ACCELERATING FOKKER–PLANCK REGULARIZATION FOR FORCE FIELD LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models can learn molecular equilibrium distributions without force labels, but they often predict inaccurate forces that cause subsequent Langevin simulations to diverge. While standard Fokker–Planck (FP) regularization mitigates this inconsistency by constraining the density evolution, FP regularization heavily relies on expensive Hessian computations, which makes training computationally prohibitive. To address this, we introduce a novel weak FP regularization that achieves both consistency and computational efficiency. Rather than relying on expensive higher-order derivatives, our method approximates standard FP using a small set of efficiently-computable test functions. To give a correctness guarantee, we then derive an unbiased U-statistic estimator for minibatched moments. Through extensive experiments on real-world data using augmented as well as SE(3)-equivariant graph transformers, we demonstrate that our objective achieves the accuracy of the state-of-the-art FP regularization at a fraction of the memory and runtime.

1 INTRODUCTION

Molecular dynamics (MD) is an important tool in a wide range of areas such as computational biology, material sciences, and drug discovery. There, force fields are needed to produce the correct equilibrium ensemble and stable trajectories in simulations. Classical force fields trade physical accuracy for speed, whereas precise calculations for large systems are prohibitively expensive to compute. Learning to predict forces can accelerate these evaluations (Noid, 2013). Therefore, the development of the next generation of molecular dynamics force-fields is increasingly done with machine learning and deep learning methods presently. Learned coarse-grained models reduce this cost further by operating in a lower-dimensional representation (Husic et al., 2020).

One approach leverages generative AI for molecular simulations. For example, score-based diffusion models learn an equilibrium distribution directly from configurations alone (Song et al., 2021), and its endpoint score is proportional to the physical force (if the training data follows a Boltzmann distribution). Thus, denoising score matching (DSM) can both yield samples from a learned Boltzmann distribution and provide forces to a Langevin simulation *without force labels*. In practice, however, DSM is highly unstable and ill-conditioned at endpoints—exactly where it yields forces used for MD simulations in turn leading to simulations with wrong dynamics (Arts et al., 2023a; Plainer et al., 2025).

Fokker–Planck (FP) regularization alleviates this problem by favoring energies that consistently evolve with the forward diffusion process. Recent work on FP regularization has improved simulation consistency, but they evaluate score divergences or equivalent higher-order derivatives (Lai et al., 2023; Plainer et al., 2025). These derivatives make each training step 5-7x slower than DSM and raise peak memory 6-7x, which limits batch size, model size and system size.

We address this using a weak FP formulation that replaces higher-order derivatives with an estimator. In a nutshell, starting from the same energy-PDE as strong FP regularization, we preserve FP's benefits without paying the computational cost for higher-order derivatives. For this, we introduce an unbiased statistical estimator that only evaluates a few residual *moments*, making the regularizer significantly more efficient. Our result accelerates FP regularization while keeping the training objective consistent with the population counterpart.

In sum, our main contributions are:

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.

1. We introduce two weak FP objectives that replace the costly higher-order derivatives with a few test function evaluations. Galerkin FP constrains the residual across diffusion times, and Stein FP constrains only the endpoint force.
2. We derive an unbiased estimator of the FP objective.
3. We evaluate forces, simulation robustness, and computational cost using SE(3) and augmented graph transformers on benchmark molecular datasets and analytical forces. Our code will be provided at `anonymous.4open.science/r/DiffusionMD-DB8D`

2 BACKGROUND

Our goal is to learn molecular dynamics using score-based diffusion models from equilibrium configurations without requiring energy or force labels. However, diffusion models learn scores across different noise levels and diffusion times, whereas molecular dynamics only uses the score at the single and most difficult time $t = 0$. Suppose we observe equilibrium configurations $x_0 \sim p_0$ in $\mathbb{R}^D$ but do not know the potential that generated them. Although challenging, we could, in principle, estimate the density p_0 itself, but molecular dynamics actually only needs its logarithmic gradient, the *score*, $\nabla_x \log p_0(x)$. For a Boltzmann distribution, this score is proportional to the molecular force, and taking the gradient removes the unknown partition function Z, thus making learning tractable (Plainer et al., 2025). To learn tractably, score-based diffusion perturbs data over a continuum of noise levels, usually proportional to the diffusion time $t \in [0, 1]$. The forward diffusion process computes each equilibrium configuration according to

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t, \tag{1}$$

with marginal density p_t. In practice, we resort to a variance-preserving (VP) process where transitions have the form $x_t = \mu_t(x_0) + \sigma_t\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$.

To sample, we need a backward diffusion process, for which we need the score $s_p(x, t) = \nabla_x \log p_t(x)$ (Song et al., 2021). Because molecular simulation requires a conservative force field, we parameterize the learned score as the gradient of an energy $s_\theta(x, t) = -\nabla_x E_\theta(x, t)$, and learn it by denoising score matching (DSM)

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\Big[\lambda(t)\, \|s_\theta(x_t, t) + \epsilon/\sigma_t\|_2^2\Big] \ . \tag{2}$$

We know that a molecular system with potential energy $U(x)$ at temperature T has the equilibrium

$$p_0(x) = Z^{-1} \exp[-U(x)/(k_B T)] \tag{3}$$

density (Boltzmann, 1868) whose logarithmic gradient yields a score

$$F(x) = -\nabla_x U(x) = k_B T\, s_p(x, 0) \ . \tag{4}$$

We see that the equilibrium score contains exactly the information required to recover the physical force. That means, a diffusion model which learns $s_p(x, 0)$ from configurations can provide a force field without observing energies or forces during training. While standard reverse diffusion is defined across all noise levels, simulations only require $s_\theta(x, 0)$ at time $t = 0$, as overdamped Langevin

$$dX_\tau = k_B T\, s_\theta(X_\tau, 0)\, d\tau + \sqrt{2k_B T}\, dW_\tau \ , \tag{5}$$

where τ denotes physical simulation time. However, the DSM target ϵ/σ_t has variance $\propto \sigma_t^{-2}$, which explodes as $t \to 0$, exactly where simulation reads the force. Since simulation integrates these forces over long trajectories, even small endpoint errors accumulate. To address this issue, recent work has demonstrated that this inconsistent behavior can be mitigated through enforcing Fokker–Planck consistent evolutions, as detailed next.

3 FOKKER–PLANCK EQUATION

Because DSM fits the score at each noise level separately, it is not forced to learn scores that evolve under the forward process. The Fokker–Planck (FP) (Øksendal, 2003) equation is exactly this missing coupling, as it describes the time-evolution of the probability distribution of a stochastic process

$$\partial_t p_t = -\nabla\cdot(f p_t) + \tfrac{1}{2} g^2(t) \Delta p_t \ . \tag{6}$$

and 'couples' the density at every diffusion time to the known forward dynamics in Eq. 1. If E_θ represents the evolving negative log density $-\log p_t$, substituting $p_t \propto e^{-E_\theta}$ must satisfy this equation. We measure its violation with the strong residual

$$r_\theta = \partial_t E_\theta - \nabla\cdot f + f\cdot\nabla E_\theta - \tfrac{1}{2}g^2\Delta E_\theta + \tfrac{1}{2}g^2\left\|\nabla E_\theta\right\|_2^2 \ . \tag{7}$$

A strong FP penalty minimizes $\mathbb{E}_{p_t}[r_\theta^2]$ and thereby encourages one energy model to evolve consistently across diffusion times. As desirable as this may be, a strong FP penalty is computationally expensive, as the *exact* term $\Delta E_\theta = \operatorname{tr}\nabla^2 E_\theta$ relies on D Hessian–vector products. Next, we describe the existing pointwise estimators, followed by our weak formulation.

3.1 Fokker–Planck Estimator

To avoid the computational bottleneck of evaluating FP, we first replace this trace with a Hutchinson estimate $z^\top\nabla^2 E_\theta\, z$ (Hutchinson, 1989) for Rademacher or Gaussian probes z. Then, we can replace the expensive higher-order differentials with a finite difference of the score along z (Plainer et al., 2025). However, simply adding this estimate as penalty to the loss will add the estimator's variance to our objective which may degrade training performance. To cancel the variance, we multiply the residuals from two independent probes which yields an unbiased penalty term (Plainer et al., 2025).

We consider two such estimators, which differ only in *where* they evaluate the residual. `Smoothed FP`, the published regularizer of Plainer et al. (2025) (released as `ScoreMD`), evaluates the remaining terms at a randomly shifted point $x_t + \sigma z$, thus targeting a residual that is spatially averaged around x_t at the cost of exactness at x_t. `Pointwise FP` evaluates every term at the sample x_t and thus estimates $\mathbb{E}_{p_t}[r_\theta^2]$ itself, i.e., exactly the objective our weak formulation approximates; both coincide as $\sigma \to 0$. Either way, they must differentiate E_θ twice in x and once more in θ, and every probe enlarges this compute graph, which dominates training cost. To alleviate this, we introduce our weak FP approximation, `Galerkin FP`, which avoids second derivatives of E_θ entirely.

4 Weak Galerkin Fokker–Planck

The FP equation *strongly* requires that the residuals $r_\theta = 0$ are zero everywhere. When FP holds, then the residual must vanish against every function ϕ

$$\mathbb{E}_{p_t}[\phi(x) r_\theta(x,t)] = 0 \ , \tag{8}$$

which is the *weak* condition. Because this expectation is expressed as an integral, we can apply integration by parts and move the derivative from E_θ onto ϕ as in Eq. 13. By choosing test functions ϕ with *analytic gradients*, we avoid second derivatives of E_θ, thus avoiding the computational bottleneck. While testing against every admissible function recovers the strong equation, we would lose computational benefits. Instead, we choose a finite-dimensional test space and enforce a few violations within it.

For a fixed diffusion time t and a finite test function space $\mathcal{V}_M = \operatorname{span}\{\phi_1,\dots,\phi_M\} \subset L^2(p_t)$, we define the residual moments and Gram matrix

$$m_j(\theta,t) = \mathbb{E}_{p_t}[\phi_j r_\theta], \qquad G_{ij}(t) = \mathbb{E}_{p_t}[\phi_i\phi_j] \ . \tag{9}$$

while the vector m encodes FP violations detected by our functions. As a simple squared $\|m\|_2^2$ term is susceptible to rescaling of the (arbitrary) functions and overvalues potentially redundant basis directions, employing it as penalty is challenging. We therefore correct this by introducing a Gram matrix G as weight, which encodes the geometry of the test space under p_t, leading to the Galerkin loss (Brenner & Scott, 2008), denoting the Moore–Penrose inverse as †,

$$\mathcal{L}_{\mathrm{G,strong}}(\theta,t) = m(\theta,t)^\top G(t)^\dagger m(\theta,t) \ . \tag{10}$$

Proposition 1 (Projection identity (Brenner & Scott, 2008))**.** *Assume $r_\theta \in L^2(p_t)$ and $\phi_j \in L^2(p_t)$. Let $P_{\mathcal{V}_M}$ be the $L^2(p_t)$ orthogonal projection onto $\mathcal{V}_M$. Then*

$$m^\top G^\dagger m = \|P_{\mathcal{V}_M} r_\theta\|^2_{L^2(p_t)} \le \|r_\theta\|^2_{L^2(p_t)} \, . \tag{11}$$

If a nested sequence of test spaces is dense in $L^2(p_t)$, the left-hand side converges monotonically to the strong residual norm.

