WASSERSTEIN SAMPLING VIA BRIDGE INTERPOLANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Langevin dynamics provides a classical route to sampling from unnormalized densities by evolving a distribution along the Wasserstein gradient flow of its Kullback–Leibler divergence to the target. We introduce Wasserstein Sampling via Bridge Interpolants (WASABI), a method for amortizing this sampling process using diffusion models. WASABI translates Wasserstein descent steps in distribution space into updates of the diffusion model's drift, learned through regression on samples generated by the model itself. We derive unbiased estimators that yield a tractable regression objective without evaluating the current model's density or score. The resulting fixed-point algorithm alternates between regression updates using a frozen teacher and sample generation from the updated model, with a replay buffer reusing generated samples and energy-gradient evaluations. On alanine dipeptide, WASABI outperforms reported baselines using an estimated 3.3 times fewer energy evaluations, while remaining competitive on interacting particle systems benchmarks.

1 INTRODUCTION

Sampling from unnormalized densities is of fundamental importance in biology, physics, chemistry, and machine learning. We focus on Boltzmann distributions, $\pi(x) = e^{-U(x)}/\mathcal{Z}$, where $U(x)$ is an energy function and $\mathcal{Z} = \int e^{-U(x)}\,dx$ is the normalizing constant. In statistical physics, Boltzmann distributions characterize equilibrium ensembles and are used to compute thermodynamic observables, including free-energy differences, and study phase behavior (Frenkel & Smit, 2001; Jarzynski, 2002). They are also central to biological problems including protein folding, ligand binding, and conformational changes. (Liu, 2001; Noé et al., 2009; Lindorff-Larsen et al., 2011; Buch et al., 2010). Sampling in these settings is distinct from standard generative modeling, where independent samples from the target distribution are readily available for training. Here, the target is specified only through its energy U and its derivatives, which makes sampling from Boltzmann distributions a significant challenge.

Classical methods such as Markov chain Monte Carlo (MCMC) and molecular dynamics (MD) generate samples sequentially by evolving a Markov chain or simulating a physical trajectory. These methods can yield asymptotically exact samples (Leimkuhler & Matthews, 2015; Frenkel & Smit, 2001), but obtaining representative samples can be computationally demanding. Multimodal targets can contain metastable states separated by high energy barriers, which local updates cross only rarely. The resulting slow mixing produces strongly correlated samples and can require prohibitively long runs to capture transitions between metastable states (Tierney, 1994). These long runs require repeated energy (U) or force ($-\nabla U$) evaluations, each of which can itself be expensive, for example, when computed using density functional theory (Noé et al., 2020).

These costs motivate learning *amortized samplers*, which pay an upfront training cost to construct a generative model that can efficiently produce independent target samples at inference time (Noé et al., 2019). Unlike MCMC or MD, generating additional samples requires only evaluations of the learned model, without further target energy evaluations (Zhang & Chen, 2022; Vargas et al., 2023). This makes amortized samplers particularly useful when many samples from the same target are required. Moreover, parameter sharing and transferability of learned structure across related systems offer a route toward scaling to larger systems (Cheng et al., 2026; Schebek et al., 2026).

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

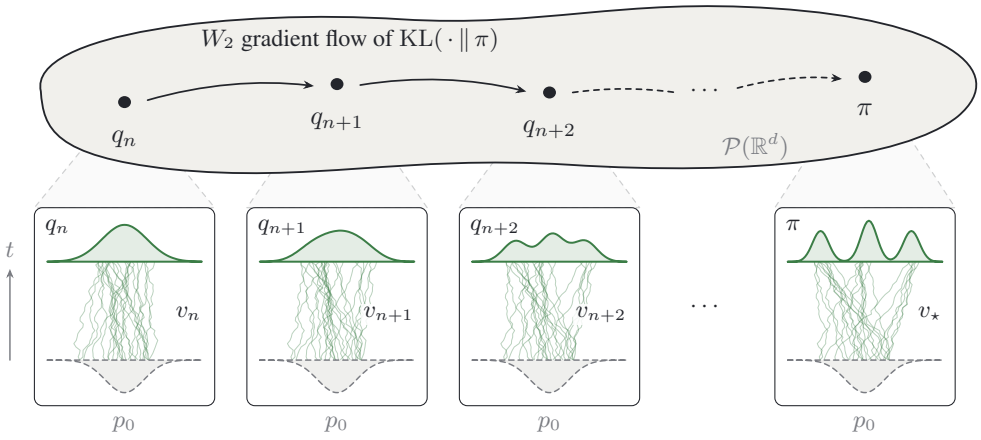


Figure 1: Overview of WASABI. **Top:** the sampling distributions q_n evolve toward the target π through Wasserstein descent of the KL divergence. **Bottom:** the sampling distribution q_n is parameterized by a diffusion model with a prior p_0; WASABI translates Wasserstein descent into updates of the diffusion model, gradually moving it toward the target across iterations n.

