}), I^N_{u,t})\|^2\right]. \tag{14}$$

Note that at $s = t$ the coefficients are identically zero, $I^N_{t,t} = 0$; we therefore pass zero directly to the $s = t$ evaluations of the networks.

Evaluation of the training objective requires $[x, v, f(x)]$, plus a randomly sampled time-step $h = t - s$. Since Langevin Dynamics is a time-homogeneous system, no time-dependent data is required for training. This means that we can evaluate the objective with a dataset of uncorrelated-in-time positions and forces, $[x, f(x)]$, only. Correspondingly, the velocities, v, are sampled independently from the Maxwell-Boltzmann distribution at the desired temperature.

Algorithm 1 Langevin Flow Map Training

Require: Dataset $\mathcal{D}$, temperature T, batch size M and split η

 # Sampling

$x^i, f(x^i) \sim \mathcal{D}, \quad v^i \sim$ Maxwell-Boltzmann(T)

 # Matching objective

Sample $(x^i, f(x^i), v^i)$ for $i = 0, \dots, \lfloor \eta M \rfloor$

Compute loss $\mathcal{L}_{v,f}(\theta) = \mathbb{E}[\|v^\theta_0(z^i,0) - v^i\|^2 + \|f^\theta_0(z^i,0) - f(x^i)\|^2]$

 # Semigroup objective

Sample $(x^i, f(x^i), v^i)$ for $i = 0, \dots, M - \lfloor \eta M \rfloor$

Sample timesteps, $h^i \sim \log U(h_{\min}, h_{\max})$

Sample $I^N_{s,u}, I^N_{u,t} \sim \mathcal{N}(0, h^i/(4n+2))$

Chen combine $I^N_{s,t} = I^N_{s,u} \oplus I^N_{u,t}$

Compute EMA teacher $\hat{z}_t = \texttt{stopgrad}[\Psi^{\hat{\theta}}_{u,t}(\Psi^{\hat{\theta}}_{s,u}(z^i, I^N_{s,u}), I^N_{u,t})]$

Compute student $z_t = \Psi^\theta_h(z^i, I^N_{s,t})$

Compute loss $\mathcal{L}_{\mathrm{D}}(\theta) = \mathbb{E}\left[\|z_t - \hat{z}_t\|^2/h^i\right]$

 # Gradient

Update $\theta \leftarrow \theta - \lambda \nabla_\theta(\mathcal{L}_{v,f} + \mathcal{L}_{\mathrm{D}})$

EMA $\hat{\theta} \leftarrow \beta\hat{\theta} + (1-\beta)\theta$

Training algorithm. The full training algorithm is presented in Algorithm 1. For the semigroup loss we use the self-distillation formulation, where the two-step map is implemented using an exponential moving average (EMA) of the network's weights. This EMA network is called the teacher. We apply `stopgrad` to the output of the two-step teacher following common practice (Boffi et al., 2025a; McCallum et al., 2026). These two choices have been shown to stabilize training with self-distillation objectives (Frans et al., 2025).

4 Experiments

The experiments are ordered in two sections: single-molecule dynamics and transferable dynamics. The single-molecule section trains an LFM to learn the forces and solution map of each molecule individually, where we use the MD17 small-molecules dataset from Chmiela et al. (2017) and the alanine dipeptide dataset from Köhler et al. (2021). The transferable section applies one LFM model to a large dataset of peptide sequences from Tan et al. (2025a) and investigates the integration performance when generalizing to unseen peptides.

4.1 Single-molecule Dynamics

4.1.1 Organic molecules

We first investigate the performance of the LFM for large-timestep molecular dynamics when trained on single-molecule datasets. We use the MD17 dataset, which releases equilibrium configurations and forces for a set of organic molecules computed via density functional theory (Chmiela et al., 2017).

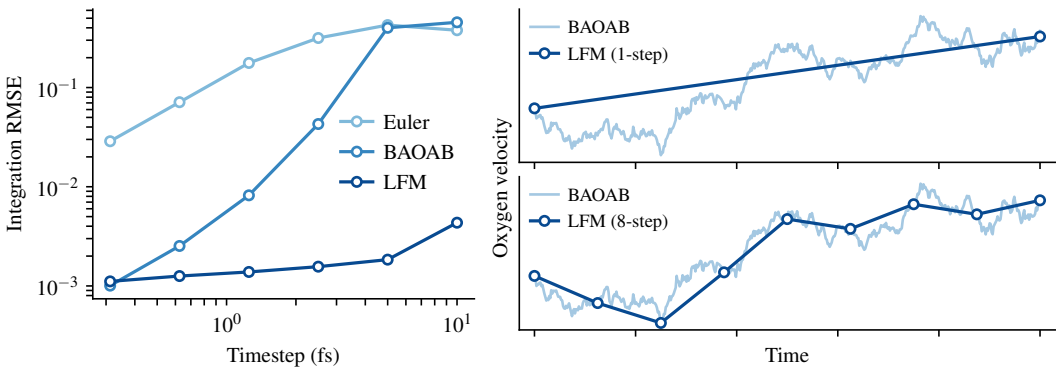

Figure 2: *Left*. Integration RMSE for the learned ethanol dynamics. *Right*. Pathwise solution comparison between the fine reference BAOAB trajectory and the large-timestep LFM prediction.

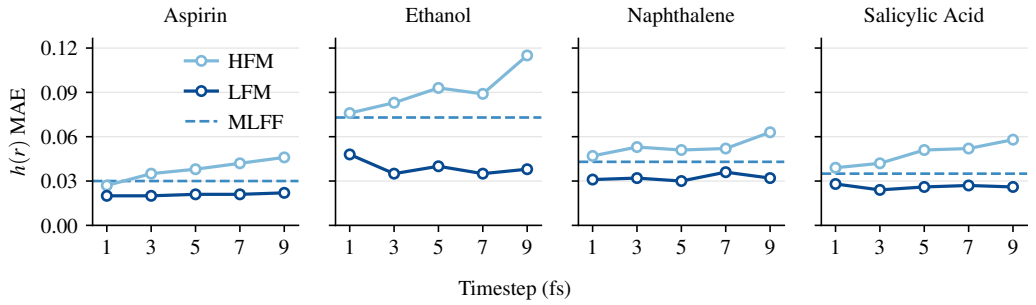

Figure 3: Interatomic distances $h(r)$ MAE as a function of step size for 300 ps of MD simulation. Reference MLFF shown with 0.5fs timestep integration.

Integration Error. To measure the accuracy of the pathwise solution, we compute the integration root-mean-squared-error (RMSE) between the predicted solution and a fine-timestep reference. The results for the ethanol molecule are shown in Figure 2. We see that the LFM at 10fs timesteps obtains comparable accuracy to BAOAB at 1fs, while also improving upon the small-timestep accuracy.

Molecule Structure. Next we compare the molecular structure predicted by the LFM as a function of the timestep. The distribution of interatomic distances, $h(r)$, is calculated for the predicted molecules and compared to the reference dataset. The LFM is compared to Hamiltonian Flow Maps and the reference MLFF baseline (Ripken et al., 2026); see Figure 3.

We see that the MAE obtained by the LFM predictions is lower than that obtained by the HFM for all timesteps and molecules. Additionally, the LFM predictions achieve a lower MAE than the reference MLFF at 0.5fs timesteps. This demonstrates that the LFM has not only learned an accurate large-timestep integrator, but is able to improve upon the integration error at the small time steps commonly used to run molecular dynamics.