In other words, while Eq. 11 controls the strong FP, it is blind when violations are not expressible by these functions, e.g., when they are orthogonal to the test space. Still, directly evaluating m_j involves the residual r that requires the costly ΔE_θ computation in Eq. 7. To mitigate this cost, we instead resort to integrate this term by parts. Assume that $p_t \phi_j \nabla E_\theta$ vanishes at the boundary (or decays sufficiently fast on $\mathbb{R}^D$). Since $s_\theta = -\nabla E_\theta$,

$$\mathbb{E}_{p_t}[\phi_j \Delta E_\theta] = -\mathbb{E}_{p_t}[\nabla \phi_j \cdot \nabla E_\theta] - \mathbb{E}_{p_t}[\phi_j \nabla \log p_t \cdot \nabla E_\theta] \tag{12}$$

$$= \mathbb{E}_{p_t}[\nabla \phi_j \cdot s_\theta] + \mathbb{E}_{p_t}[\phi_j s_p \cdot s_\theta] \, . \tag{13}$$

The denoising identity replaces the unknown s_p in expectation

$$\mathbb{E}_{x_t}[\phi_j s_p \cdot s_\theta] = \mathbb{E}_{x_0,\epsilon}[\phi_j(-\epsilon/\sigma_t) \cdot s_\theta] \, . \tag{14}$$

Substituting Eqs. 13–14 into the moment of Eq. 7 yields an estimator that uses the energy, its time derivative, its score, and analytic derivatives of ϕ_j, but no coordinate Hessian of the model (we reduce its variance with antithetic noise pairs, App. B). Note that Eq. 14 inherits the ϵ/σ_t target, so we sample $t \geq t_{\min}$ for Galerkin FP; Stein FP covers the endpoint. Weak formulations often estimate m by a sample mean and optimize its squared norm, thereby accidentally changing the objective at finite batch by introducing a minibatch bias.

Proposition 2 (Minibatch bias). *Let $a_1, \ldots, a_n \in \mathbb{R}^M$ be independent cluster moments conditional on t and a fixed positive semidefinite matrix A. If $\mathbb{E}[a_i] = m$ and $\mathrm{Cov}(a_i) = \Sigma$, then*

$$\mathbb{E}\big[\bar{a}^\top A \bar{a}\big] = m^\top A m + \frac{1}{n} \operatorname{tr}(A\Sigma) \, , \tag{15}$$

The extra term depends on θ, so its gradient moves even an exact model away from zero residual. We therefore remove the diagonal products and use the *U-statistic* (Hoeffding, 1948)

$$\widehat{\mathcal{L}}_U = \frac{1}{n(n-1)} \sum_{i \neq k} a_i^\top A a_k \, , \tag{16}$$

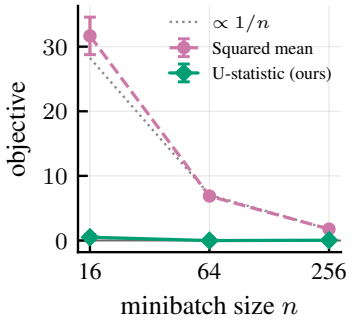


Figure 1: Our loss is unbiased.

with metric $A = (\widehat{G} + \rho I)^{-1}$ estimated from *independent* samples, and by conditioning on them, Eq. 16 becomes unbiased for $m^\top A m$. This is a property of an unbiased stochastic estimator, not a failure of the population quadratic. In Fig. 1 we see how gradient descent moves an *exact* model away from the solution, while the U-statistic is essentially zero.

Simulations infer forces $s_p(\cdot, 0)$ exactly when the target ϵ/σ_t is undefined. Here we use *Stein's method* (Gorham & Mackey, 2015) $\mathbb{E}_{p_0}[\nabla \phi^\top s_p(\cdot, 0) + \Delta \phi] = 0$ we replace this target for a Gram matrix $H_{ij} = \mathbb{E}_{p_0}[\nabla \phi_i^\top \nabla \phi_j]$, obtaining

$$c_j(\theta) = \mathbb{E}_{p_0}\left[\nabla \phi_j^\top s_\theta(x, 0) + \Delta \phi_j\right] \, . \tag{17}$$

Proposition 3. *Let $\mathcal{W}_M = \mathrm{span}\{\nabla \phi_1, \ldots, \nabla \phi_M\}$ in $L^2(p_0; \mathbb{R}^D)$ and $e_\theta = s_\theta(\cdot, 0) - s_p(\cdot, 0)$. Under the same integrability assumptions,*

$$\mathcal{L}_{\mathrm{Stein}}(\theta) = c^\top H^\dagger c = \| P_{\mathcal{W}_M} e_\theta \|^2_{L^2(p_0)} \, . \tag{18}$$

By Eq. 4, Eq. 18 is a projected force error up to a factor $(k_B T)^2$. Moreover, $k_B T(s_\theta^\top \nabla \phi + \Delta \phi)$ is the generator of Eq. 5, so $c = 0$ is the weak *stationary* FP equation of the simulation itself: the data distribution is invariant under the dynamics we run, tested on $\mathrm{span}\{\phi_j\}$. Both derivatives act on the analytic ϕ_j, so Stein FP needs one score evaluation on clean data and neither $\partial_t E_\theta$ nor ΔE_θ. In contrast to Galerkin FP, which ties the energy across diffusion times, Stein FP constrains only the endpoint force.

Last, both Galerkin and Stein FP require a concrete choice of test functions. As each moment only detects violations it can express, we want tests that are *local*, i.e., resolve where the residual is violated, *smooth*, i.e., have analytic gradients and Laplacians, and respect the molecule's symmetries. We therefore resort to radial basis functions on selected pair distances $q(x)$

$$\phi_j(x) = \exp\left[-\frac{\|q(x) - c_j\|_2^2}{2h_j^2}\right] , \tag{19}$$

where we fix centers c_j and bandwidths h_j from independent samples (App. B). For Galerkin FP, we add the constant test $\phi_0 \equiv 1$, which captures spatially uniform violations (its gradient vanishes, so Stein FP omits it). As pair distances are invariant to translation and rotation, so are the tests, while their gradients transform equivariantly. Finally, we remove center-of-mass motion before computing all losses, as a translation-invariant energy exerts no force along it. Thereby, evaluating all M moments costs only analytic test-function evaluations on top of one score evaluation, independent of the model's size. In sum, we train with the `Galerkin FP` loss

$$\min_{\theta} \; \mathcal{L}_{\mathrm{DSM}}(\theta) + \alpha \, \mathbb{E}_{t\sim\pi_{\mathrm{FP}}}[\widehat{\mathcal{L}}_{\mathrm{G}}(\theta, t)] \; , \tag{20}$$

where $\widehat{\mathcal{L}}_{\mathrm{G}}$ is the U-statistic Eq. 16 of Eq. 10, and t is drawn uniformly on $[t_{\min}, t_{\max}]$, once per minibatch. For `Stein FP`, we replace the time average by $\mathcal{L}_{\mathrm{Stein}}$ in Eq. 18.

5 Related Work

Chmiela et al. (2017) used gradient-domain machine learning and learning from MD trajectories to recover potential energy surfaces. Frank et al. (2023) and Batatia et al. (2023) use a message-passing equivariant GNN to learn forces from distances and atom features. Recently foundational force-fields have emerged, like (Wood et al., 2026), trained on 500 million atomic structures in an effort to obtain universal force-fields for a wide range of applications.

Recently, diffusion models have emerged as another paradigm for force-field generation. (Arts et al., 2023b) used diffusion models for coarse-grained force fields in MD simulations for both sampling from the equilibrium distribution as well as using the learned force field for MD simulations. (Wu & Li, 2023) also use score-matching to predict molecular conformations based on perturbations of molecular priors. (Noé et al., 2019) and (Schebek et al., 2024) use Boltzmann generators to sample equilibrium distributions, while (Köhler et al., 2020), (Abdin & Kim, 2024) and (Midgley et al., 2023) use normalizing flows for the same task. (Wirnsberger et al., 2020) learn free energy estimates by treating free energy perturbation as a machine learning problem. Other methods use force-matching to generate force fields, like (Zheng et al., 2024) which requires explicit force labels. A recent model (Lewis et al., 2025) uses AlphaFold sequence generation as input for a denoising diffusion model to produce equilibrium distributions. However, the equilibrium distribution of a system is not the full picture. Many properties are only accessible through simulation of the systems dynamics via MD.

While, in principle, learned forces should follow the Fokker–Planck evolution, Plainer et al. (2025) found that they are often in practice inconsistent, especially when prediction time approaches zero. To ensure FP consistency, they introduced a FP regularizer at a considerable hit to the learning efficiency, often leading to $5-7\times$ times slower per-step performance—even after employing Hutchinson's trick. We seek to narrow this gap significantly, without degrading force quality.

Furthermore, FP regularization has been identified as a useful approach to improve diffusion processes outside molecular simulations. For instance in *image generation*, Niemann et al. (2026) have demonstrated that lightweight pointwise penalties, e.g., on the score norm or its divergence, recover much of the benefit of FP regularization at a fraction of its cost. Closer to our work, the locally smoothed estimator of Plainer et al. (2025) averages the residual over a small Gaussian neighborhood of each sample, which can be read as a local weak test; yet, it still evaluates ΔE_θ, whereas we move all derivatives onto fixed test functions.

To the best of our knowledge, this is the first contribution using Galerkin theory in combination with diffusion models for molecular simulations.

6 Experiments

Having introduced weak FP regularization, we now test what a projected residual can and cannot replace. Specifically, we ask three questions.

- **RQ1** Does enforcing the FP equation on a finite test space retain the simulation quality of full-residual FP regularization?
- **RQ2** How much compute does avoiding second derivatives save, and does this saving improve accuracy at equal compute?

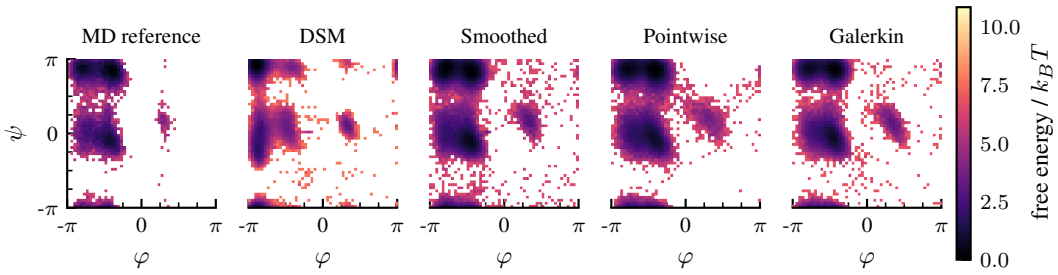


Figure 2: FP objectives reproduce the free-energy landscape, whereas DSM distorts it. We show the free energy over (φ, ψ) $[k_BT]$ from Langevin simulations with the $t = 0$ score (alanine dipeptide, from scratch, SE(3)).

RQ3 When does simulation need consistency across diffusion times, as enforced by Galerkin FP, and when does a correct endpoint force, as enforced by Stein FP, suffice?

Data. To answer these questions, we study three coarse-grained systems, *alanine dipeptide* (5 backbone beads) (Köhler et al., 2021), *Lys–Ser (KS) dipeptide* (10 beads) (Klein et al., 2023), and *Chignolin* (10 C_α) (Lindorff-Larsen et al., 2011), as well as the all-atom *MD22* Ac-Ala_3-NHMe (42 atoms) (Chmiela et al., 2023). Thereby, we cover increasing system size, from two slow dihedrals to a folding mini-protein and an all-atom peptide.