Early approaches such as Boltzmann Generators build on normalizing flows to construct samplers for equilibrium distributions (Noé et al., 2019). More recently, diffusion and flow based methods (Song et al., 2021; Lipman et al., 2023) have enabled amortized samplers trained directly from unnormalized targets using evaluations of U and ∇U, without requiring samples from the target distribution (Havens et al., 2025; 2026; Blessing et al., 2026). Recent methods derive tractable training updates from different formulations of the sampling process. Adjoint Sampling uses stochastic optimal control (Havens et al., 2025), BMS constructs a fixed-point iteration from bridge drift identities (Blessing et al., 2026), and Flow Sampling learns denoising dynamics associated with source-conditioned probability paths (Havens et al., 2026). These formulations characterize a generative process connecting the prior to the target and provide regression updates for learning it. Appendix C provides a broader discussion of related work and connections to existing sampling methods.

We consider a different perspective for training amortized samplers: how should the sampling distribution evolve to approach the target π? Langevin dynamics provides a canonical answer: its distributional evolution follows the 2-Wasserstein gradient flow of the KL divergence to the target (Pavliotis, 2014; Jordan et al., 1998). Classical Langevin sampling realizes this evolution by propagating particles. We ask whether this evolution can instead be amortized into a diffusion sampler.

In this paper, we introduce **WAsserstein SAmpling via Bridge Interpolants (WASABI)**, a method that amortizes the Wasserstein gradient flow underlying Langevin dynamics into a reusable diffusion sampler. The key idea is to represent the sampling distribution through a diffusion model's denoiser, so that a prescribed change in the distribution can be translated into an update of the model (Fig. 1). We derive how an infinitesimal Wasserstein descent step toward the target changes this denoising representation. While the resulting update involves intractable conditional expectations, we derive practical, unbiased estimators using a coupled Gaussian-noise construction. These estimators yield a tractable regression objective using generated samples and target energy gradients. Training follows a fixed-point iteration that alternates between fitting the denoiser update using a replay buffer of generated samples and target energy gradients, and refreshing the buffer with new samples from the updated sampler. For difficult multimodal targets, the same procedure can be applied along a sequence of temperature-annealed target distributions.

We summarize our contributions as follows:

- **Amortizing Wasserstein gradient flows.** We introduce a new diffusion sampler by amortizing the Wasserstein gradient flow underlying Langevin dynamics, deriving denoiser updates from infinitesimal Wasserstein descent steps of the sampling distribution.
- **Tractable regression-based learning.** We derive practical, unbiased estimators that make the denoiser updates trainable through regression using generated samples and target energy

gradients. We further develop a coupled estimator and a replay-buffer training scheme that reuses samples and energy-gradient evaluations.

- **Accurate molecular sampling with fewer energy evaluations.** On alanine dipeptide, WASABI reproduces the dominant conformational basins and achieves lower Jensen–Shannon divergence than the best reported ASBS/BMS baseline (Blessing et al., 2026), using an estimated 30% of their energy/force evaluations.

2 BACKGROUND

Diffusion models and bridge interpolants Diffusion models (Ho et al., 2020; Song et al., 2021) and related developments (Lipman et al., 2023; Albergo et al., 2025; Heng et al., 2025) provide tools for generative modeling and sampling through continuous-time dynamics. For independent $\widetilde{X}_0 \sim p_0$, $\widetilde{X}_1 \sim p_1$, and $Z \sim \mathcal{N}(0, I_d)$, where p_0 is the source distribution and p_1 is the target distribution, we define the *bridge interpolants*

$$Y_t = (1-t)\widetilde{X}_0 + t\widetilde{X}_1 + B\sqrt{t(1-t)}\, Z, \qquad 0 \le t \le 1, \tag{1}$$

where $B > 0$ controls the noise scale. The posterior mean of the endpoint under this interpolation defines the *denoiser* $m_t(x) = \mathbb{E}[\widetilde{X}_1 \mid Y_t = x]$. We consider the sampling process via the following stochastic differential equation (SDE)

$$dX_t = \frac{m_t(X_t) - X_t}{1-t}\, dt + B\, dW_t, \qquad X_0 \sim p_0, \quad 0 \le t < 1, \tag{2}$$

where W_t is standard Brownian motion. Under suitable regularity conditions, the SDE has the same marginal distribution at each time t as Y_t (i.e., $X_t \overset{d}{=} Y_t$), and therefore has terminal distribution p_1.