Force Error. The LFM implements the standard force matching loss to constrain the $h = 0$ force network. We can therefore compare the force accuracy obtained by the LFM to standard MLFF training. This is shown in Table 2. We see that the force error obtained by the MLFF is lower than the LFM, as expected. However, when analyzed alongside Figure 3 where the LFM obtains a higher structural accuracy than the MLFF, it becomes clear that force error alone is not a sufficient training objective for accurate molecular dynamics. This result verifies the sentiment developing in the MLFF literature that 'forces are not enough' (Fu et al., 2023).

Table 2: Force MAE in units of meV/Å.

Molecule	MLFF	HFM	LFM
Aspirin	**3.8**	7.0	5.9
Ethanol	**1.5**	2.8	2.9
Naphthalene	**1.5**	3.0	2.8
Salicylic acid	**2.2**	4.7	4.3

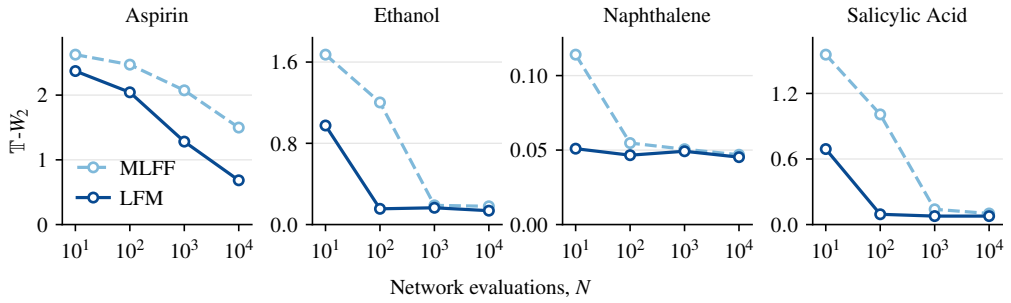

Figure 4: Convergence to the Boltzmann distribution measured by the 2-Wasserstein distance over torsion angles, $\mathbb{T}$-W_2, as a function of the number of network evaluations, N.

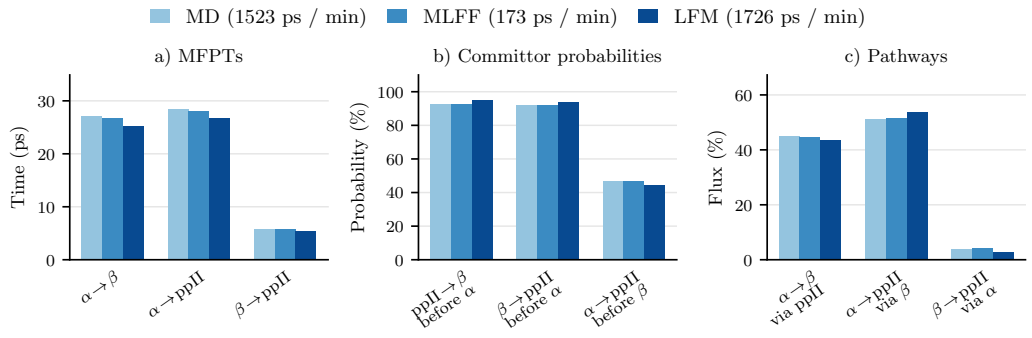

Figure 5: Alanine dipeptide transition dynamics for trajectories generated by each method. Integration throughput is shown in picoseconds per minute, measured on 1x RTX A6000.

Boltzmann Convergence. A key property of Langevin Dynamics is that the sample distribution converges to the Boltzmann distribution. This convergence occurs over long timescales and therefore many function evaluations of the force-field are required. To compare the same infinitesimal dynamics, we define the MLFF by the LFM's force network evaluated at $h = 0$. We compare the Boltzmann convergence speed of the MLFF integrated with BAOAB at 0.5fs and the LFM at 9fs in terms of the number of function evaluations. We measure convergence by the 2-Wasserstein distance on the torsional molecule angles from a biased initialization; see Figure 4. We see that the convergence speed of the LFM is faster than the integrated MLFF. For example, the ethanol / naphthalene / salicylic acid distributions predicted by the LFM have converged after 100 function evaluations compared to the 1,000 evaluations required by the MLFF.

4.1.2 Alanine Dipeptide

We next train the LFM on the canonical alanine dipeptide system which admits multiple meta-stable torsional states with slow mixing times (Köhler et al., 2021). We use this system to investigate the LFMs ability to recover accurate dynamical properties and sample transition pathways.

Transition dynamics. In Figure 5, we show the transition dynamics between meta-stable states predicted by the LFM alongside the MLFF and reference MD trajectory. We see that the transition dynamics obtained by the LFM are comparable to the MLFF, while performing the calculation significantly faster. Additionally, the LFM predicts transition metrics accurate to classical MD using the Amber99SB force-field and even improves upon the integration speed. This indicates the possibility for obtaining QM-accurate molecular dynamics at comparable speed to classical mechanics.

Pathwise observables. In Figure 6, the transition paths between the α and β states are shown alongside the free energy observed during the transition. Specifically, a batch of molecules are seeded in the α state and integrated with the LFM until reaching the β state. This enables the tran-

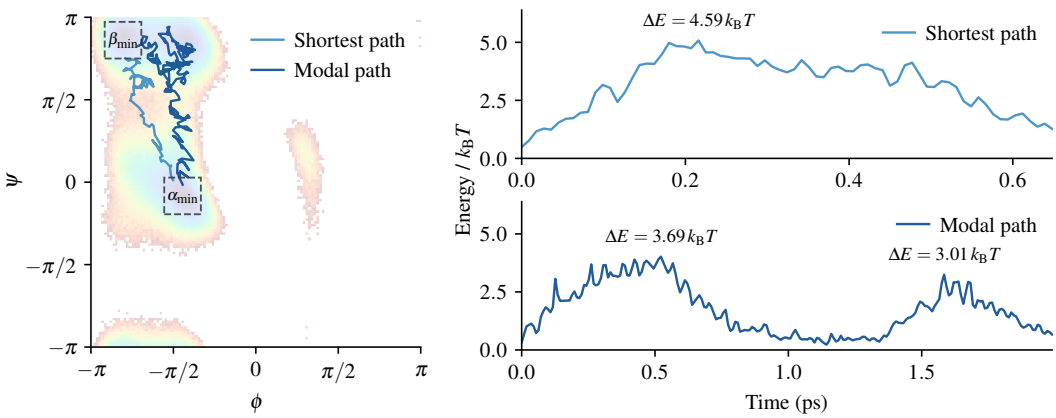

Figure 6: Transition paths for the $\alpha \to \beta$ transition in alanine dipeptide. Activation energies are shown for each free energy maximum, measured from the preceding free energy minima.

sition time density to be calculated. Then, the shortest path and modal path are extracted from this distribution. Interestingly, we see that the shortest path walks the transition by directly overcoming one large activation energy barrier whereas the modal path transitions via polyproline (ppII) and overcomes two smaller activation energy barriers.

4.2 Transferable Dynamics

The LFM architecture requires only atomic numbers to define the molecular system and therefore naturally extends to modelling a diverse molecular input space. We use this property to train a transferable LFM on a diverse set of peptide conformations from the Many Peptides dataset (Tan et al., 2025a) and investigate the extent to which the LFM can generalize to accurately integrate unseen peptide sequences. We investigate the integration error, Boltzmann convergence and transition dynamics predicted by the model for held-out peptide sequences. See Appendix C for details.