Models and baselines. Throughout, we employ an SE(3)-equivariant many-body graph transformer with chiral channels, which builds on graph transformers (Shi et al., 2021) and equivariant vector and tensor message passing (Schütt et al., 2021; Simeon & de Fabritiis, 2023) (App. B); App. C repeats all comparisons with a graph transformer that learns rotations by augmentation (AugGT) (Abramson et al., 2024). We compare our Galerkin FP and Stein FP against three kinds of competitors. First, *DSM* without regularization shows what FP regularization must improve on. Second, the full-residual estimators control the whole residual (Sec. 4): *Smoothed FP* (Plainer et al., 2025) is the published state of the art, and *Pointwise FP* is the exact objective Galerkin FP approximates, i.e., the quality our weak objectives should retain. For *Smoothed FP* we averaged over diffusion times to obtain their results. Third, *WG-RFF* tests whether any weak formulation suffices: it tests the probability-flow equation integrated over a time window against random Fourier features, and penalizes the squared minibatch moments without a Gram metric. In contrast to Galerkin FP, it is thus neither local in time, nor geometry-aware, nor unbiased (Sec. 4).

Metrics and training. As molecular dynamics reads only the endpoint force, we evaluate every model through *Langevin simulations*: we use the learned score at exactly $t = 0$ to obtain forces, and integrate Eq. 5 with $dt = 10^{-4}$ for 30k steps, starting 1024 chains from both reference and model states. To test whether simulations reproduce the free-energy landscape, we measure the *PMF error* (Durumeric et al., 2024) between simulated and reference distributions of the slow coordinates, i.e., the (φ, ψ) angles (dipeptides) or the two slowest TICA coordinates (Pérez-Hernández et al., 2013) (Chignolin, MD22). Additionally, we report Jensen-Shannon (JS) divergence in Apx. C. As these coordinates are blind to local distortions, we additionally measure *stability* as the fraction of simulated states in which any bond deviates by more than 0.5 Å from its reference length (Fu et al., 2023). To separate shaping a landscape from refining one (RQ3), we train either *from scratch* for 30k steps, or *continue* a shared 20k-step DSM checkpoint for 10k regularized steps. We ran all experiments on a single NVIDIA H200, report mean and standard error over 3 seeds, and list all hyperparameters in Apx. B.

6.1 RQ1: Simulation quality

If a finite projection suffices, the Galerkin FP should match the full-residual estimators. Reporting simulation quality from scratch in Tbl. 1 (top), we see that Galerkin FP matches both full-residual estimators within two standard errors on every system, while DSM and WG-RFF remain far from the reference (e.g., $6.55 \to 2.40$ on alanine dipeptide and $7.68 \to 3.58$ on KS dipeptide). Crucially, this

Table 1: Galerkin FP matches the full-residual estimators from scratch, at $2.5 - 3.5$x lower cost per step, while Stein FP is best at refining a DSM model for Chignolin. We show the PMF error between simulated and reference distributions and continuation (SE(3)), grouped into no FP, full-residual FP, and ours, as mean and standard error over three seeds; cost is the approximate time per training step relative to DSM.

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe	cost
SE(3)					
DSM	6.55 ± 1.22	7.68 ± 0.42	5.76 ± 0.58	8.93 ± 0.70	∼1×
WG-RFF	11.93 ± 4.37	10.59 ± 0.68	5.45 ± 0.43	9.83 ± 0.46	∼2×
Smoothed FP	2.75 ± 0.14	3.59 ± 0.16	3.69 ± 0.12	**6.29** ± 0.27	∼6×
Pointwise FP	2.48 ± 0.10	3.63 ± 0.27	**3.69** ± 0.07	6.52 ± 0.12	∼6×
Galerkin FP (ours)	**2.40** ± 0.13	**3.58** ± 0.14	3.73 ± 0.03	6.57 ± 0.41	∼2×
AugGT					
DSM	9.72 ± 0.55	16.07 ± 2.45	5.76 ± 0.40	9.77 ± 0.43	∼1×
WG-RFF	9.65 ± 0.23	9.17 ± 1.39	5.57 ± 0.18	9.42 ± 0.69	∼2×
Smoothed FP	10.52 ± 0.27	9.88 ± 1.34	**4.72** ± 0.02	8.42 ± 1.32	∼5×
Pointwise FP	10.02 ± 0.20	10.57 ± 0.93	4.79 ± 0.05	**6.77** ± 0.29	∼5×
Galerkin FP (ours)	**9.51** ± 0.10	**9.06** ± 0.36	4.96 ± 0.22	8.14 ± 2.03	∼2×
Continuation (20k DSM + 10k steps)					
DSM	5.20 ± 0.67	8.22 ± 2.05	5.52 ± 0.34	9.60 ± 0.15	∼1×
WG-RFF	5.17 ± 0.71	7.81 ± 1.99	5.49 ± 0.28	9.60 ± 0.46	∼2×
Smoothed FP	**2.96** ± 0.09	5.07 ± 0.32	5.25 ± 0.24	9.14 ± 0.51	∼6×
Pointwise FP	3.20 ± 0.31	5.24 ± 0.08	5.18 ± 0.33	8.83 ± 0.52	∼6×
Galerkin FP (ours)	3.98 ± 0.71	**4.65** ± 0.34	5.50 ± 0.35	**8.64** ± 0.55	∼2×
Stein FP (ours)	3.05 ± 0.08	5.70 ± 0.45	**5.11** ± 0.19	9.64 ± 0.46	∼2×

holds, although Galerkin FP only controls the projection of the residual onto 64–128 test functions (Eq. 11). Thus, for these systems, the residual directions that matter for simulation lie largely within a small, smooth test space. However, not every weak formulation finds them: WG-RFF is on par with DSM, although it also avoids second derivatives. As swapping our unbiased estimator for WG-RFF's squared minibatch moments leaves Galerkin FP's simulation quality unchanged (Tbl. 7), the gap stems from testing locally in time with a geometry-aware metric. What matters, therefore, is how the residual is tested, not only that it is tested weakly. Interestingly, the performance of all regularizers is worse with AugGT, indicating that sophisticated regularization schemes cannot mitigate errors from GT approximations. Stein FP shows the best performance when using AugGT and the continuation scheme, see Tbl. 9 reporting JS divergence. In the Ramachandran plots of Fig. 2, we see how DSM smears and misplaces basins, whereas all FP estimators track the reference closely.

Reporting stability in Fig. 3(a,b), we see that on SE(3) all FP estimators keep the fraction of states with broken bonds within a few percent, on par with DSM, while Stein FP trained from scratch breaks up to every bond (RQ3). Along the folding coordinate of Chignolin (Fig. 3d), DSM misplaces the free-energy minimum, whereas Smoothed and Galerkin FP stay closer to the reference profile. With AugGT, the stability degrades for all methods. Galerkin FP is stable in contrast to DSM on KS dipeptide ($52\% \rightarrow 0.8\%$) and Chignolin ($16\% \rightarrow 6.5\%$). Our method achieves this, without degrading the reverse-diffusion samples quality, as demonstrated by the low denoising loss in Fig. 14 and Tbl. 8.

As molecular dynamics integrates the force over long trajectories, we finally test whether simulations stay in equilibrium once they are there. Starting chains from reference MD states, a correct force field leaves their distribution invariant, i.e., it satisfies the stationary FP equation of Eq. 5. In Fig. 4, we see that DSM drifts away from the reference over the whole simulation on every system, whereas Smoothed and Galerkin FP stay stationary. Thus, FP regularization does not merely improve the average ensemble, but keeps the simulated dynamics in equilibrium.

6.2 RQ2: Computational cost

Having ascertained simulation quality, we now study where the cost of FP regularization comes from. In Fig. 5(a,b), we see that Galerkin FP is 2.5–$3.5\times$ faster per step than Smoothed FP and needs about

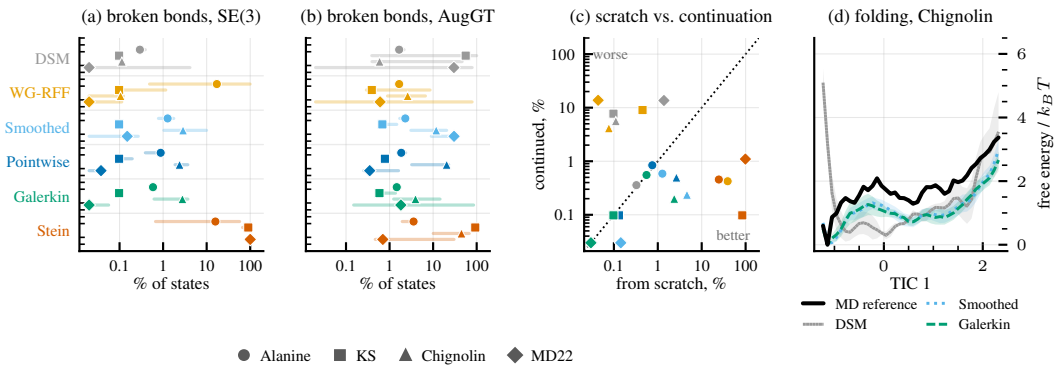


Figure 3: FP objectives keep simulations stable and close to the folding landscape; Stein FP cannot train a model but repairs one. We show (a, b) the broken-bond rate from scratch (marker: median, line: range over seeds), (c) broken bonds from scratch against continued from a DSM checkpoint (SE(3)), and (d) the free energy of Chignolin along its slowest TIC (SE(3), mean±se).

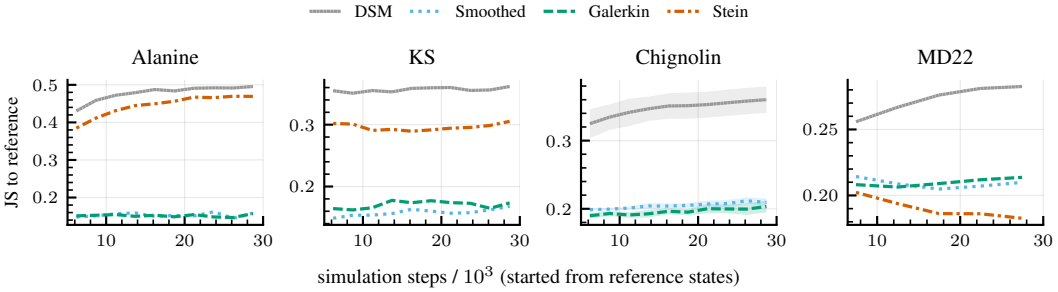


Figure 4: Started in equilibrium, FP-regularized simulations stay there, whereas DSM drifts away. We show the JS distance to the reference over simulation time for chains started from reference MD states, in windows of the retained frames after 5k burn-in steps (from scratch, SE(3), mean and standard error over three seeds).

2.3× less peak memory (Tbl. 5), while Stein FP costs $1.8 - 2.0\times$ DSM, similar to WG-RFF. As all FP estimators enforce the same equation, this cost is an artifact of the second derivatives, not of the constraint. Moreover, at matched compute (Fig. 5d), Galerkin FP after 30k steps outperforms both full-residual estimators after 10k steps, which cost about the same GPU time, on every system, e.g., 0.144 against 0.281 (Smoothed) and 0.238 (Pointwise) on alanine dipeptide.