Thus, the denoiser provides an implicit representation of p_1: draw $X_0 \sim p_0$ and simulate Eq. (2) to its terminal time. Conversely, for fixed p_0 and B, the distribution p_1 determines the posterior-mean denoiser through the conditional expectation above. This motivates a denoising representation of a target distribution p_1 via the sampling process in Eq. (2):

Definition 2.1 (Denoising representation of p_1)**.** We call the family $m = (m_t)_{0 \le t < 1}$ a *denoising representation* of p_1 relative to p_0 and B if the solution of Eq. (2), initialized with $X_0 \sim p_0$, has terminal distribution p_1: as $t \uparrow 1$.

Wasserstein gradient flows Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the probability measures on $\mathbb{R}^d$ with finite second moment. In Euclidean space, the gradient describes how a function changes when its input is displaced. Here, we consider a functional $\mathcal{F} : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ that assigns a scalar objective value to a probability distribution, e.g. a divergence to some target distribution. Its gradient likewise describes how the objective value $\mathcal{F}$ changes when that distribution is displaced. We displace a distribution q in the direction of a vector field v by moving each point x to $x + \epsilon v(x)$, where $\epsilon > 0$ is a small step size, which gives $q_\epsilon^v = (\mathrm{id} + \epsilon v)_{\#} q$, where id is the identity map and $\#$ denotes the pushforward (Santambrogio, 2015).[1] This notion of displacement allows us to define the *Wasserstein gradient* of $\mathcal{F}$ at q, denoted $\nabla_{W_2}\mathcal{F}[q]$, as the vector field satisfying

$$\frac{d}{d\epsilon}\mathcal{F}(q_\epsilon^v)\bigg|_{\epsilon=0} = \mathbb{E}_{X\sim q}[\nabla_{W_2}\mathcal{F}[q](X) \cdot v(X)] \tag{3}$$

for every sufficiently regular velocity field v. This identity expresses the directional derivative of $\mathcal{F}$ as an $L^2(q)$ inner product; consequently, the negative Wasserstein gradient gives the direction of steepest descent among velocity fields v with the same $\mathbb{E}_q\|v\|^2$.

We say that a sufficiently regular curve $(q_\tau)_{\tau \ge 0}$ follows 2-Wasserstein gradient flow if, to first order,

$$q_{\tau+\epsilon} = \big(\mathrm{id} - \epsilon\nabla_{W_2}\mathcal{F}[q_\tau]\big)_{\#} q_\tau + o(\epsilon), \tag{4}$$

where the expansion is understood weakly, that is, after integration against smooth test functions. Equivalently, when q_τ admits a density, its evolution satisfies the continuity equation

$$\partial_\tau q_\tau = \nabla \cdot \big(q_\tau \nabla_{W_2}\mathcal{F}[q_\tau]\big). \tag{5}$$

[1] For a map T, the pushforward $T_{\#}q$ is the probability distribution of $T(X)$ when $X \sim q$.

For the Boltzmann target $\pi(x) \propto e^{-U(x)}$, consider the reverse KL objective $\mathcal{F}[q] = \mathrm{KL}(q\|\pi)$. Its Wasserstein gradient is

$$\nabla_{W_2}\mathcal{F}[q](x) = \nabla \log q(x) - \nabla \log \pi(x) = \nabla \log q(x) + \nabla U(x). \tag{6}$$

Substituting this gradient yields the Fokker–Planck equation for overdamped Langevin dynamics,

$$dZ_\tau = -\nabla U(Z_\tau)\, d\tau + \sqrt{2}\, dW_\tau, \tag{7}$$

where W_τ is standard Brownian motion and π is the stationary density. Thus, the distribution of the Langevin process follows the 2-Wasserstein gradient flow of the reverse KL (Jordan et al., 1998). Here τ denotes the time of this distributional evolution, distinct from the bridge time t.

Stein's identity Let r be a smooth density with score $s_r = \nabla \log r$. For any sufficiently regular vector field $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, integration by parts gives

$$\mathbb{E}_{X\sim r}[s_r(X) \cdot \varphi(X)] = -\mathbb{E}_{X\sim r}[\nabla \cdot \varphi(X)]\,, \tag{8}$$

provided the boundary term vanishes. Thus, inside an expectation, a derivative of the density can be transferred to the test field. For $Z \sim \mathcal{N}(0, I_d)$, whose score is $-Z$, this becomes $\mathbb{E}[Z \cdot \varphi(Z)] = \mathbb{E}[\nabla \cdot \varphi(Z)]$. This is *Stein's identity*, which we use to derive the estimators in Sec. 3.2.

3 Method

In this section, we introduce WASABI, a diffusion sampler for the Boltzmann distribution π that amortizes the distributional evolution of Langevin dynamics. The key idea is to parameterize the sampling distribution through a diffusion model's denoising representation and translate the prescribed distributional evolution into an update of this representation. This yields a regression update learned from generated samples and target energy gradients.