Integration Error. In Figure 7, we calculate the integration RMSE of the transferable LFM on 92 unseen peptide sequences, ranging from dipeptides through to decapeptides. We compute the error throughout training as measured by the number of atoms per parameter (A/N). We see that at 1fs timesteps the LFM obtains comparable integration error to BAOAB, while maintaining a significantly lower error up to 10fs timesteps.

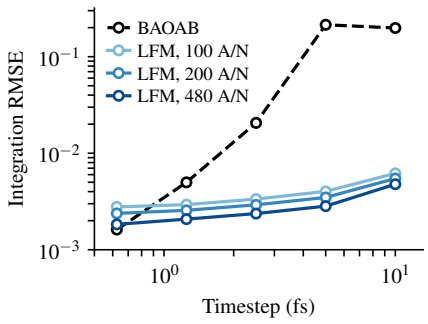

Figure 7: Transferable LFM integration error on *Many Peptides*.

Boltzmann convergence. In Table 3, we investigate the LFMs ability to recover the Boltzmann distribution when initialized from a single conformation of an unseen peptide sequence. We integrate the peptide for 10^7 function evaluations and report the cumulative energy and backbone torsion 2-Wasserstein distances ($\mathcal{E}$-W_2, $\mathbb{T}$-W_2). We see that the ensemble metrics improve as the LFMs timestep decreases, with BAOAB integration on the LFMs $h = 0$ force network returning metrics comparable to the ground-truth Amber14 force-field.

Transition dynamics. Additionally, in Table 3, we calculate the meta-stable state occupation, identified by a time-lagged independent component analysis (TICA), and compute transition probabilities between these states. We compute the total variation over the state occupation π-TV and transition $\mathbb{P}$-TV densities with respect to the reference dataset. The large-timestep LFM metrics consistently outperform the Amber14 force-field, obtaining more accurate meta-stable state occupation and transition probabilities.

Table 3: Transferable LFM sampling on nine unseen peptide sequences with length 2, 4, 8 from *Many Peptides*. Each metric is reported as the mean across three peptides at each length.

			2AA				4AA				8AA			
Forces	Integrator	Timestep	$\mathcal{E}$-W_2	$\mathbb{T}$-W_2	π-TV	$\mathbb{P}$-TV	$\mathcal{E}$-W_2	$\mathbb{T}$-W_2	π-TV	$\mathbb{P}$-TV	$\mathcal{E}$-W_2	$\mathbb{T}$-W_2	π-TV	$\mathbb{P}$-TV
LFM	LFM	8 fs	3.64	0.81	0.13	0.10	11.75	2.31	0.26	0.19	25.91	4.17	**0.03**	**0.45**
LFM	LFM	4 fs	2.73	0.61	**0.07**	**0.04**	8.76	2.10	**0.08**	**0.18**	20.90	4.08	0.03	0.49
LFM	LFM	2 fs	1.81	0.62	0.28	0.06	7.45	1.95	0.14	0.32	18.57	4.27	0.19	0.48
LFM	LFM	1 fs	1.20	0.73	0.21	0.07	5.08	1.70	0.15	0.43	11.45	4.26	0.05	0.47
LFM	BAOAB	1 fs	0.49	0.44	0.21	0.13	1.22	1.42	0.34	0.31	6.10	**3.18**	0.07	0.49
Amber14	BAOAB	1 fs	**0.30**	**0.35**	0.15	0.12	**1.04**	**1.13**	0.18	0.27	**4.04**	3.41	0.07	0.47

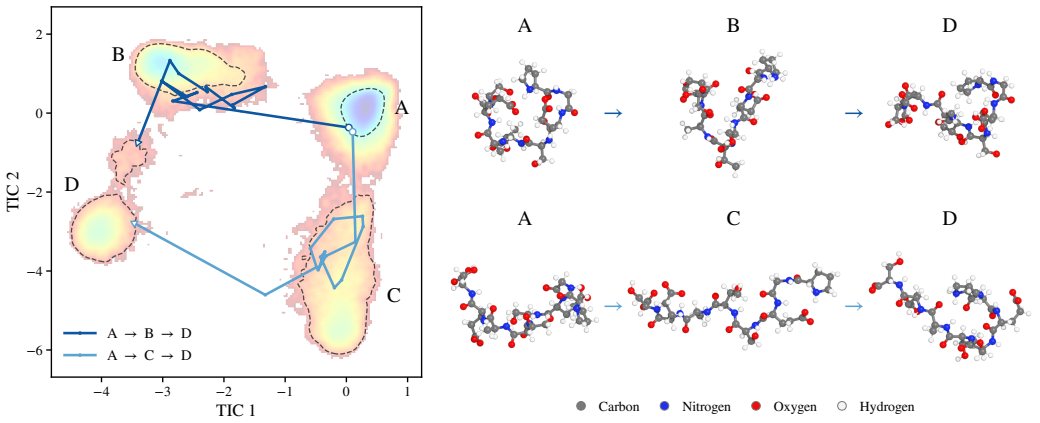

Figure 8: *Left*. Transition pathways for octapeptide PGESTAES predicted by the LFM and projected onto the first two time-lagged independent components (TIC). *Right*. Peptide conformations measured along each transition pathway.

State discovery. We see from Table 3 that the small-timestep dynamics more accurately represent the Boltzmann ensemble whereas the large-timestep dynamics best represent the meta-stable state occupation and transition dynamics. This presents the opportunity to initially explore the state-space with large LFM timesteps and perform a final relaxation with small-timestep integration. We demonstrate the advantage of this procedure in Figure 13, where the LFM discovers a meta-stable mode not reached by long molecular dynamics rollouts.

Transition pathways. We can apply the transferable LFM to calculate transition paths between meta-stable states defined by the TICA. In Figure 8, we show the free energy surface for the PGESTAES octapeptide and the four meta-stable states obtained via clustering. To obtain transition pathways, the conformer is seeded in state A and integrated with the LFM at 4fs until reaching state D. We plot two coarsened pathways that transition via states B and C with the corresponding molecular conformations along each path. This peptide sequence was not seen by the LFM during training, yet we see that the model is able to recover consistent transition pathways.

5 Conclusions

In this work, we have introduced a method to learn the solution map of Langevin dynamics, termed Langevin Flow Maps (LFMs). This construction extends machine-learned force fields to learn both the instantaneous forces and the integrated system dynamics directly without access to explicit trajectory data. We demonstrated that LFMs enable large-timestep molecular dynamics while predicting accurate ensemble properties and transition dynamics. Further, a transferable LFM model trained on a diverse molecular dataset was shown to generalize to unseen peptide sequences, indicating the possibility of a foundational molecular dynamics integrator.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing code to run the experiments, searching the literature for related work and text editing to improve clarity. All code, literature and text edits were checked manually for correctness. We take responsibility for the final content of this work, including text, claims and artifacts produced.

ETHICS STATEMENT

This work contributes to the field of molecular dynamics by improving simulation efficiency. The method may support applications in chemistry, materials science and drug design. We do not anticipate additional ethical concerns from this work beyond those established in the molecular dynamics literature.

REPRODUCIBILITY STATEMENT

We plan to release the code required to reproduce the experimental results on GitHub. Model architectures, training settings and evaluation procedures are described in the appendix. We also plan to release the trained transferable model checkpoints and the forces computed for the Many Peptides dataset.

REFERENCES

Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98(14):146401, 2007.

Filippo Bigi, Sanggyu Chong, Agustinus Kristiadi, and Michele Ceriotti. FlashMD: long-stride, universal prediction of molecular dynamics. *Advances in Neural Information Processing Systems*, 38:88537–88574, 2025.