6.3 RQ3: Diffusion-time consistency

Finally, we ask when each of our two constraints matters. As Stein FP constrains only the endpoint while Galerkin FP constrains all diffusion times, comparing them from scratch and from a DSM checkpoint reveals which constraint shapes the landscape and which refines it. From scratch, Stein FP fails: it reaches JS 0.37 (Tbl. 2) and breaks bonds in 16–100% of states (Fig. 3c). In contrast, when refining a DSM checkpoint, Stein FP gives the lowest JS (sim) on three of four systems (e.g., 0.237 vs. 0.307 for Pointwise FP on alanine dipeptide) and keeps broken bonds below 1%. Galerkin FP shows the opposite pattern: it is on par with the pointwise estimators from scratch, but no better than them under continuation. We interpret this as follows: an endpoint constraint alone does not shape the energy landscape, as nothing ties the $t = 0$ score to the data away from the samples, but it efficiently corrects a landscape that DSM has already shaped. Consistency across diffusion times, in contrast, is what lets FP regularization shape the landscape from scratch. Note that no continuation reaches the from-scratch quality; as continuation also uses fewer regularized steps (10k vs. 30k), we cannot attribute this gap to the initialization alone. In sum, we recommend Galerkin FP for training from scratch. For refining a DSM model, Stein FP performed best at our default coefficients. Galerkin

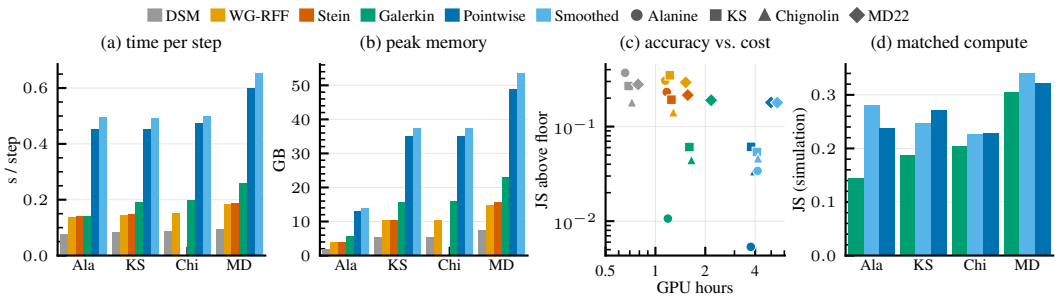


Figure 5: Our FP regularization is cheaper without losing accuracy. We show (a) the time per step, (b) the peak memory, (c) the JS distance above each system's reference floor against the GPU hours of 30k steps, and (d) the JS distance at matched compute, i.e., Galerkin FP after 30k steps against the full-residual estimators after 10k steps (from scratch, SE(3)).

FP's continuation results depend strongly on its coefficients App. C, so this ranking may change with tuning.

7 Conclusion

In this work, we examined how to make diffusion-based force-field generation more efficient while retaining the accuracy of the expensive FP method. Using FP in general is necessary to obtain accurate results, as DSM alone does not capture the physics well. We omitted the expensive computation of Hessians by relying on efficiently computable test functions. By testing our method on biomolecular test data, we showed that we achieve comparable results to the state of the art FP method at a fraction of the cost. Moreover, we found that the two constraints play different roles: consistency across diffusion times (Galerkin FP) shapes the energy landscape from scratch, whereas the endpoint constraint (Stein FP) cannot train a model alone, but is the cheapest and most accurate way to refine an existing DSM model. Still, a finite test space is blind to residual directions outside its span, which can leave local curvature wrong even when basin populations are right (Fig. 12), and on the largest all-atom system, stability without equivariance remains seed-dependent. In addition, using faster GT approximations not only results in reduced computation time but also degrades the quality of the results. The best results were achieved by explicitly using SE(3) equivariant GTs. Richer, e.g., chiral or many-body, test spaces, and more seeds are natural next steps.

AI use statement

In this work, we used generative AI tools for code generation and for editing parts of the manuscript text. Specifically, to provide feedback on research methodology or experiments and assistance with benchmarking infrastructure. We have not used generative AI tools for formulating the research questions. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by all authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

App. B specifies the systems and splits, optimization, architectures, regularizer coefficients, test basis, and evaluation protocol; App. A proves Props. 1–3. Per-seed results, including diverged runs, are in App. C and Fig. 23.

References

Osama Abdin and Philip M. Kim. Direct conformational sampling from peptide energy landscapes through hypernetwork-conditioned diffusion. *Nature Machine Intelligence*, 6(7):775–786, June 2024.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024. ISSN 1476-4687.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023a.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zuegner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics, 2023b. URL https://arxiv.org/abs/2302.00600.

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields, 2023. URL https://arxiv.org/abs/2206.07697.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

L. Boltzmann. *Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten: vorgelegt in der Sitzung am 8. October 1868.* k. und k. Hof- und Staatsdr., 1868.

Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2008.

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. doi: 10.1126/sciadv.1603015. URL https://www.science.org/doi/abs/10.1126/sciadv.1603015.

Stefan Chmiela, Valentin Väsasselä, Simon Batzner, Filippo Musil, et al. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023.

Aleksander E. P. Durumeric, Yaoyi Chen, Frank Noé, and Cecilia Clementi. Learning data efficient coarse-grained molecular dynamics from forces and noise, 2024. arXiv:2407.01286.

J. Thorben Frank, Oliver T. Unke, and Klaus-Robert Müller. So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems, 2023. URL https://arxiv.org/abs/2205.14276.

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.

Jackson Gorham and Lester Mackey. Measuring sample quality with stein's method. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/698d51a19d8a121ce581499d7b701668-Paper.pdf.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

Brooke E. Husic, Nicholas E. Charron, Dominik Lemm, Jiang Wang, Adrià Pérez, Maciej Majewski, Andreas Krämer, Yaoyi Chen, Simon Olsson, Gianni de Fabritiis, Frank Noé, and Cecilia Clementi. Coarse graining molecular dynamics with graph neural networks. *The Journal of Chemical Physics*, 153(19):194101, 2020.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52863–52883. Curran Associates, Inc., 2023.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2796–2809. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/167434fa6219316417cd4160c0c5e7d2-Paper.pdf.

Chieh-Hsin Lai, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. FP-Diffusion: Improving score-based diffusion models by enforcing the underlying score fokker–planck equation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Adam E. Foster, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Roberto Sordillo, Ryota Tomioka, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761): eadv9817, 2025. doi: 10.1126/science.adv9817. URL https://www.science.org/doi/abs/10.1126/science.adv9817.

Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Energy Based Models Workshop*, 2019.

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XCTVFJwS9LJ.

Onno Niemann, Gonzalo Martínez Muñoz, and Alberto Suárez Gonzalez. An analysis of regularization and fokker-planck residuals in diffusion models for image generation, 2026. URL https://arxiv.org/abs/2604.15171.

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.

William George Noid. Perspective: Coarse-grained models for biomolecular systems. *The Journal of chemical physics*, 139(9), 2013.

Bernt Øksendal. *Stochastic Differential Equations*, pp. 65–84. Springer Berlin Heidelberg, 2003.

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.

Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noé. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models. *arXiv preprint arXiv:2506.17139*, 2025.

Maximilian Schebek, Michele Invernizzi, Frank Noé, and Jutta Rogal. Efficient mapping of phase diagrams with conditional boltzmann generators. *Machine Learning: Science and Technology*, 2024.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *CoRR*, abs/2102.03150, 2021. URL https://arxiv.org/abs/2102.03150.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021.

Guillem Simeon and Gianni de Fabritiis. Tensornet: Cartesian tensor representations for efficient learning of molecular potentials, 2023. URL https://arxiv.org/abs/2306.06482.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racaniere, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112, 2020.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. Uma: A family of universal models for atoms, 2026. URL https://arxiv.org/abs/2506.23971.

Fang Wu and Stan Z Li. Diffmd: a geometric diffusion model for molecular dynamics simulations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 5321–5329, 2023.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, He Zhang, Shidi Tang, Hongxia Hao, Peiran Jin, Chi Chen, Frank Noé, Haiguang Liu, and Tie-Yan Liu. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567, 2024. doi: 10.1038/s42256-024-00837-3. URL https://doi.org/10.1038/s42256-024-00837-3.

A Proofs

A.1 Proof of the projection identity

Let $\Phi : \mathbb{R}^M \to \mathcal{V}_M$ map coefficients c to $\sum_j c_j \phi_j$. The normal equations for projecting r_θ are $Gc = m$. The minimum-norm solution is $c = G^\dagger m$ because m lies in the range of G. Therefore

$$\|P_{\mathcal{V}_M} r_\theta\|^2_{L^2(p_t)} = c^\top G c = m^\top G^\dagger m. \tag{21}$$

As orthogonal projection is contractive, we can derive the inequality in Eq. 11, and using standard Hilbert-space approximation we obtain convergence for nested dense spaces.

A.2 Proof of the minibatch bias identity

We write $a_i = m + \delta_i$, with $\mathbb{E}[\delta_i] = 0$ and independent δ_i. Expanding the quadratic gives

$$\mathbb{E}[\bar{a}^\top A \bar{a}] = m^\top A m + \mathbb{E}\left[\left(\frac{1}{n}\sum_i \delta_i\right)^\top A \left(\frac{1}{n}\sum_k \delta_k\right)\right] \tag{22}$$

$$= m^\top A m + \frac{1}{n^2}\sum_i \mathbb{E}[\delta_i^\top A \delta_i] = m^\top A m + \frac{1}{n}\operatorname{tr}(A\Sigma)\ . \tag{23}$$

Removing the $i = k$ terms yields Eq. 16; independence makes every remaining cross product have expectation $m^\top A m$.

A.3 Proof of the Stein projection identity

Stein's identity follows from integration by parts, $\mathbb{E}_{p_0}[\Delta\phi_j] = -\mathbb{E}_{p_0}[\nabla\phi_j^\top \nabla \log p_0]$, assuming $p_0 \nabla\phi_j$ vanishes at infinity. Subtracting it from Eq. 17 gives

$$c_j = \mathbb{E}_{p_0}[\nabla\phi_j^\top (s_\theta(\cdot, 0) - s_p(\cdot, 0))]. \tag{24}$$

These are the inner products between e_θ and the vector-valued basis elements $\nabla\phi_j$; H is their Gram matrix. Applying the preceding projection argument in $L^2(p_0; \mathbb{R}^D)$ proves Eq. 18.

B Experimental Details

Systems and splits. We use a five-bead coarse graining for *alanine dipeptide* with `[ACE-C, ALA-N, ALA-CA, ALA-C, NME-N]` at 300 K. Lys–Ser (KS) dipeptide is from `ScoreMD`'s minipeptides (Plainer et al., 2025), coarse grained to `[N, CA, CB, C, O]` per residue (10 beads, 49k frames, 310 K). For Chignolin, we use one C_α bead per residue (10 beads, 350 K) (Lindorff-Larsen et al., 2011). MD22 Ac-Ala_3-NHMe is all-atom (42 atoms, 85k frames) (Chmiela et al., 2023). We split trajectories into contiguous training, validation and test block splits, so that correlated frames cannot leak across splits.

Optimization. In all runs we use a VP schedule with $\beta \in [0.1, 20]$. We use AdamW (Loshchilov & Hutter, 2019), gradient clipping at 1.0, weight decay 10^{-3}, and diffusion times sampled in $[10^{-3}, 0.999]$. For continuation, we train for 10k regularized steps at learning rate 3×10^{-5} with 250 warm-up steps from a shared 20k-step DSM checkpoint. From scratch, we train for 30k steps at 10^{-4} with 500 warm-up steps. Batch sizes are 1024 for the dipeptides and 128 for MD22; Chignolin uses 1024 from scratch with SE(3) and 512 otherwise. Every regularizer is warmed up linearly over 1000 steps.

Architectures. We use a graph transformer (Shi et al., 2021) as our neural network architecture, as it is an established neural network to represent and learn atomic interactions (Schütt et al., 2017; Batzner et al., 2022; Frank et al., 2023). The architecture has to satisfy a number of physical constraints in order to obtain physically accurate forces and energies (Batzner et al., 2022; Schütt et al., 2021). The score needs to be translation invariant and rotational equivariant to accurately represent molecules in space: rotation/translation of molecules in space leaves the energy unchanged, and the forces are transformed accordingly. Translational invariance is achieved by using relative distances instead of absolute coordinates (Plainer et al., 2025). For rotational equivariance we employ two different methods and compare them. First, rotational equivariance can be approximated via data augmentation (Abramson et al., 2024) by applying random rotations in the training phase. Second, it can also be enforced directly by using only invariant representations (Schütt et al., 2021), (Simeon & de Fabritiis, 2023). Both augmented GT and SE(3) use hidden width 128, four layers and eight heads. Moreover, while SE(3) adds 16 chiral channels, we use rotation augmentation for AugGT, but it excludes reflection symmetries.