In Sec. 3.1, we describe the model setup and derive the denoiser update induced by Wasserstein descent. In Sec. 3.2, we construct estimators that avoid evaluating the unknown score of the model's distribution. Finally, Sec. 3.3 combines these ingredients into the WASABI training algorithm.

3.1 Model Setup and Denoiser Updates

We take $\mathcal{F}[q] = \mathrm{KL}(q\|\pi)$ as the objective and parameterize the sampling distribution q through a diffusion model's denoising representation (Definition 2.1). With the source distribution p_0 and diffusion coefficient $B > 0$ fixed, WASABI progressively updates this representation. We index these updates by n, writing q_n for the current sampling distribution and $m_t^{(n)}$ for its denoiser. The denoiser determines the drift of the sampling SDE in Eq. (2), whose endpoint distribution is q_n.

To minimize KL, we take a small step along the negative Wasserstein gradient in Eq. (6) at q_n,

$$-\nabla_{W_2}\mathcal{F}[q_n] = \nabla \log \pi - \nabla \log q_n.$$

For a step size $\eta > 0$, the desired endpoint distribution is

$$q_{n,\eta} = \big(\mathrm{id} - \eta \nabla_{W_2}\mathcal{F}[q_n]\big)_{\#} q_n. \tag{9}$$

For the same source p_0 and diffusion coefficient B, the transported distribution $q_{n,\eta}$ induces a new denoising representation $m_{t,\eta}^{(n)}$. We characterize how the denoiser changes under a small endpoint transport step by computing its derivative with respect to η at $\eta = 0$. The following theorem gives this derivative.

Theorem 3.1 (Denoiser change induced by Wasserstein descent)**.** *Let $q_{n,\eta} = (\mathrm{id} - \eta\nabla_{W_2}\mathcal{F}[q_n])_{\#}q_n$, and let $m_{t,\eta}^{(n)}$ be its denoiser, using the same initial distribution p_0 and noise scale B, so that $m_{t,0}^{(n)} = m_t^{(n)}$. Write $h_t = B\sqrt{t(1-t)}$ and $Y_t = (1-t)\widetilde{X}_0 + t\widetilde{X}_1 + h_t Z$, where $\widetilde{X}_0 \sim p_0$, $\widetilde{X}_1 \sim q_n$, and $Z \sim \mathcal{N}(0, I)$ are independent. The expectation and covariance in Eq.* (10) *are conditional on $Y_t = y$. Assume the conditional moments below are finite and differentiation*

under the conditional integrals is valid. Then, for $0 < t < 1$*,*

$$\frac{\partial}{\partial \eta} m_{t,\eta}^{(n)}(y)\Big|_{\eta=0} = \underbrace{-\mathbb{E}\Big[\nabla_{W_2}\mathcal{F}[q_n](\widetilde{X}_1)\Big]}_{\mathcal{M}_t[q_n](y)} + \underbrace{\frac{t}{h_t}\,\mathrm{Cov}\Big(\widetilde{X}_1, -Z\cdot\nabla_{W_2}\mathcal{F}[q_n](\widetilde{X}_1)\Big)}_{\mathcal{C}_t[q_n](y)}. \tag{10}$$

Moreover, if $q_n = \pi$*, then* $\nabla_{W_2}\mathcal{F}[\pi] = 0$*, and hence* $\mathcal{M}_t[\pi] = \mathcal{C}_t[\pi] = 0$*.*

Transporting the endpoints changes the denoiser's conditional average in two ways. The mean term $\mathcal{M}_t[q_n](y)$ captures their movement, using the original conditional distribution. But moving an endpoint also changes how likely it is to produce the same observation $Y_t = y$ under the interpolation. Endpoints that become more likely given this observation contribute more to the conditional average, and those that become less likely contribute less. The covariance term $\mathcal{C}_t[q_n](y)$ accounts for this change in the conditional distribution. Together, the two terms give the full first-order change of the denoiser. This motivates the WASABI update operator, with a time-dependent step size $\eta_t > 0$ and a weight λ_t controlling the covariance contribution,

$$\mathcal{T}[m^{(n)}]_t(y) := m_t^{(n)}(y) + \eta_t\Big(\mathcal{M}_t[q_n](y) + \lambda_t\mathcal{C}_t[q_n](y)\Big). \tag{11}$$

With $\lambda_t = 1$ and a common step size $\eta_t = \eta$, this update agrees to first order with the denoiser induced by the desired transport in Eq. (9). Allowing η_t and λ_t to vary with bridge time modifies this direction or its step size. Since both contributions vanish at $q_n = \pi$, the update leaves the denoiser of π unchanged for any finite λ_t; hence, π is a fixed point of the endpoint update.