Nicholas Boffi, Michael Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *Advances in Neural Information Processing Systems*, 38: 33346–33382, 2025a.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025b.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017.

Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, 9 (1):3887, 2018.

Avishek Das, Dominic C Rose, Juan P Garrahan, and David T Limmer. Reinforcement learning of rare diffusive dynamics. *The Journal of Chemical Physics*, 155(13), 2021.

Felix Dietrich, Alexei Makeev, George Kevrekidis, Nikolaos Evangelou, Tom Bertalan, Sebastian Reich, and Ioannis G Kevrekidis. Learning effective stochastic differential equations from microscopic simulations: Linking stochastic numerics to deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(2), 2023.

Juan Viguera Diez, Mathias Schreiner, and Simon Olsson. Transferable generative models bridge femtosecond to nanosecond time-step molecular dynamics. *Science Advances*, 12(15):eaed2333, 2026.

Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noe, Carla P Gomes, Alan Aspuru-Guzik, and Kirill Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. *Advances in Neural Information Processing Systems*, 37:65791–65822, 2024.

James Foster, Terry Lyons, and Harald Oberhauser. An optimal polynomial approximation of Brownian motion. *SIAM Journal on Numerical Analysis*, 58(3):1393–1421, 2020.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Academic Press, 1996.

Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *Transactions on Machine Learning Research*, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2025.

Herbert Goldstein, Charles P. Poole, and John Safko. *Classical Mechanics*. Pearson, 3 edition, 2011.

Karen Habermann. A semicircle law and decorrelation phenomena for iterated Kolmogorov loops. *Journal of the London Mathematical Society*, 103(2):558–586, 2021.

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*, 2021.

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. *International Conference on Machine Learning*, 2026.

Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6): 1129–1143, 2018.

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36:11102–11126, 2023.

John A Keith, Valentin Vassilev-Galindo, Bingqing Cheng, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chemical Reviews*, 121(16):9816–9872, 2021.

Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry J Lyons. Neural SDEs as infinite-dimensional GANs. In *International conference on machine learning*, pp. 5453–5463. PMLR, 2021.

Naoki Kiyohara, Edward Johns, and Yingzhen Li. Neural stochastic flows: Solver-free modelling and inference for SDE solutions. *Advances in Neural Information Processing Systems*, 38:16917–16962, 2025.

Leon Klein and Frank Noé. Transferable boltzmann generators. *Advances in Neural Information Processing Systems*, 37:45281–45314, 2024.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36:52863–52883, 2023a.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910, 2023b.

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021.

Benedict Leimkuhler and Charles Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013(1):34–56, 2013.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International conference on artificial intelligence and statistics*, pp. 3870–3882. PMLR, 2020.

Sam McCallum, Zander W Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong stochastic flow maps. *arXiv preprint arXiv:2606.01086*, 2026.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71(1):361–390, 2020.

Richard W Pastor, Bernard R Brooks, and Attila Szabo. An analysis of the accuracy of Langevin and molecular dynamics algorithms. *Molecular Physics*, 65(6):1409–1419, 1988.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. *arXiv preprint arXiv:2601.14430*, 2026.

Sanjeev Raja, Martin Šípka, Michael Psenka, Tobias Kreiman, Michal Pavelka, and Aditi S Krishnapriyan. Action-minimization meets generative modeling: Efficient transition path sampling with the onsager-machlup functional. *International Conference on Machine Learning*, 2025.

Danyal Rehman, Oscar Davis, Jiarui Lu, Jian Tang, Michael Bronstein, Yoshua Bengio, Alexander Tong, and Joey Bose. Efficient regression-based training of normalizing flows for Boltzmann generators. In *International Conference on Learning Representations*, volume 2026, pp. 127171–127194, 2026a.

Danyal Rehman, Charlie B Tan, Yoshua Bengio, Avishek Joey Bose, and Alexander Tong. Autoregressive Boltzmann generators. *International Conference on Machine Learning*, 2026b.

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. *International Conference on Machine Learning*, 2026.

Dominic C Rose, Jamie F Mair, and Juan P Garrahan. A reinforcement learning approach to rare trajectory sampling. *New Journal of Physics*, 23(1):013013, 2021.

Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108(5):058301, 2012.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. *Advances in Neural Information Processing Systems*, 36:36449–36462, 2023.

Aditya N Singh and David T Limmer. Variational deep learning of equilibrium transition path ensembles. *The Journal of Chemical Physics*, 159(2), 2023.

Martin Sipka, Johannes CB Dietschreit, Lukáš Grajciar, and Rafael Gómez-Bombarelli. Differentiable simulations for enhanced sampling of rare events. In *International Conference on Machine Learning*, pp. 31990–32007. PMLR, 2023.

Charlie Tan, Majdi Hassan, Leon Klein, Saifuddin Syed, Dominique Beaini, Michael Bronstein, Alexander Tong, and Kirill Neklyudov. Amortized sampling with transferable normalizing flows. *Advances in Neural Information Processing Systems*, 38:94290–94325, 2025a.

Charlie B Tan, Avishek Joey Bose, Chen Lin, Leon Klein, Michael M Bronstein, and Alexander Tong. Scalable equilibrium sampling with sequential Boltzmann generators. *International Conference on Machine Learning*, 2025b.

Fabian L Thiemann, Thiago Reschützegger, Massimiliano Esposito, Tseden Taddese, Juan D Olarte-Plata, and Fausto Martelli. Force-free molecular dynamics through autoregressive equivariant networks. *Nature Machine Intelligence*, pp. 1–13, 2026.

Mark E Tuckerman and Glenn J Martyna. Understanding modern molecular dynamics: Techniques and applications. *The Journal of Physical Chemistry B*, 104(2):159–178, 2000.

Oliver T Unke, Stefan Chmiela, Michael Gastegger, Kristof T Schütt, Huziel E Sauceda, and Klaus-Robert Müller. Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature communications*, 12(1):7273, 2021a.

Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müuller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021b.

Yuanqing Wang, Kenichiro Takaba, Michael S Chen, Marcus Wieder, Yuzhi Xu, Tong Zhu, John ZH Zhang, Arnav Nagle, Kuang Yu, Xinyan Wang, et al. On the design space between molecular mechanics and machine learning force fields. *Applied Physics Reviews*, 12(2), 2025.

A THEORY

A.1 BROWNIAN MOTION

First introduced by Foster et al. (2020) and Habermann (2021), the degree-N polynomial expansion of Brownian motion over $[s, t]$ is given by,

$$W_{u,v}^{(N)} = \sum_{n=0}^{N-1} \frac{2n+1}{t-s} I_{s,t}^{(n)} \int_u^v \widetilde{P}_n \left(\frac{r-s}{t-s} \right) dr, \tag{15}$$

where $\widetilde{P}_n$ is the n-th shifted Legendre polynomial on $[0, 1]$ and $I_{s,t}^{(n)}$ are the polynomial coefficients given by,

$$I_{s,t}^{(n)} = \int_s^t \widetilde{P}_n \left(\frac{u-s}{t-s} \right) dW_u. \tag{16}$$

For example, taking the degree-3 representation we get

$$W_{s,v}^{(3)} = I_{s,t}^{(0)} x + 3I_{s,t}^{(1)}(x^2 - x) + 5I_{s,t}^{(2)}(2x^3 - 3x^2 + x), \qquad x = \frac{v-s}{t-s}. \tag{17}$$