Regularizer coefficients. Based on preliminary studies on validation reference data, we select and fix parameters for our experiments, cf. Appendix C for ablation details. For Smoothed FP and Pointwise FP, we use 5×10^{-4} (continuation) and 2×10^{-3} (from scratch). WG-RFF uses 0.3 with 128 features and dataset-specific frequency scales. For Galerkin FP, we use 5×10^{-4}; Stein FP uses 10^{-3}, except 3×10^{-4} for MD22 from scratch.

Test basis and pilot. Test functions are radial basis functions of pair distances, with centers chosen by *farthest-point selection* from an independent pilot drawn uniformly without replacement from the training split, and bandwidths distributed over the scales $\{0.5, 1, 2\}$ times the pilot median distance. The basis is frozen before training and stored with the checkpoint. At low noise, we pair $+\epsilon$ and $-\epsilon$ for each configuration and average the two moment vectors into one cluster a_i; this cancels leading odd noise fluctuations while preserving the conditional moment. We use $M = 64$ test functions, raised to $M = 128$ on Chignolin and MD22. The pilot (4096 configurations) also provides the quadrature for the Gram matrix, from which we draw $n = 256$ points per step, i.e., at least $2M$; with $n \lesssim M$, the empirical Gram would be rank-deficient and the conditioning ridge would silently replace the metric (App. D.1).

Conditioning. We add the smallest ridge satisfying a relative eigenvalue floor and a maximum condition number of 10^5, solve by Cholesky factorization, and log the realized ridge for every step. Negative realizations of the U-statistic are never clamped: they are ordinary fluctuations of an unbiased estimator.

Molecular symmetries. Configurations are centered, noise and scores are projected onto the zero-center-of-mass subspace, and losses are normalized by the active dimension. Scalar pair-distance tests are invariant to rigid motions and their coordinate gradients are equivariant; they are not chirality-complete, which we state as a limitation rather than correct for.

Simulation and evaluation. Equilibrium simulations use overdamped Langevin dynamics at $t = 0$ with $\Delta\tau = 10^{-4}$, 30k steps, 5k steps of burn-in and a stride of 100, from 1024 chains (64 for MD22) started from both reference structures and model samples. Reverse-diffusion samples use 200 integration steps. JS distances use 100 bins per axis; free-energy plots use 48 bins per dihedral axis and 30 per TICA axis, and are capped at the largest value a finite sample resolves, $\log N$, above which $-\log p$ reflects empty bins rather than barriers.

Reporting. Every configuration runs three seeds; tables give means and standard errors and figures show individual seeds. Diverged runs are retained rather than filtered. Raw logs, configurations, evaluation samples and the scripts that produce every table and figure are released with the paper.

C Additional results

As several configurations are bimodal, i.e., a seed either trains or collapses, means and standard errors alone can mislead. We, therefore, show every seed in Fig. 23.

The non-equivariant architecture. Tbl. 3 repeats the main comparison with AugGT. Its reverse-diffusion samples are substantially worse than the SE(3) model's on every system (e.g., JS 0.358 vs. 0.132 on alanine dipeptide), and no regularizer repairs that. Still, all FP estimators improve simulation over DSM (e.g., $0.473 \to 0.346$ for Galerkin FP on KS dipeptide) and remove DSM's bond breaking on KS dipeptide ($52\% \to <1\%$; Tbl. 4). On Chignolin, Galerkin FP reduces broken bonds from 16% to 6.5%; on MD22, only Pointwise FP does ($36\% \to 0.7\%$), while one Galerkin FP seed collapses ($29 \pm 28\%$).

Equal compute. As Galerkin FP is 2.3–3.2$\times$ cheaper per step than Pointwise FP (Tbl. 5), equal-step and equal-compute comparisons answer different questions. On alanine dipeptide from scratch, Pointwise FP reaches JS 0.139 after 30k steps (3.8 GPU-hours), while Galerkin FP reaches 0.144 after 30k steps (1.2 GPU-hours). At matched compute, the ordering reverses: after 10k steps, i.e., the same budget, Pointwise FP scores only 0.238.

Table 2: Protocol dependence on alanine dipeptide (SE(3), three seeds). Both protocols spend the same budget: 30k steps from random initialization, or a 20k-step DSM trunk followed by 10k regularized steps.

Method, protocol	JS (sim) ↓	JS (iid) ↓	broken bonds ↓
DSM, continuation	0.457 ± 0.029	0.134 ± 0.003	0.004 ± 0.000
DSM, scratch	0.504 ± 0.030	0.132 ± 0.006	**0.003** ± 0.000
WG-RFF, continuation	0.458 ± 0.030	0.134 ± 0.002	0.004 ± 0.000
WG-RFF, scratch	0.440 ± 0.006	0.132 ± 0.003	0.390 ± 0.306
Smoothed FP, continuation	0.315 ± 0.053	0.131 ± 0.002	0.006 ± 0.001
Smoothed FP, scratch	0.168 ± 0.006	0.150 ± 0.008	0.013 ± 0.003
Pointwise FP, continuation	0.307 ± 0.031	0.131 ± 0.003	0.008 ± 0.003
Pointwise FP, scratch	**0.139** ± 0.002	0.138 ± 0.002	0.007 ± 0.002
Galerkin FP (ours), continuation	0.384 ± 0.041	0.130 ± 0.003	0.006 ± 0.001
Galerkin FP (ours), scratch	0.144 ± 0.005	0.134 ± 0.001	0.006 ± 0.000
Stein FP (ours), continuation	0.237 ± 0.014	**0.129** ± 0.001	0.005 ± 0.001
Stein FP (ours), scratch	0.366 ± 0.033	0.133 ± 0.002	0.245 ± 0.169

Regularization strength. To test how sensitive our objectives are to their coefficient, we sweep it for one seed (seed 17). Under continuation, Galerkin FP is sensitive to its coefficient (JS 0.329 at 5×10^{-4} against 0.214 at 2×10^{-3} on alanine dipeptide), whereas from scratch it is nearly insensitive: three coefficients spanning a twentyfold range (5×10^{-4} to 10^{-2}) give 0.150–0.158, within the spread over seeds at a fixed coefficient (0.134–0.155). Stein FP shows a monotone trade-off instead: on MD22 under continuation, its validation score magnitude at $t_{\min}$ falls from 22.8 to 13.7 as the coefficient grows from 3×10^{-4} to 3×10^{-3}, while the fraction of states with a broken bond rises from 0.2% to 13.0%. We therefore select its coefficient by the stability constraint rather than by divergence alone.

Where the spikes come from. Figure 17 plots the Galerkin penalty against the diffusion time of the step that produced it. Steps with a penalty above 100 have median time 0.03; the remaining steps have median time 0.50. The strong and projected time-dependent residuals both inherit this behaviour, which is what the endpoint objective avoids by construction.

D Further analyses

D.1 Sensitivity and limitations

Beyond RQ1–3, we ask when our objectives are sensitive to their setup. We separate three sources of sensitivity, i.e., the training protocol, the resolution of the test space, and optimization at low diffusion times, and close with the failures we observed.

Training. As Stein FP constrains only $t = 0$, it can only correct a landscape that another objective has shaped; comparing protocols thus separates the two constraints (RQ3). Tbl. 2 breaks this comparison down for alanine dipeptide at equal total budget, including reverse-diffusion quality and stability. We see that Galerkin FP improves markedly when trained from scratch ($0.384 \to 0.144$), while Stein FP degrades ($0.237 \to 0.366$) and breaks bonds in 25% of states; neither protocol affects reverse-diffusion quality.

Test-space resolution. A finite projection could limit performance through too few test functions or through a poorly estimated Gram geometry; we test which one applies. The projected objective has two knobs that the theory ties together: the number of test functions M, and the number of pilot draws used to estimate the Gram matrix. With n quadrature points the empirical Gram of M test functions has rank at most n; when $n \lesssim M$ the conditioning cap raises the ridge and the ridge, rather than the test space, determines the projection. We observed exactly this: holding n fixed while increasing M from 64 to 512 raised the realized ridge tenfold. We therefore repeat the sweep with $n = 4096$ (seed 17). With the Gram properly resolved, the basis rank has little effect: $M \in \{64, 128, 256, 512\}$ gives JS 0.144, 0.147, 0.160 and 0.147 on alanine dipeptide at nearly identical cost ($+2\%$ per step from

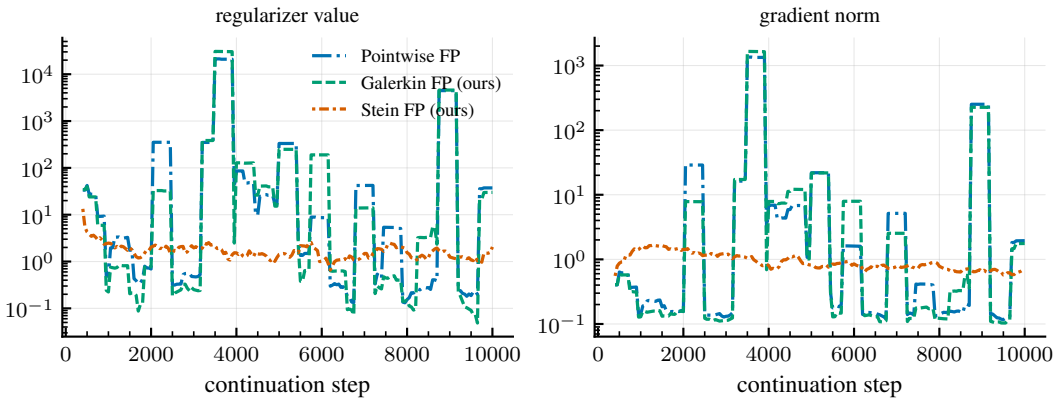


Figure 6: Regularizer value and gradient norm during training (alanine dipeptide, SE(3), seed 17, moving average).

Table 3: Simulation quality for both architectures, trained from scratch. We show the JS distance between simulated and reference distributions, mean and standard error over three seeds.

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe
SE(3)				
MD reference (floor)	0.134	0.128	0.160	0.115
DSM	0.504 ± 0.030	0.396 ± 0.015	0.341 ± 0.021	0.394 ± 0.016
WG-RFF	0.440 ± 0.006	0.476 ± 0.031	0.302 ± 0.034	0.408 ± 0.023
Smoothed FP	0.168 ± 0.006	**0.181** ± 0.003	0.207 ± 0.005	**0.295** ± 0.010
Pointwise FP	**0.139** ± 0.002	0.189 ± 0.017	**0.194** ± 0.003	0.295 ± 0.003
Galerkin FP (ours)	0.144 ± 0.005	0.188 ± 0.009	0.205 ± 0.008	0.305 ± 0.012
AugGT				
MD reference (floor)	0.134	0.128	0.160	0.115
DSM	0.445 ± 0.042	0.473 ± 0.014	0.320 ± 0.020	0.377 ± 0.026
WG-RFF	0.423 ± 0.048	0.401 ± 0.031	0.319 ± 0.012	0.363 ± 0.025
Smoothed FP	0.365 ± 0.002	0.373 ± 0.027	**0.273** ± 0.007	0.323 ± 0.025
Pointwise FP	0.363 ± 0.003	0.388 ± 0.034	0.285 ± 0.011	0.312 ± 0.006
Galerkin FP (ours)	**0.361** ± 0.003	**0.346** ± 0.006	0.276 ± 0.013	**0.312** ± 0.025

$M = 64$ to 512), so $M = 64$ already captures what the projection can recover here, and our main runs ($M \leq 128$, $n = 256$) stay in the resolved regime.