3.2 Estimating the denoiser update

The update direction is now specified, but it involves the unknown endpoint score $\nabla \log q_n$. We can generate samples from q_n using the current sampler and evaluate the target score $-\nabla U$. Stein's identity (Eq. (8)) lets us express the required expectations in terms of these samples, energy gradients, and bridge noise, eliminating the need to evaluate $\nabla \log q_n$. The estimators below use independent endpoints. For ALA2, we instead Kabsch-align the prior configuration to its paired endpoint; the resulting coupling and the corresponding estimators are derived in Appendices A.2–A.5.

Estimator for $\mathcal{M}_t[q_n]$. The first component of the denoiser update is

$$\mathcal{M}_t[q_n](y) = \mathbb{E}\Big[-\nabla U(\widetilde{X}_1) - \nabla \log q_n(\widetilde{X}_1)\,\Big|\,Y_t = y\Big]. \tag{12}$$

The target score $-\nabla U$ is available through energy gradients. A straightforward application of Stein's identity (Eq. (8)) yields

$$\mathcal{M}_t[q_n](y) = \mathbb{E}\left[-\nabla U(\widetilde{X}_1) + \frac{t}{h_t}Z\,\middle|\,Y_t = y\right]. \tag{13}$$

The expression inside the expectation is a computable regression target: its conditional mean given $Y_t = y$ equals $\mathcal{M}_t[q_n](y)$. Thus, endpoint samples and bridge-noise suffice to learn this component.

Estimator for $\mathcal{C}_t[q_n]$. To estimate the covariance term, we consider its expected inner product with a smooth test function $\varphi : \mathbb{R}^d \to \mathbb{R}^d$. Writing $s_\pi = -\nabla U$ and $F_\varphi(y, x) := \varphi(y)\cdot(x - m_t^{(n)}(y))$, an application of Stein's identity gives

$$\mathbb{E}[\varphi(Y_t)\cdot\mathcal{C}_t[q_n](Y_t)] = \frac{t}{h_t}\mathbb{E}\left[\varphi(Y_t)\cdot Z + F_\varphi(Y_t, \widetilde{X}_1)\left(Z\cdot s_\pi(\widetilde{X}_1) + \frac{t}{h_t}(\|Z\|^2 - d)\right)\right]. \tag{14}$$

The derivation is provided in App. A.4.

Although this expression removes the unknown score, direct Monte Carlo estimates had high variance in our early experiments. We seek an alternative representation of the same expectation by changing how bridge noise is sampled. Completing the square, with $a_t(x) := \frac{h_t}{2t}s_\pi(x)$, gives $Z\cdot s_\pi(x) + \frac{t}{h_t}(\|Z\|^2 - d) = \frac{t}{h_t}(\|Z + a_t(x)\|^2 - d - \|a_t(x)\|^2)$.

The nonnegative factor $\|Z + a_t(x)\|^2$ can be absorbed into a normalized noise density. This replaces a random weight by sampling from a tilted Gaussian distribution. The resulting estimator compares evaluations under the original and tilted noise distributions while preserving the same expected covariance contribution.

Theorem 3.2 (Tilted-noise covariance estimator)**.** *Let ϕ be the standard Gaussian density on $\mathbb{R}^d$. Then*

$$\widetilde{\phi}_t(z \mid x) = \frac{\|z + a_t(x)\|^2}{d + \|a_t(x)\|^2}\,\phi(z) \tag{15}$$

is a probability density. Conditional on $(\widetilde{X}_0, \widetilde{X}_1)$, let Z and $\widetilde{Z}$ have respective distributions ϕ and $\widetilde{\phi}_t(\cdot \mid \widetilde{X}_1)$, with any coupling. Set $\widetilde{Y}_t = (1-t)\widetilde{X}_0 + t\widetilde{X}_1 + h_t\widetilde{Z}$. The estimator

$$\widehat{\mathcal{C}}_t[\varphi] = \frac{t}{h_t}\varphi(Y_t)\cdot Z + \frac{t^2}{h_t^2}\big(d + \|a_t(\widetilde{X}_1)\|^2\big)\big[F_\varphi(\widetilde{Y}_t, \widetilde{X}_1) - F_\varphi(Y_t, \widetilde{X}_1)\big] \tag{16}$$

is an unbiased estimator of $\mathbb{E}[\varphi(Y_t)\cdot \mathcal{C}_t[q_n](Y_t)]$, provided the relevant expectations are finite, the integration-by-parts boundary terms vanish, and $m_t^{(n)}(y) = \mathbb{E}[\widetilde{X}_1 \mid Y_t = y]$.

The proof is given in App. A.4; Lemma A.8 provides a procedure for sampling the tilted noise.

The two noise draws may be coupled so that the estimator uses paired evaluations at Y_t and $\widetilde{Y}_t$. Its expectation is unchanged by the coupling, but its variance can depend on that choice. Unbiasedness does not imply lower variance.