Properties of $I_{s,t}$. Since the coefficients of the polynomial expansion are defined by the stochastic integral in Equation (16), they are normally distributed and can be sampled directly (McCallum et al., 2026). Specifically, the coefficients are distributed by

$$I_{s,t}^{(n)} \sim \mathcal{N}\left(0, \frac{t-s}{2n+1}\right). \tag{18}$$

Additionally, these coefficients can be combined over sub-intervals, $[s, u], [u, t]$, to form the coefficients over the full interval, $[s, t]$. This is achieved by the following Chen relations,

$$I_{s,t}^{(n)} = \sum_{m=0}^{n} c_{n,m} \Big(I_{s,u}^{(m)} + (-1)^{n+m} I_{u,t}^{(m)} \Big), \tag{19}$$

where,

$$c_{n,m} := (-1)^n (2m+1) \sum_{k=m}^{n} \Big(-\frac{1}{2} \Big)^k \frac{(n+k)!}{(n-k)!(k-m)!(k+m+1)!}. \tag{20}$$

For example, the $n = 0$ and $n = 1$ coefficients combine as

$$I_{s,t}^{(0)} = I_{s,u}^{(0)} + I_{u,t}^{(0)}, \tag{21}$$

$$I_{s,t}^{(1)} = \frac{1}{2} \left(I_{s,u}^{(1)} + I_{u,t}^{(1)} \right) + \frac{1}{2} \left(I_{u,t}^{(0)} - I_{s,u}^{(0)} \right). \tag{22}$$

A.2 CONSISTENCY CONDITION

In this section we provide a short proof of Proposition 3.1. This follows directly from McCallum et al. (2026, Proposition 3.3). The statement is reproduced here for convenience. We inherit the standard regularity assumptions on the drift and diffusion fields (McCallum et al., 2026, Assumption A.1).

Proposition A.1 (McCallum et al. (2026, Proposition 3.3))**.** Let $\Psi_{s,t}(z_s, W_{[s,t]})$ denote the solution map of the following additive-noise SDE,

$$dz_t = b(t, z_t)dt + g(t)dW_t, \tag{23}$$

which be written as

$$\Psi_{s,t}(z_s, W_{[s,t]}) = z_s + b_{s,t}(z_s, W_{[s,t]})(t-s) + g_{s,t}(W_{[s,t]})(W_t - W_s). \tag{24}$$

Then $\Psi_{s,t}$ is the solution map if and only if

$$b_{t,t}(z_t) = b(t, z_t), \quad g_{t,t} = g(t), \quad \text{and} \tag{25}$$

$$\Psi_{s,t}(z_s, W_{[s,t]}) = \Psi_{u,t}(\Psi_{s,u}(z_s, W_{[s,u]}), W_{[u,t]}). \tag{26}$$

To show sufficiency of Proposition 3.1 for obtaining the solution map to Langevin Dynamics, it is therefore sufficient to demonstrate the equivalence to Proposition A.1. First, consider that the Langevin Dynamics SDE can be written in the form of Equation (23) as

$$d\begin{bmatrix} x_t \\ v_t \end{bmatrix} = \begin{bmatrix} v_t \\ -\gamma v_t + m^{-1} f(x_t) \end{bmatrix} dt + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} dW_t. \tag{27}$$

Then, the matching condition in Equation (25) requires

$$b_{t,t}(z_t) = \begin{bmatrix} v_t \\ -\gamma v_t + m^{-1} f(x_t) \end{bmatrix}, \tag{28}$$

which is equivalent to $v_{t,t}(z_t, W_{[t,t]}) = v_t$ and $f_{t,t}(z_t, W_{[t,t]}) = f(x_t)$ with $W_{[t,t]} := \varnothing$.

Finally, the semigroup condition Equation (26) required by Proposition A.1 is retained exactly in Proposition 3.1. This completes the proof.

B Model Architecture

Following the Hamiltonian Flow Map architecture of Ripken et al. (2026), we parameterize $v^\theta_{s,t}$ and $f^\theta_{s,t}$ using a shared translation-invariant transformer. The architectural addition required to learn the Langevin Flow Map is to pass the Brownian polynomial coefficients $I^N_{s,t}$ as conditioning information. We describe the full architecture here for completeness.

Embeddings. For atom i with atomic number Z_i, we initialize a token $h_i^{(0)}$ and construct relative positional embeddings,

$$h_i^{(0)} = \mathrm{Embed}(Z_i), \qquad e_{ij} = \mathrm{MLP}\left([\phi(d_{ij}); \hat{r}_{ij}]\right), \tag{29}$$

where $d_{ij} = \|x_i - x_j\|$, $\hat{r}_{ij} = (x_i - x_j)/d_{ij}$ and ϕ is a cosine basis expansion. Here x_i denotes the initial position and $[\cdot;\cdot]$ denotes concatenation.

Attention. For a single attention head, the positional embeddings enter the attention operation via

$$\begin{aligned} a_{ij} &= \frac{(W_Q h_i)^\top (W_K h_j \odot W_{K,e} e_{ij})}{\sqrt{d_h}}, \\ \mathrm{Attn}(h)_i &= \sum_j \mathrm{softmax}_j(a_{ij}) \left(W_V h_j \odot W_{V,e} e_{ij}\right), \end{aligned} \tag{30}$$

where $\odot$ denotes element-wise multiplication.

Conditioning. The time-step $h = t - s$, initial velocity v_i and mass m_i are embedded as

$$c^h = \mathrm{MLP}_h(\phi_h(h)), \quad c^m_i = \mathrm{MLP}_m(m_i), \quad c^v_i = \mathrm{MLP}_v\left([W_{Z_i}\psi_v(\|v_i\|); v_i]\right), \tag{31}$$

where ϕ_h denotes Gaussian Fourier features with learnable frequencies, ψ_v is a Gaussian basis expansion and W_{Z_i} is an element-specific matrix. We add a Brownian embedding using the coefficient vectors $I^{(n)}_{s,t,i} \in \mathbb{R}^3$ associated with atom i via

$$c^I_i = \mathrm{MLP}_I\left(\left[\psi_I\left(\|I^{(n)}_{s,t,i}\|\right); I^{(n)}_{s,t,i}\right]\right), \tag{32}$$

and obtain the total conditioning input with

$$c_i = \mathrm{MLP}\left([c^h; c^v_i; c^I_i; c^m_i]\right). \tag{33}$$

Each transformer block uses c_i to compute separate scale, shift and gating parameters for the hidden state, given by

$$\begin{aligned} [\gamma_i; \beta_i; \alpha_i] &= W\,\mathrm{SiLU}(\mathrm{LN}(c_i)) + b, \\ \mathrm{AdaLN}(h_i; \gamma_i, \beta_i) &= (1 + \gamma_i) \odot \mathrm{LN}(h_i) + \beta_i, \\ h_i &\leftarrow h_i + \alpha_i \odot \mathrm{Block}\left(\mathrm{AdaLN}(h; \gamma, \beta)\right)_i, \end{aligned} \tag{34}$$

where Block denotes the attention or feed-forward update.

Prediction heads. After L transformer blocks, separate heads predict the mean velocity and force,

$$v^{\theta}_{s,t,i} = \mathrm{MLP}^{\mathrm{out}}_{v}\left(\mathrm{AdaLN}(h^{(L)}_i;\gamma^v_i,\beta^v_i)\right),$$
$$f^{\theta}_{s,t,i} = \mathrm{MLP}^{\mathrm{out}}_{f}\left(\mathrm{AdaLN}(h^{(L)}_i;\gamma^f_i,\beta^f_i)\right). \tag{35}$$

The task-specific scale and shift parameters are obtained from c_i as above. These predictions define the Langevin Flow Map through Equation (11). The hyperparameters used for this architecture are given in Table 4.