Optimization at low diffusion times. Removing second derivatives does not remove the σ_t^{-2} scaling of the residual at low noise, so we test whether Galerkin FP inherits the optimization difficulty of pointwise FP. Fig. 6 shows that both Pointwise FP and Galerkin FP produce gradient spikes four to five orders of magnitude above their typical value, which stem from small diffusion times (Fig. 17). Thus, Galerkin FP removes the derivative cost but not this difficulty; only Stein FP avoids it by construction.

Failures. We distinguish failures shared with pointwise FP from those specific to the projection. Shared failures stem from the architecture: without equivariance (AugGT), all FP estimators break 6–16% of bonds on Chignolin (Tbl. 4). Specific to the projection, one Galerkin FP seed collapses on MD22 with AugGT ($29 \pm 28\%$ broken bonds), where Pointwise FP stays stable, and on the Müller–Brown potential Galerkin FP turns a minimum into a saddle while matching the basin populations (Fig. 12); both are consistent with residual directions outside the test space. Finally, Stein FP fails from scratch by design, as it constrains only the endpoint.

Table 4: Simulation stability, trained from scratch. We show the fraction of simulated states in which any bond deviates by more than 0.5 Å from its reference length (Fu et al., 2023).

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe
SE(3)				
DSM	**0.003** ± 0.000	**0.001** ± 0.000	0.001 ± 0.000	0.014 ± 0.014
WG-RFF	0.390 ± 0.306	0.004 ± 0.004	**0.001** ± 0.000	0.000 ± 0.000
Smoothed FP	0.013 ± 0.003	**0.001** ± 0.000	0.046 ± 0.027	0.001 ± 0.001
Pointwise FP	0.007 ± 0.002	0.001 ± 0.000	0.027 ± 0.005	0.000 ± 0.000
Galerkin FP (ours)	0.006 ± 0.000	**0.001** ± 0.000	0.024 ± 0.009	**0.000** ± 0.000
AugGT				
DSM	0.018 ± 0.002	0.521 ± 0.288	0.158 ± 0.153	0.362 ± 0.229
WG-RFF	0.016 ± 0.001	0.031 ± 0.027	**0.034** ± 0.017	0.264 ± 0.261
Smoothed FP	0.021 ± 0.002	0.009 ± 0.003	0.118 ± 0.087	0.241 ± 0.076
Pointwise FP	0.019 ± 0.002	**0.007** ± 0.000	0.155 ± 0.062	**0.007** ± 0.004
Galerkin FP (ours)	**0.014** ± 0.001	0.008 ± 0.003	0.065 ± 0.040	0.290 ± 0.280

Table 5: Our regularizer is considerably faster than the FP baseline. We show the training cost per optimizer step (seconds, SE(3))

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe
Training cost per optimizer step (s)				
DSM	**0.079** ± **0.000**	**0.083** ± **0.000**	**0.087** ± **0.000**	**0.094** ± **0.000**
WG-RFF	0.138 ± 0.000	0.147 ± 0.000	0.154 ± 0.000	0.183 ± 0.000
Smoothed FP	0.497 ± 0.018	0.492 ± 0.001	0.499 ± 0.005	0.653 ± 0.003
Pointwise FP	0.452 ± 0.014	0.454 ± 0.001	0.473 ± 0.001	0.598 ± 0.002
Galerkin FP (ours)	0.143 ± 0.002	0.192 ± 0.001	0.198 ± 0.000	0.260 ± 0.000
Stein FP (ours)	0.140 ± 0.002	0.150 ± 0.001	–	0.188 ± 0.001
Peak GPU memory (GB)				
DSM	**2.0** ± **0.0**	**5.3** ± **0.0**	**5.3** ± **0.0**	**7.6** ± **0.0**
WG-RFF	3.9 ± 0.0	10.4 ± 0.0	10.4 ± 0.0	14.9 ± 0.0
Smoothed FP	13.9 ± 0.0	37.4 ± 0.0	37.4 ± 0.0	53.5 ± 0.0
Pointwise FP	13.1 ± 0.0	35.0 ± 0.0	35.0 ± 0.0	49.0 ± 0.0
Galerkin FP (ours)	5.9 ± 0.0	15.8 ± 0.0	15.9 ± 0.0	23.0 ± 0.0
Stein FP (ours)	3.9 ± 0.0	10.5 ± 0.0	–	15.7 ± 0.0

E Additional tables

Continuation protocol. Table 9 repeats the comparison when every method continues from a shared 20k-step DSM checkpoint for 10k regularized steps. Here the endpoint objective (Stein FP) gives the lowest divergences, while the integrated weak formulation (WG-RFF) is indistinguishable from DSM.

Stein FP from scratch. Trained from random initialization, the endpoint objective reaches worse divergences than Galerkin FP and distorts bond geometry (Tables 10 and 11, Figures 15 and 16). It constrains only $t = 0$ and therefore carries no information about the rest of the diffusion path; we use it to refine trained models rather than to train them.

F Additional figures

We group the remaining figures by question: structural observables for every system (Figs. 7–11), the mechanism on an analytic potential (Fig. 12), and further free-energy landscapes, training curves, and reverse-diffusion samples.

Table 6: Null test at the exact model (stationary Gaussian data, zero Fokker–Planck residual), where the population objective and its gradient vanish. Mean $\pm$ standard error over 2000 minibatches of n configurations. The squared minibatch mean is biased by $\mathrm{tr}(A\Sigma)/n$ and its gradient moves the exact model; the U-statistic is unbiased.

n	Estimator	Objective	Gradient
16	Squared minibatch mean	$+31.67 \pm 2.90$	$+8.26 \pm 1.33$
16	Cluster U-statistic (ours)	$+0.51 \pm 0.41$	-0.88 ± 0.66
64	Squared minibatch mean	$+6.87 \pm 0.20$	$+2.12 \pm 0.15$
64	Cluster U-statistic (ours)	-0.01 ± 0.06	$+0.12 \pm 0.12$
256	Squared minibatch mean	$+1.77 \pm 0.04$	$+0.57 \pm 0.05$
256	Cluster U-statistic (ours)	$+0.04 \pm 0.02$	$+0.07 \pm 0.05$

Table 7: Galerkin FP with the biased squared minibatch moment versus the unbiased cluster U-statistic, otherwise identical (alanine dipeptide, SE(3), from scratch, three seeds).

Estimator	JS (sim) ↓	broken bonds ↓	JS (iid) ↓
Squared minibatch mean	0.145 ± 0.002	0.005 ± 0.001	0.134 ± 0.001
Cluster U-statistic (ours)	0.144 ± 0.005	0.006 ± 0.000	0.134 ± 0.001

Table 8: Reverse-diffusion (iid) sample quality, trained from scratch (SE(3)). No objective degrades the generative model.

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe
DSM	**0.132** ± 0.006	0.179 ± 0.002	**0.183** ± 0.004	**0.259** ± 0.001
WG-RFF	0.132 ± 0.003	0.176 ± 0.002	0.188 ± 0.004	0.261 ± 0.001
Smoothed FP	0.150 ± 0.008	0.180 ± 0.001	0.193 ± 0.002	0.264 ± 0.002
Pointwise FP	0.138 ± 0.002	0.178 ± 0.003	0.186 ± 0.002	0.262 ± 0.001
Galerkin FP (ours)	0.134 ± 0.001	**0.175** ± 0.005	0.188 ± 0.002	0.263 ± 0.001

Table 9: Simulation quality under continuation (20k-step DSM trunk plus 10k regularized steps); the reference-versus-reference floors are 0.134, 0.128, 0.160 and 0.115.

Method	Alanine dipeptide	KS dipeptide	Chignolin	MD22 Ac-Ala_3-NHMe
SE(3)				
MD reference (floor)	0.134	0.128	0.160	0.115
DSM	0.457 ± 0.029	0.410 ± 0.020	0.313 ± 0.026	0.407 ± 0.016
WG-RFF	0.458 ± 0.030	0.401 ± 0.021	0.314 ± 0.026	0.397 ± 0.009
Smoothed FP	0.315 ± 0.053	0.349 ± 0.015	0.290 ± 0.003	0.396 ± 0.025
Pointwise FP	0.307 ± 0.031	0.368 ± 0.017	0.299 ± 0.025	0.380 ± 0.020
Galerkin FP (ours)	0.384 ± 0.041	**0.310** ± 0.032	0.315 ± 0.012	0.377 ± 0.025
Stein FP (ours)	**0.237** ± 0.014	0.336 ± 0.028	**0.243** ± 0.004	**0.366** ± 0.009
AugGT				
MD reference (floor)	0.134	0.128	0.160	0.115
DSM	0.456 ± 0.038	0.456 ± 0.047	0.311 ± 0.006	0.352 ± 0.021
WG-RFF	0.457 ± 0.040	0.450 ± 0.052	0.311 ± 0.006	0.347 ± 0.021
Smoothed FP	**0.397** ± 0.015	0.392 ± 0.018	0.301 ± 0.017	0.374 ± 0.013
Pointwise FP	0.404 ± 0.010	0.422 ± 0.027	0.304 ± 0.016	0.375 ± 0.011
Galerkin FP (ours)	0.402 ± 0.003	0.398 ± 0.022	0.296 ± 0.008	0.376 ± 0.008
Stein FP (ours)	0.400 ± 0.004	**0.369** ± 0.007	**0.287** ± 0.008	**0.328** ± 0.008

Table 10: Stein FP trained from scratch (SE(3)). Its divergences are worse than Galerkin FP's, and it distorts bond geometry: the endpoint objective constrains only $t = 0$ and should be used to refine a trained model (Table 9).

System, objective	JS (sim) ↓	broken bonds ↓
Alanine dipeptide, Galerkin FP	0.144 ± 0.005	0.006 ± 0.000
Alanine dipeptide, Stein FP	0.366 ± 0.033	0.245 ± 0.169
KS dipeptide, Galerkin FP	0.188 ± 0.009	0.001 ± 0.000
KS dipeptide, Stein FP	0.320 ± 0.019	0.840 ± 0.089
Chignolin, Galerkin FP	0.205 ± 0.008	0.024 ± 0.009
MD22 Ac-Ala_3-NHMe, Galerkin FP	0.305 ± 0.012	0.000 ± 0.000
MD22 Ac-Ala_3-NHMe, Stein FP	0.331 ± 0.018	0.988 ± 0.012

Table 11: Stein FP trained from scratch (AugGT). Its divergences are worse than Galerkin FP's, and it distorts bond geometry: the endpoint objective constrains only $t = 0$ and should be used to refine a trained model (Table 9).

System, objective	JS (sim) ↓	broken bonds ↓
Alanine dipeptide, Galerkin FP	0.361 ± 0.003	0.014 ± 0.001
Alanine dipeptide, Stein FP	0.458 ± 0.016	0.030 ± 0.005
KS dipeptide, Galerkin FP	0.346 ± 0.006	0.008 ± 0.003
KS dipeptide, Stein FP	0.414 ± 0.017	0.929 ± 0.035
Chignolin, Galerkin FP	0.276 ± 0.013	0.065 ± 0.040
Chignolin, Stein FP	0.310 ± 0.006	0.408 ± 0.167
MD22 Ac-Ala_3-NHMe, Galerkin FP	0.312 ± 0.025	0.290 ± 0.280
MD22 Ac-Ala_3-NHMe, Stein FP	0.333 ± 0.003	0.104 ± 0.099

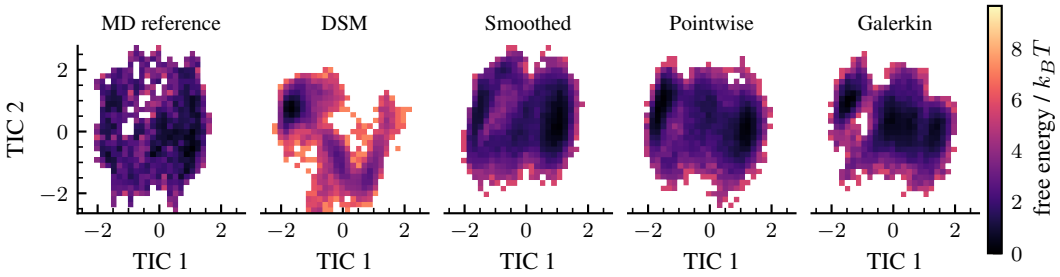


Figure 7: FP objectives cover the reference ensemble of MD22 Ac-Ala_3-NHMe, whereas DSM concentrates on a narrow arc. We show the free energy over the two slowest TICs (from scratch, SE(3)).