3.3 The WASABI Algorithm

We now turn the distributional update into a training loop. At each outer iteration, we fit the denoiser $m^{(n)}$ using an endpoint sample set $\mathcal{D}_n$ drawn from q_n. Holding this denoiser fixed, we learn a correction $f_\theta(y,t)$ to approximate the direction

$$r_t^{(n)}(y) := \eta_t^{-1}\left(\mathcal{T}[m^{(n)}]_t(y) - m_t^{(n)}(y)\right) = \mathcal{M}_t[q_n](y) + \lambda_t \mathcal{C}_t[q_n](y). \tag{17}$$

Fitting this direction amounts to the ideal regression problem

$$\min_\theta \ \frac{1}{2}\mathbb{E}\left[\left\| f_\theta(Y_t,t) - r_t^{(n)}(Y_t)\right\|^2\right]. \tag{18}$$

We do not need to evaluate $r_t^{(n)}(Y_t)$ itself to train the correction. Expanding the square leaves a known quadratic term and an inner product with the update direction; the remaining term is independent of θ. Substituting the mean and covariance estimators from Sec. 3.2 yields an objective with the same population minimizers,

$$\mathcal{L}_n(\theta) = \mathbb{E}\left[\frac{1}{2}\|f_\theta(Y_t,t)\|^2 - f_\theta(Y_t,t)\cdot\left(-\nabla U(\widetilde{X}_1) + \frac{t}{h_t}Z\right) - \lambda_t\widehat{\mathcal{C}}_t[f_\theta(\cdot,t)]\right]. \tag{19}$$

The expectation includes the sampled bridge time, endpoints, and original and tilted noise. We minimize this objective using minibatch estimates, keeping $m_t^{(n)}$ fixed inside the covariance estimator. After fitting the correction, we use the updated denoiser $\widetilde{m}_t^{(n+1)}(y) = m_t^{(n)}(y) + \eta_t f_\theta(y,t)$ in Eq. (2) to generate fresh endpoints for $\mathcal{D}_{n+1}$. The updated sampler's endpoint distribution defines the next iterate q_{n+1}. The learned update need not be the exact denoiser for this new distribution, so we refit that posterior mean before constructing the next correction. This closes the loop: fit the denoiser, learn its correction, update the sampler, and generate new endpoints, as summarized in Algorithm 1.

In practice, we maintain the endpoint sample set as a replay buffer, retaining samples across iterations and partially refreshing it with samples from the updated sampler. Batches from this buffer are used to fit both the denoiser and its correction. Because retained samples may come from earlier samplers, this reuse approximates the population iteration written in terms of q_n. For difficult

Algorithm 1 WASABI

Inputs: Energy U; source p_0; diffusion coefficient B; initial endpoint sample set $\mathcal{D}_0$; iterations N; schedules η_t, λ_t.

1: **for** $n = 0, \dots, N-1$ **do**
2: **Fit denoiser:** $m^{(n)} \approx \arg\min_m \mathbb{E}[\|m_t(Y_t) - \widetilde{X}_1\|^2]$, using $\widetilde{X}_1 \sim \mathcal{D}_n$.
3: **Fit correction:** freeze $m^{(n)}$ and minimize $\mathcal{L}_n(\theta)$ in Eq. (19) to obtain θ_n.
4: **Update sampler:** $\widetilde{m}_t^{(n+1)}(y) \leftarrow m_t^{(n)}(y) + \eta_t f_{\theta_n}(y, t)$.
5: **Refresh sample set:** generate $\mathcal{D}_{n+1}$ by simulating Eq. (2) with $\widetilde{m}^{(n+1)}$.
6: **end for**
7: **return** The diffusion sampler driven by $\widetilde{m}^{(N)}$.

targets, we also use annealing: $U_{\beta_n} = \beta_n U$, with β_n increasing toward 1. At each stage, the target score $-\beta_n \nabla U$ replaces $-\nabla U$ in both estimators, allowing the sampler to adapt through a sequence of tempered targets. Algorithm 2 gives the extended procedure with replay, optional alignment, and annealing.

4 Experiments

Following previous literature, we evaluate WASABI on three benchmark Boltzmann distributions: the four-particle double well (DW-4), a 13-particle Lennard–Jones cluster (LJ-13), and alanine dipeptide (ALA2) (Köhler et al., 2020; Midgley et al., 2023). We assess how well it reproduces each target's energy and structure, and compare its performance to existing diffusion samplers.