Table 4: Model hyperparameters.

Hyperparameter	Value
Transformer blocks	6
Attention heads	8
Hidden dimension	256
Distance basis functions	10
Velocity Gaussian basis functions	8
Brownian Gaussian basis functions	8
Brownian coefficients	4
MLP layers	2
Feed-forward hidden width	1024

C Experimental Details

C.1 Single-molecule Dynamics

We train an LFM for each MD17 molecule at 500 K with friction $\gamma = 10\,\mathrm{ps}^{-1}$. The model hyperparameters are shown in Table 4. The dataset is split into training, validation and test sets in an 80:10:10 ratio, with the same split as Ripken et al. (2026).

During training, each configuration is centered and a random rotation is applied. The models are trained for 100,000 updates with batch size 512 using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.95$) and learning rate warmup from 10^{-6} to 10^{-3} over 2,000 steps before cosine decay to a final learning rate of 10^{-7}. The fraction of the batch allocated to the matching objective is $\eta = 0.25$ and we use EMA decay 0.99, gradient clipping at norm 10 and log-uniform timesteps between 0.01 and 10fs.

Molecule Structure. We initialize 256 trajectories from test configurations with thermal velocities and integrate for 300ps. Following Ripken et al. (2026), the interatomic distance distribution for a configuration x of N atoms is

$$h(r) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \delta(r - \|x_i - x_j\|_2), \tag{36}$$

where δ is the Dirac delta.

The MAE between the generated distribution, $h(r)$, and the reference dataset distribution $\hat{h}(r)$ is calculated as

$$\mathrm{MAE}_{h(r)} = \int_0^{\infty} \left| \langle h(r) \rangle - \langle \hat{h}(r) \rangle \right| dr, \tag{37}$$

where $\langle \cdot \rangle$ denotes the mean. Frames from the first bond-length deviation exceeding 0.5 Å from the mean are excluded from each trajectory (Ripken et al., 2026; Fu et al., 2023).

Boltzmann Convergence. We initialize 256 trajectories at the same configuration with independently sampled velocities and Brownian coefficients. Integration is performed for $N = 10{,}000$ network evaluations with both the LFM and BAOAB. At each evaluation, we compare the torsion-angles of the ensemble to 256 configurations sampled from the reference dataset using the periodic 2-Wasserstein distance.

C.2 Alanine Dipeptide

We use the same data split, architecture and training settings as for the single-molecule experiments, but with a temperature of 300 K and 200,000 training steps.

Transition dynamics. We initialize 256 trajectories and integrate for 1ns with the LFM at 10fs and BAOAB with the LFM's $h = 0$ force network at 1fs. The MD reference is a 1μs OpenMM trajectory after 1ns equilibration, using Amber99SB-ILDN with OBC implicit solvent and a 1fs timestep. Frames are saved every 1ps.

Each trajectory is fit with a maximum-likelihood Markov State Model (MSM) using a 1 ps lag. The discrete states of the MSM are chosen to be the α, β, ppII meta-stable states with an additional state to handle the remaining state-space. The backbone angles (ϕ, ψ) of each state are defined by: α with $\phi \in [-160, -20]^\circ$ and $\psi \in [-120, 50]^\circ$; β with $\phi \in [160, 270]^\circ$ and $\psi \in [50, 240]^\circ$; ppII with $\phi \in [-90, -20]^\circ$ and $\psi \in [50, 240]^\circ$.

Pathwise observables. We initialize 256 trajectories from the α state and integrate with the LFM at 9fs timesteps until first entry into the β state. We show the shortest path and the path nearest the mode of the transition time density. The plotted free energy is obtained from the reference-data histogram.

C.3 Transferable Dynamics

C.3.1 Training & Inference

We calculated forces for conformers within the *Many Peptides* dataset (Tan et al., 2025a), spanning 430M conformers of 21k peptides of length 2-8. The 9M parameter LFM model was trained with hyperparameters from Table 4 at three training compute budgets, expressed in atoms per parameters (100, 200, and 480).

We use the same single-molecule architecture and training setup described above, but supply input atomic numbers and masses for each molecular configuration. To batch systems with different numbers of atoms, we pad the atom arrays to a maximum size and mask padded atoms throughout the transformer and flow map forward pass. At inference, we additionally apply a temperature corrector algorithm to stabilize the kinetic temperature, described below.

Temperature corrector. The learned flow map velocity and force heads are not perfectly trained to zero loss. The remaining bias enters each update and over many steps can introduce a bias to the kinetic temperature. This can be corrected by making small changes to the Langevin thermostat at each step.

For a group of atoms with mass m, let v_i denote the predicted velocities and $n_i = \sigma I^{(0)}_{s,t,i}$ the explicit noise updates. We look to obtain corrected velocities $\hat{v}_i = v_i + \alpha n_i$ satisfying

$$\mathbb{E}[\|\hat{v}_i\|^2] = R, \qquad R = 3k_B T/m. \tag{38}$$

Then, defining

$$A = \mathbb{E}[\|v_i\|^2], \qquad B = \mathbb{E}[v_i \cdot n_i], \qquad C = \mathbb{E}[\|n_i\|^2], \tag{39}$$

and expanding Equation (38), we get

$$C\alpha^2 + 2B\alpha + A - R = 0. \tag{40}$$

This quadratic can be solved to obtain a real-valued α that corrects velocities to the target temperature T. Since the corrected noise update is given by $(1 + \alpha)\sigma I^{(0)}_{s,t,i}$ with $1 + \alpha \geq 0$, this update simply scales the noise magnitude and preserves the Langevin update. If no permissible root exists, then we fallback to a velocity rescaling $\hat{v}_i = \sqrt{R/A}v_i$.

C.3.2 Metrics

Ensemble metrics. We evaluate the cumulative distribution of configurations from 10^7 integration steps against the reference dataset. We compute the energy $\mathcal{E}$-W_2 and backbone torsion $\mathbb{T}$-W_2 2-

Wasserstein distance metrics, following Tan et al. (2025a); Rehman et al. (2026b). The energy function is evaluated using the Amber14 force field with OBC1 implicit solvent.

Transition metrics. To enable a systematic comparison of the transition dynamics across peptides and solvers, we measure the total variation of meta-stable state occupancy π-TV and transition probabilities $\mathbb{P}$-TV. This requires (1) identifying the meta-stable states and (2) fitting a Markov State Model (MSM) over these discrete states for each trajectory.

To identify the meta-stable states, we compute a time-lagged independent component analysis (TICA) for each peptide sequence using the test set reference trajectory from *Many Peptides* (Tan et al., 2025a). We then use the deeptime package (Hoffmann et al., 2021) to cluster the TICA into K meta-stable states. Next, we fit the MSM with a maximum likelihood objective over the identified states for each generated trajectory (Hoffmann et al., 2021).