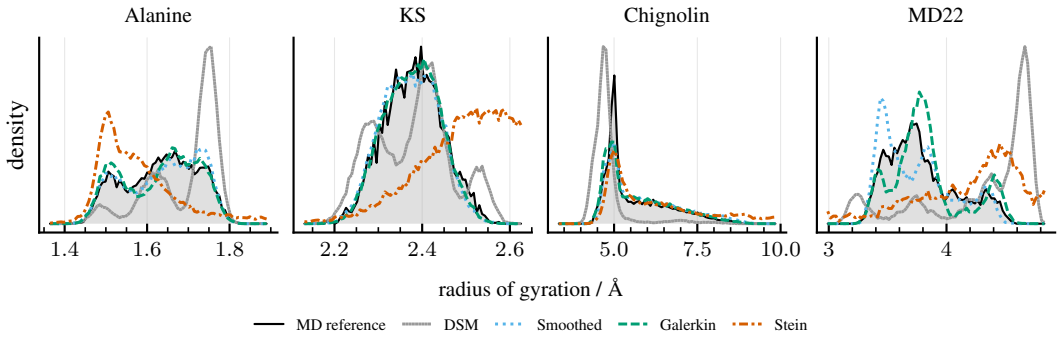


Figure 8: FP estimators reproduce the radius of gyration, whereas DSM and Stein FP from scratch miss it. We show the radius of gyration for all four systems on the shaded reference (from scratch, SE(3), seed 17); DSM is shifted on alanine dipeptide and MD22 and too compact on Chignolin, and Stein FP is shifted on the dipeptides and MD22.

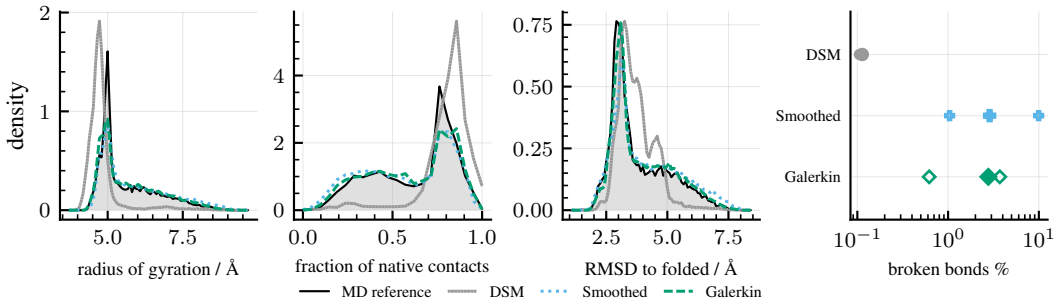


Figure 9: DSM over-concentrates Chignolin in the folded state, whereas FP estimators track the reference ensemble. We show the radius of gyration, the fraction of native contacts, and the RMSD to the folded structure on the shaded reference (from scratch, SE(3), seed 17), and broken bonds per seed (right).

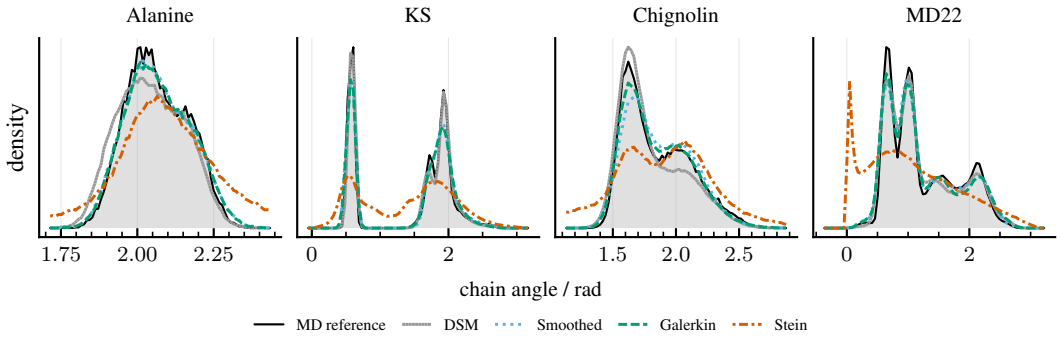


Figure 10: All objectives except Stein FP from scratch reproduce the chain angles. We show the chain bond angle for all four systems on the shaded reference (from scratch, SE(3), seed 17); Stein FP flattens the distribution on every system.

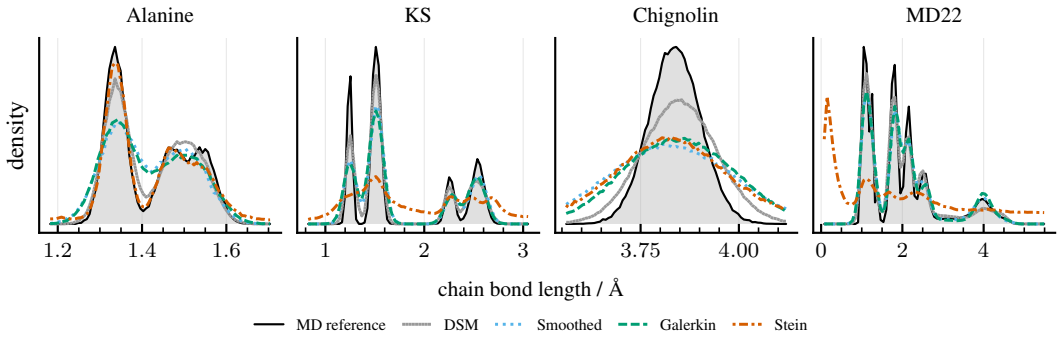


Figure 11: FP estimators broaden bond lengths, the weakest observable. We show the chain bond length for all four systems on the shaded reference (from scratch, SE(3), seed 17); Pointwise and Galerkin FP under-peak on alanine dipeptide, KS dipeptide, and Chignolin, where DSM is closer, and Stein FP breaks bonds on MD22.

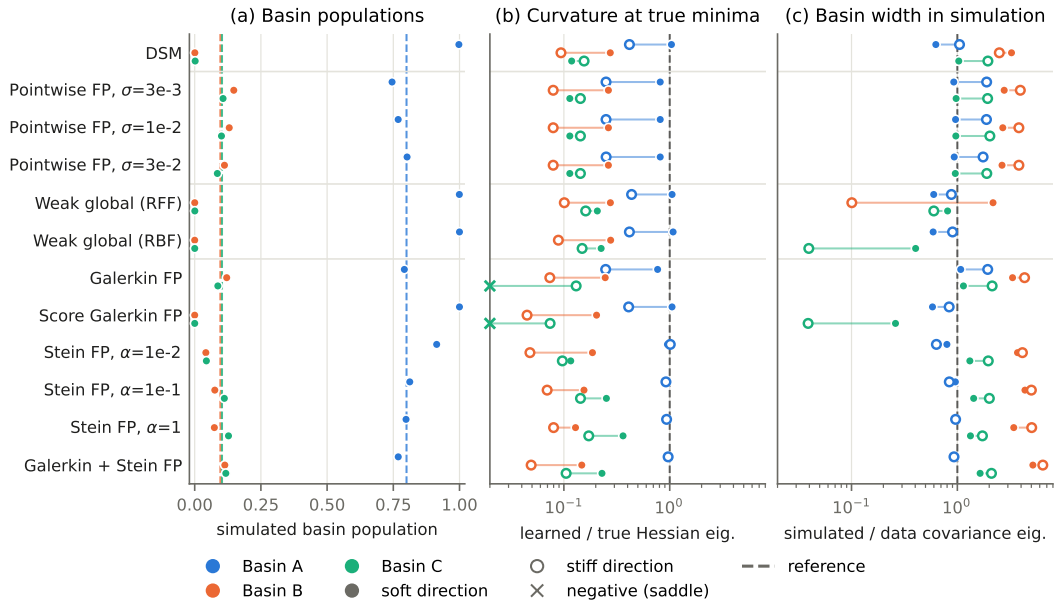


Figure 12: Local geometry on the Müller–Brown potential (one seed). (a) Basin populations in simulation, dashed: data. (b) Learned over true Hessian eigenvalues at the true minima; × marks negative curvature, i.e., a saddle. (c) Simulated over data covariance eigenvalues per basin. Only Stein FP recovers the curvature of the dominant basin in both directions, while Galerkin FP turns basin C into a saddle.

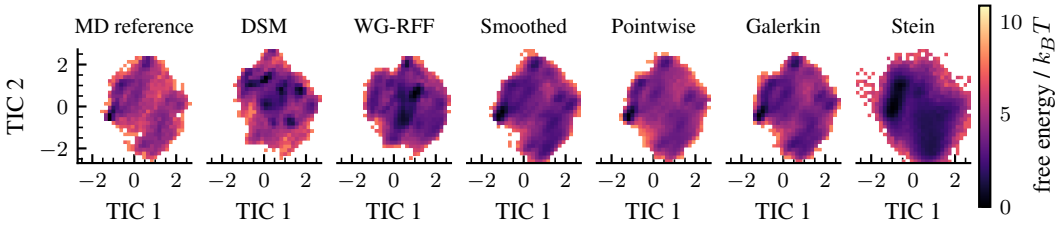


Figure 13: Our objectives recover the free-energy landscape of Chignolin. We see free energy from Langevin over the slowest TICA coordinates (SE(3))

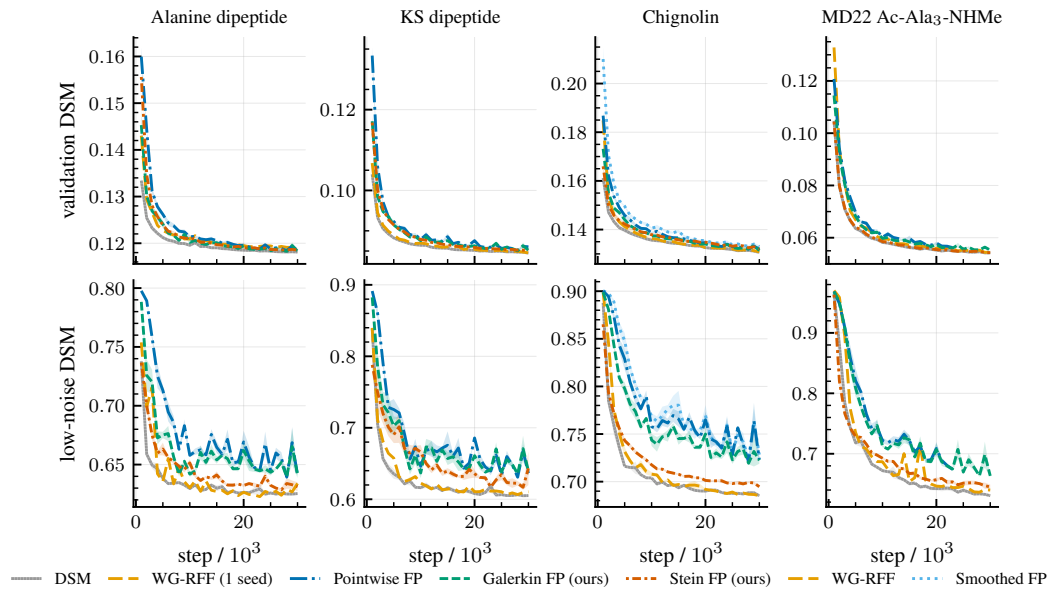


Figure 14: Our regularizer leaves the generative objective intact. Held-out denoising loss (top) and its low-noise part (bottom) over training from scratch (SE(3), mean and standard error over seeds). Both regularizers accept a slightly higher low-noise denoising loss in exchange for far better simulations, so low-noise DSM is not a proxy for force quality.