Our comparison methods are Adjoint Sampling (AS) (Havens et al., 2025), Adjoint Schrödinger Bridge Sampler (ASBS) (Liu et al., 2025), and Bridge Matching Sampler (BMS) (Blessing et al., 2026), including damped variants where reported. For LJ-13 and ALA2, baseline results are taken from Blessing et al. (2026); the DW-4 methods are trained under matched settings. Experiment details are given in Appendix B.

4.1 Double-well system (DW-4)

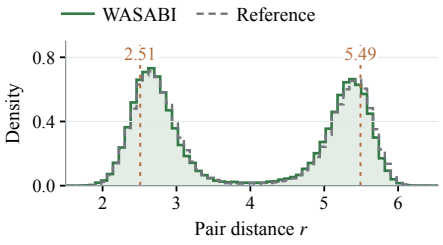


Table 1: DW-4 at 1,000 rounds.
16×4 EGNN; matched training settings.

Method	Config. $W_2 \downarrow$	Energy $W_2 \downarrow$
ASBS	$\mathbf{0.94 \pm 0.16}$	2.35 ± 0.73
ASBS (damped)	0.97 ± 0.30	2.31 ± 0.34
BMS	1.27 ± 0.19	2.87 ± 1.23
BMS (damped)†	1.48 ± 0.21	2.72 ± 0.58
WASABI	1.13 ± 0.39	$\mathbf{1.89 \pm 0.43}$

Mean $\pm$ SD over six seeds. †Five seeds; one divergent run excluded (energy $W_2 = 2{,}449$).

Figure 2: **DW-4. (Left)** Pooled pairwise distances $\{r_{ij}\}_{i<j}$ from 10,000 reference and 10,000 WASABI samples; dashed lines mark the pair-potential minima. **(Right)** Matched-training comparison using samples pooled over rounds 625–1,000. W_2 compares particle configurations (Config.) and unfiltered potential energies (Energy).

We first consider DW-4, a two-dimensional system of four particles with pairwise double-well interactions (Appendix B.2). Each pairwise term is minimized at two distances, $r_{ij} \approx 2.51$ and 5.49, favoring competing structural arrangements. Interactions among all four particles can shift the peaks of the pair-distance distribution, although we observe them near these minima. WASABI reproduces both peaks of the pair-distance distribution (Fig. 2). Under matched training settings, it achieves the lowest mean energy W_2, while ASBS achieves the lowest configuration W_2, indicating that the methods differ in how accurately they reproduce energies and particle arrangements (Tab. 1).

4.2 LENNARD–JONES CLUSTER (LJ-13)

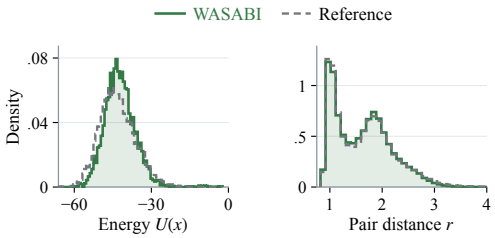


Table 2: LJ-13 comparison.

Method	Config. $W_2 \downarrow$	Energy $W_2 \downarrow$
AS	1.67 ± 0.01	2.40 ± 1.25
ASBS	1.59 ± 0.03	1.99 ± 1.01
BMS	$\mathbf{1.54 \pm 0.00}$	$\mathbf{0.53 \pm 0.04}$
WASABI	1.597 ± 0.003	1.677 ± 0.224

Figure 3: **LJ-13. (Left)** Energy and pooled pair-distance $\{r_{ij}\}_{i<j}$ densities: WASABI (green, three-seed mean) and reference (dashed). **(Right)** Configuration W_2 compares particle arrangements after matching and alignment; energy W_2 compares energy distributions. Results are mean $\pm$ SD across three seeds; baselines are quoted from Blessing et al. (2026, Table 2).

We next consider 13 identical particles in three dimensions with Lennard–Jones interactions and quadratic confinement (Appendix B.2). The pair interaction strongly repels nearby particles and favors a separation of $r_{ij} = 1$, while the quadratic term penalizes dispersed clusters.

We train a center and a correction PaiNN, each with four layers and 64 channels, from a Gaussian prior with scale $\sigma_0 = 1.75$. We evaluate three seeds after 7,000 outer iterations, using 128 sampling steps. WASABI closely matches the pooled pair-distance distribution, although its energy density is more concentrated near the peak and underrepresents the low-energy shoulder (Fig. 3). Its configuration W_2 is close to ASBS, and its mean energy W_2 is lower than AS and ASBS, while BMS performs better on both metrics (Tab. 2).

4.3 ALANINE DIPEPTIDE (ALA2)

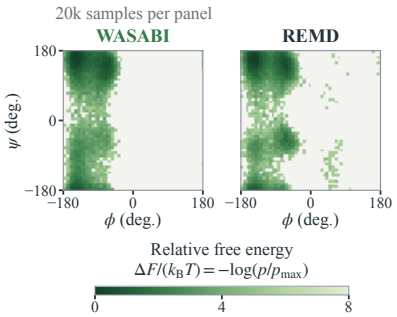


Table 3: Accuracy and computational cost.