The MSM can then be queried to obtain the occupation probability for each state π_i and transition probabilities between states $i \to j$ at a lag time τ, denoted by P_{ij}. Given a reference $(\hat{\pi}_i, \hat{P}_{ij})$ and predicted (π_i, P_{ij}), the π-TV and $\mathbb{P}$-TV metrics are calculated by

$$\begin{aligned} \pi\text{-TV} &= \frac{1}{2}\sum_{i=1}^{K} \left|\hat{\pi}_i - \pi_i\right|, \\ \mathbb{P}\text{-TV} &= \frac{1}{2K}\sum_{i=1}^{K}\sum_{j=1}^{K} \left|\hat{P}_{ij} - P_{ij}\right|. \end{aligned} \tag{41}$$

D Additional Results

D.1 Single-molecule dynamics

Polynomial degree. In Figure 9 we ablate the effect of the polynomial degree on the integration RMSE and interatomic distances. We see that as the degree increases the integration RMSE and interatomic distance MAE decrease, with the most pronounced decrease at large timesteps. The effect is most significant when moving from increment-only information $W^{(1)}$ to inclusion of the first higher-order coefficient $W^{(2)}$.

Boltzmann convergence. In Figure 10 we plot the Boltzmann convergence dynamics for the aspirin molecule over the state-space defined by the two torsion angles. The LFM (9fs) and BAOAB (0.5fs) integrators are initialized from a biased start and iterated for 10,000 function evaluations. At matched function evaluations, the state-space coverage obtained by the LFM is significantly increased in comparison to BAOAB, as expected.

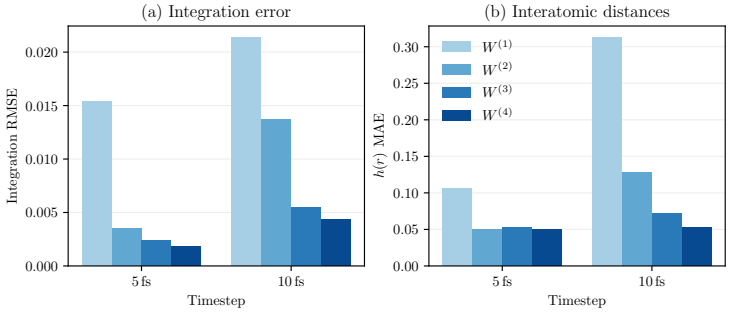

Figure 9: Polynomial degree ablation on the ethanol dataset. The degree is varied over $N = 1, 2, 3, 4$ with the polynomial expansion denoted by $W^{(N)}$.

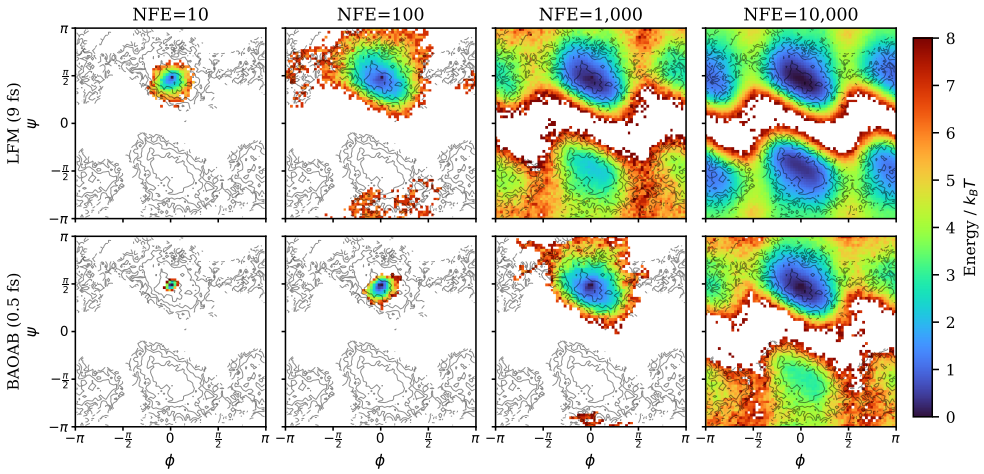

Figure 10: Torsional state-space exploration for the aspirin molecule. The LFM (*top*) and BAOAB (*bottom*) integrators are initialized from a biased start and iterated for 10,000 function evaluations.

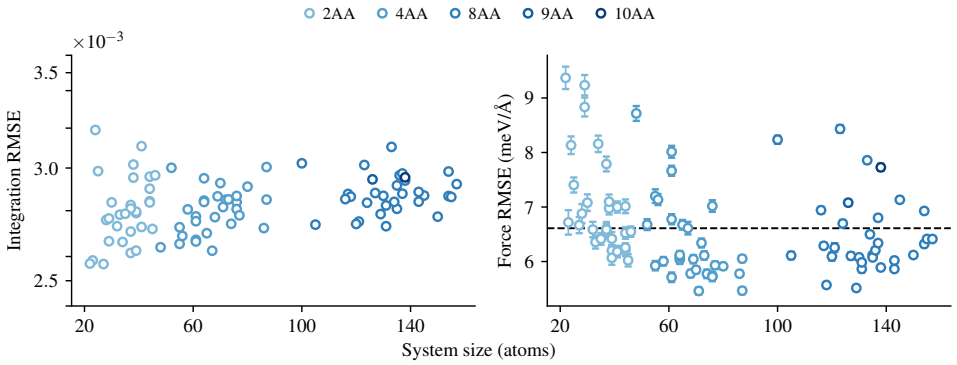

Figure 11: Integration and force errors vs. system size on 92 *Many Peptides* test systems. *Left*: integration RMSE at a 5fs timestep. *Right*: force RMSE evaluated on the $h = 0$ force network.

D.2 Transferable Dynamics

Integration and force error. In Figure 11, we show the integration RMSE and force RMSE (obtained from the $h = 0$ force network) as a function of the system size. As the system size increases, the integration RMSE is approximately constant with a slight increase for larger systems. The force RMSE is also shown to be constant with system size and the mean force RMSE is 6.61meV/Å.

State occupancy. As was shown in Table 3, the meta-stable state occupancy obtained by the large-timestep LFM is improved in comparison to MD at fixed function evaluations. This data is visualized in Figure 12, where the convergence is shown in terms of the number of function evaluations taken during integration. At matched function evaluations, the state occupancy of the LFM is significantly improved over MD.

State discovery. In Figure 13, we show the meta-stable mode discovered by the LFM for the unseen octapeptide ANKSMIEA. This mode is not present in the reference *Many Peptides* test set obtained from long timescale MD. To verify this state is stable, we initialize validation MD trajectories from LFM samples in this state and integrate using the Amber14 force-field. This validation demonstrates that this state is indeed stable, with survival times of 11.5, 6.8 and 15.8ns for each trajectory, respectively.

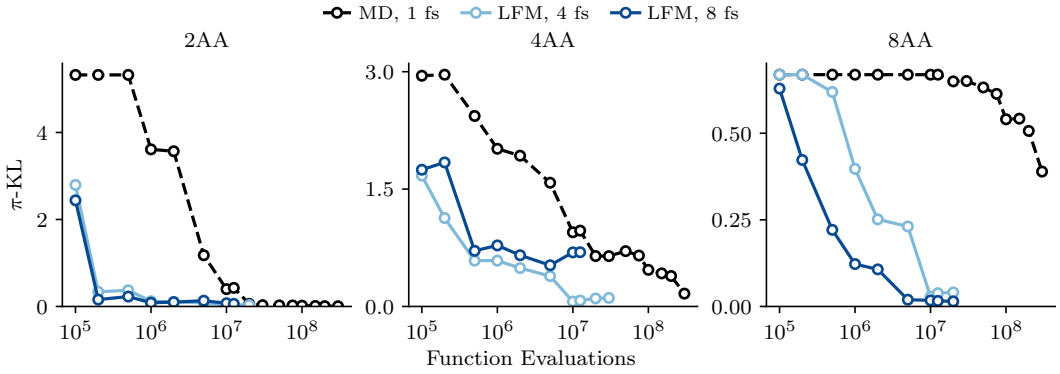

Figure 12: Meta-stable state occupancy Kullback-Leibler (KL) divergence to reference dataset occupancy, π-KL, in terms of the number of function evaluations.