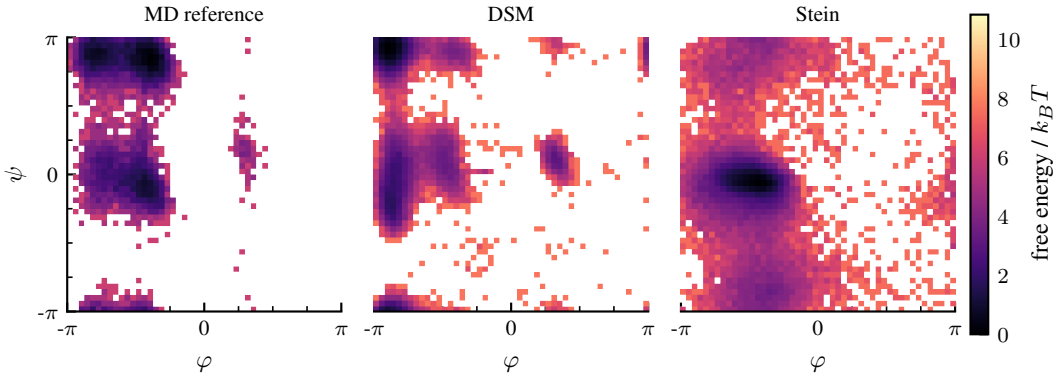


Figure 15: Stein FP trained from scratch collapses the landscape into a single diffuse basin (alanine dipeptide, SE(3)).

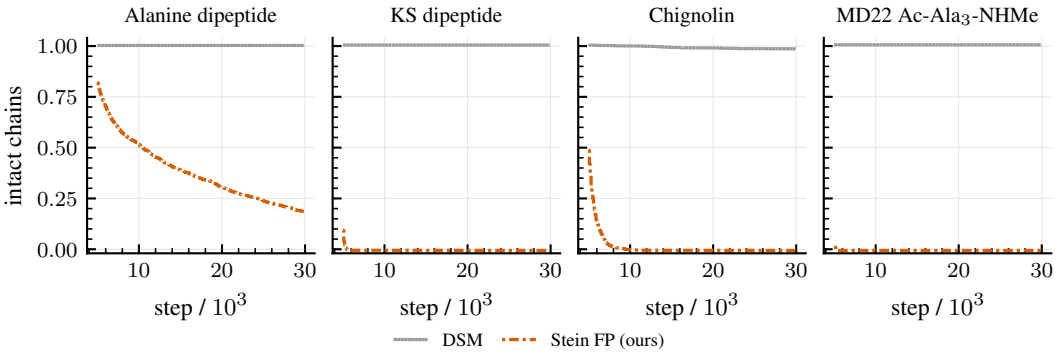


Figure 16: Intact-chain survival including Stein FP trained from scratch (SE(3), seed 17).

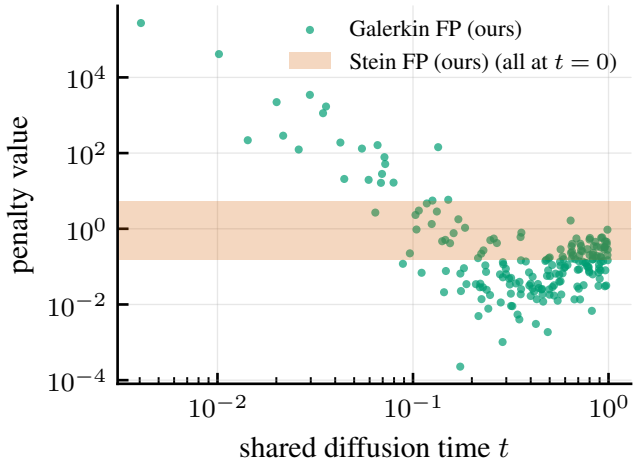


Figure 17: The FP residual grows by orders of magnitude as $t \to 0$, the regime that sets the simulation force. We plot the Galerkin penalty against the shared diffusion time of the step that produced it; the shaded band is the range of the time-independent Stein FP objective.

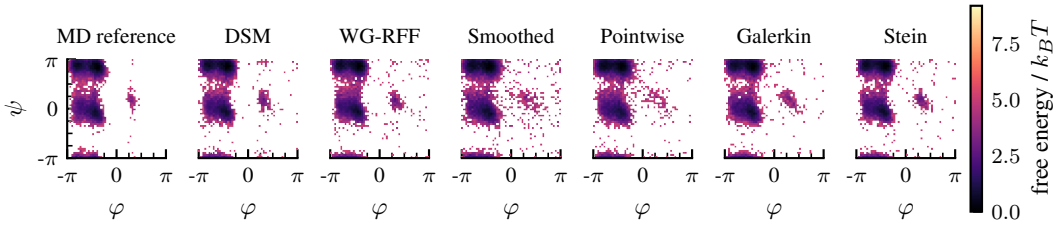


Figure 18: Our methods retain the independent reverse-diffusion samples quality across different noise levels. We show the free energy over (φ, ψ) from (alanine dipeptide, SE(3)), cf. Fig. 2

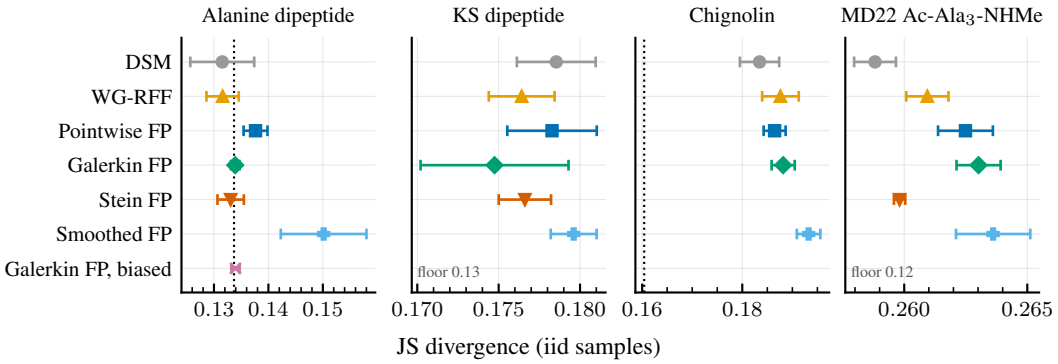


Figure 19: No objective degrades the generative model. We show the reverse-diffusion (iid) sample quality, SE(3).

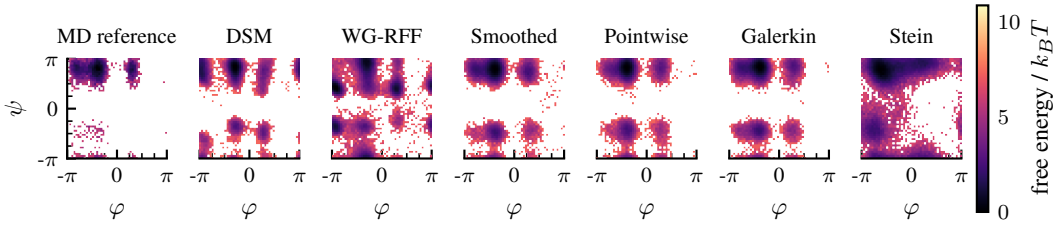


Figure 20: Free energy over (φ, ψ) from Langevin simulation for the KS dipeptide (SE(3)).

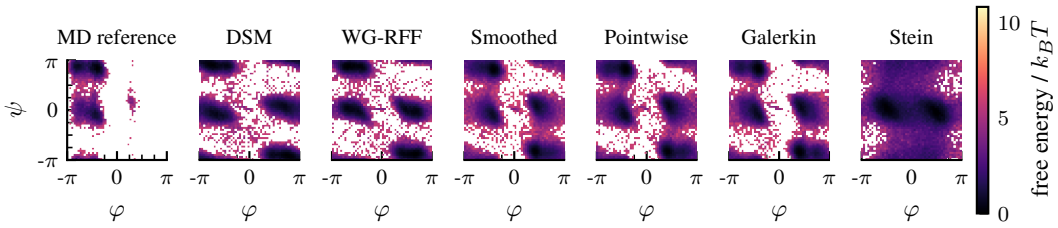


Figure 21: As Figure 2, with the AugGT architecture.

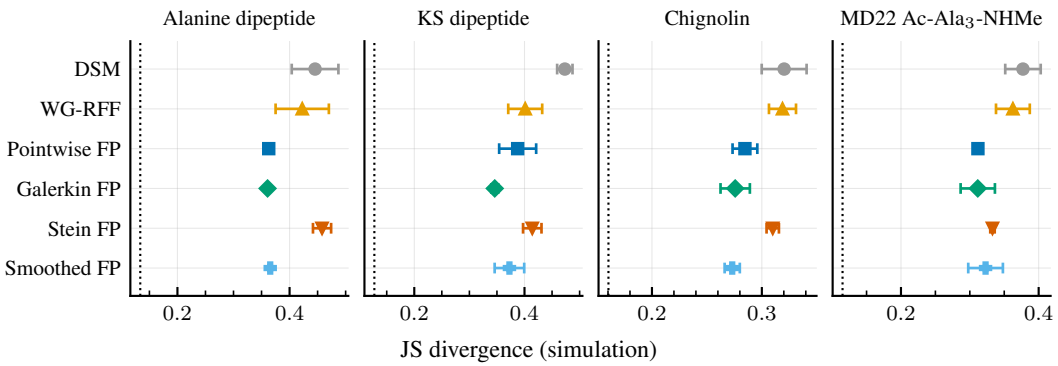


Figure 22: Simulation quality with AugGT.

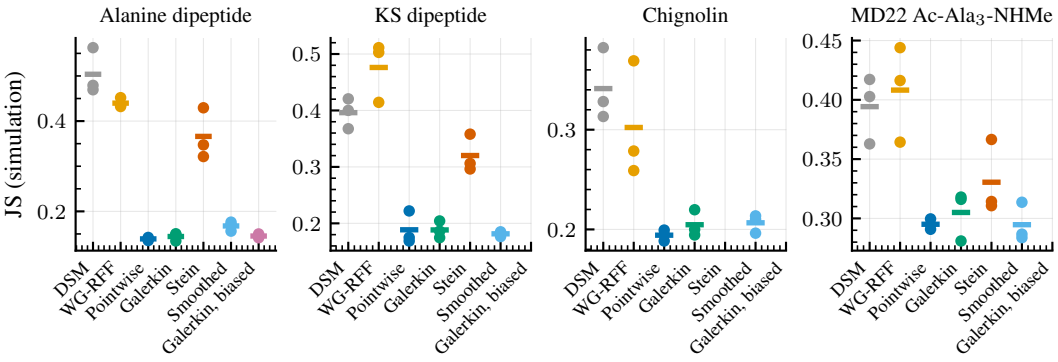


Figure 23: Our method performs consistently. We show per-seed simulation quality (SE(3)) where bars mark the means of Tbl. 3.