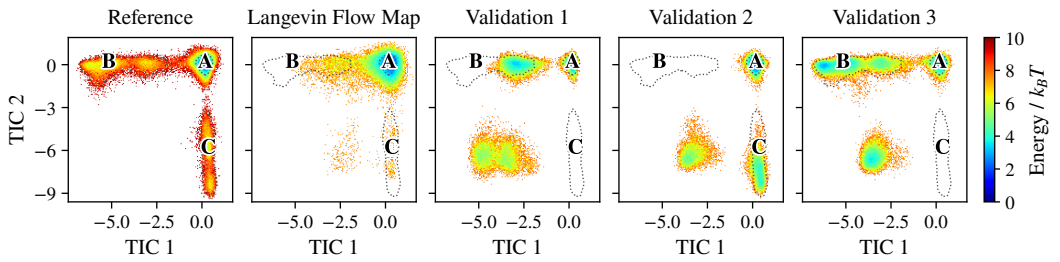

Figure 13: Meta-stable mode not present in reference MD, yet visited by the transferable LFM. Validation MD over three repeats seeded in this state found this mode to be stable for 11.5, 6.8 and 15.8ns respectively.

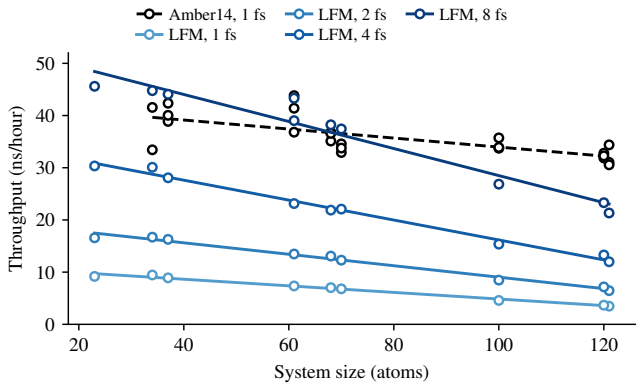

Figure 14: Simulation throughput on 1x RTX PRO 6000 Blackwell. OpenMM Amber14 OBC1 at 1fs (black, dashed), LFM at 1–8fs (blues).

Integration throughput. In Figure 14, we calculate the integration throughput for the transferable LFM and OpenMM Amber14 force-field. At matched timesteps the Amber14 force-field is significantly faster than the LFM, as expected. However, at 8fs timesteps the LFM obtains comparable simulation throughput to Amber14. This indicates the possibility to obtain MLFFs with comparable or faster integration speeds than classical MD.

E Extended Related Work

Boltzmann Generators. While this current work focuses on learning the dynamics of molecular systems, Boltzmann Generators (BGs) instead attempt to predict equilibrium molecular structures directly (Noé et al., 2019). BGs learn a generative model $p_\theta(x)$ that acts as an equilibrium proposal distribution. The samples obtained from this distribution can then be reweighted to the target equilibrium distribution with self-normalized importance sampling. The choice of architecture for $p_\theta(x)$ is an active area of research, focusing on normalizing flows (Klein et al., 2023b; Klein & Noé, 2024; Tan et al., 2025b; Rehman et al., 2026a) and recently autoregressive architectures (Rehman et al., 2026b).

Neural SDEs. Several works have looked at learning the vector fields of stochastic differential equations from a dataset of reference trajectories with applications to timeseries. These methods differ in their formulation of the objective function; for example via variational inference (Li et al., 2020), generative adversarial networks (Kidger et al., 2021), signature kernel scores (Issa et al., 2023) and maximum likelihood (Dietrich et al., 2023). These methods all require reference trajectories during training.

Machine-learned force fields. In molecular dynamics, the ground truth Langevin dynamics vector fields can be queried directly and therefore neither trajectories or training integration is required. The force-field can be regressed against directly, leading to the literature on machine-learned force-fields (Behler & Parrinello, 2007; Rupp et al., 2012; Chmiela et al., 2018; Noé et al., 2020; Keith et al., 2021; Unke et al., 2021b).

Stochastic Flow Maps. In the application of generative modelling, recent work has proposed methods to learn the solution map of a neural differential equation directly. These methods were first developed for deterministic ODE models (Boffi et al., 2025b; Frans et al., 2025; Geng et al., 2025; Boffi et al., 2025a; Ripken et al., 2026). More recently, these methods have been extended to obtain weak (distributional) solutions to stochastic dynamics (Kiyohara et al., 2025; Potaptchik et al., 2026; Holderrieth et al., 2026) but require explicit trajectory supervision, and strong (pathwise) solutions (McCallum et al., 2026) which can be trained without ordered trajectory data. In the case of diffusion models, the trajectory supervision required by weak formulations can be obtained without integration by querying the forward diffusion process or interpolant.

Transition Path Sampling. Recent literature has developed methods for obtaining the transition pathways of a system more efficiently via reinforcement learning (Das et al., 2021; Rose et al., 2021), differentiable simulation (Sipka et al., 2023), Doob's h-transform learning (Singh & Limmer, 2023; Du et al., 2024) and Onsager-Machlup action minimization (Raja et al., 2025). These approaches require stochastic sampling procedures and therefore can be paired with Langevin Flow Maps to obtain more efficient algorithms. We hope future work may explore this combination of algorithms.

F Limitations & future work

This work introduces an objective to learn the stochastic solution map of Langevin Dynamics. The resulting model class obtained by optimizing this objective, called Langevin Flow Maps, can be used to run large-timestep molecular dynamics and efficiently sample transition pathways. The current limitations of this approach and directions for future work are discussed below.

Parameterization. The current Euler-Maruyama-like update step aims to not constrain the architecture to any specific small-timestep solver or scheme. This allows the network a large amount of freedom during optimization. However, this parameterization does not constrain certain symmetries

of the update rule such as equipartition or reversibility. It may be advantageous to introduce these constraints into the update rule; this line of argument can be explored in future work.

Architecture. The current model architecture uses a translation invariant transformer and direct force prediction. To ensure translation invariance, the positional embeddings are computed as pairwise distances, r_{ij}. This embedding can become expensive at large system sizes since the input computation scales as $O(N^2)$, for an N-atom system. The MLFF literature has approached this cost by separating local and global computation (Unke et al., 2021a).

The direct force prediction head provided improved performance in early experiments over a conservative energy prediction architecture. However, it is possible to parameterize a hybrid prediction head with $f^{\theta}_{s,t}(x_s, v_s, I^N_{s,t}) = -\nabla U^{\theta}(x_s) + (t-s)\bar{f}^{\theta}_{s,t}(x_s, v_s, I^N_{s,t})$. This retains conservative forces at $s = t$ but allows the network the freedom of direct prediction for $t > s$. These higher-order flow map architectures could be explored in future work.

Transferability. The transferable LFM model demonstrated that it is possible to learn a general integrator for peptide systems that can predict accurate dynamics and transition pathways for unseen peptides. However, the model's predicted Boltzmann ensemble for unseen systems was shown to be biased at large timesteps as measured by the energy and dihedral angles, despite accurate force field prediction as $h \to 0$. This suggests that future work should focus on improving the transferable integration performance such that, as suggested by throughput analysis, the model can obtain a speedup over classical molecular dynamics simulation.