Method	Train $\downarrow$	Energy $\downarrow$	$D_{\mathrm{JS}} \downarrow$
ASBS	—	10.24^2	0.232 ± 0.018
ASBS (damped)	—	10.24^2	0.111 ± 0.017
BMS	112.9^2	10.24^2	0.311 ± 0.050
BMS (damped)	190.2^2	10.24^2	0.068 ± 0.009
WASABI	**31.7**	**3.13**[3]	$\mathbf{0.0254 \pm 0.0033}$
REMD	—	73.5	—

Figure 4: **ALA2 at 300 K. (Left)** Ramachandran plots comparing 20k WASABI and 20k REMD samples; gray bins are unobserved. **(Right)** Jensen–Shannon divergence and estimated computational cost. WASABI results are mean $\pm$ SD across three seeds; baselines are quoted from Blessing et al. (2026). Training cost is in TFLOPs per outer iteration, energy/force calls in millions, and D_{JS} in nats; — denotes unreported values.

We finally evaluate WASABI on alanine dipeptide (ALA2). Its energy function includes terms describing bond stretching and angle bending, as well as preferred molecular conformations, electrostatic interactions, and the surrounding solvent (details in Appendix B.3). As in prior work (Blessing et al., 2026), we compare the sampled energy and backbone-angle distributions with molecular-dynamics reference samples.

Throughout our ALA2 experiments, we use molecular-dynamics configurations sampled at 1,000 K as the prior. During training, each sampled prior configuration is Kabsch-aligned to its paired endpoint from the replay buffer. The alignment and its derivative corrections are described in Appendices A.2–A.5. Generating these samples on an Apple M5 Max chip takes approximately five minutes using one CPU thread. For the main results and hyperparameter sweeps, we train for 4,000

[2]Inferred eight-GPU setting based on the released codebase.

[3]3.13M calls: 2M prior production + 0.1M equilibration + 1.03M training (4,000 outer steps; 4,096-endpoint bank; 256 fresh per step).

outer iterations, annealing the target from 1,000 K to 300 K over the first 1,500 iterations. We maintain a ring replay buffer of 4,096 endpoints and replace 256 endpoints with fresh samples at each outer iteration.

Using these settings, WASABI achieves a Jensen–Shannon divergence of 0.0254 ± 0.0033, using the same evaluation procedure as ASBS and BMS (Liu et al., 2025; Blessing et al., 2026), with one million generated samples per seed and one million reference configurations. WASABI trains for 4,000 outer iterations at an estimated 31.7 TFLOPs per iteration. BMS reports 5,000 outer iterations for ALA2 (Blessing et al., 2026); under the inferred eight-GPU configuration, we estimate costs of 112.9 and 190.2 TFLOPs per iteration for its undamped and damped variants, respectively (Table 3). Figure 4 and Table 3 summarize our results.

Influence of hyperparameters. We examine the effects of covariance weight, update step size, and replay-buffer settings in Appendix B.4. Among the settings tested, WASABI performs best on ALA2 with covariance weights below 0.5. To limit the computational cost of the covariance-weight and step-size ablations, which comprised 96 training runs requiring approximately 8–10 GPU-hours each, we used a replay-buffer capacity of 4,096 endpoints and refreshed 256 endpoints per outer iteration. Low Jensen–Shannon divergence can coexist with occasional underrepresentation of a smaller backbone-angle mode. Increasing buffer capacity to 8,192 and the refresh count to 512 mitigates this behavior in the evaluated runs.

5 Conclusion

In this paper, we introduced WASABI, an algorithm that learns a diffusion sampler for Boltzmann distributions by translating Wasserstein descent of $\mathrm{KL}(q_n\|\pi)$ into first-order updates of its bridge denoiser. On alanine dipeptide, WASABI achieves a Jensen–Shannon divergence of 0.0254 ± 0.0033, compared with 0.068 ± 0.009 for the best reported ASBS/BMS variant. Including training and local high-temperature prior generation, we estimate a total budget of 3.13 million energy/force evaluations, compared with 10.24 million for ASBS/BMS—a roughly 3.3-fold reduction. Under matched training settings on DW-4, WASABI also achieves the lowest mean energy W_2.

More broadly, WASABI shows how the distributional evolution underlying Langevin dynamics can be translated into a practical learning rule for a diffusion sampler. We hope this connection between Wasserstein gradient flows and generative modeling offers a useful direction for molecular sampling, where information obtained from expensive energy and force evaluations can be learned once and reused across many subsequent samples.

Limitations WASABI uses first-order denoiser updates, and finite model fitting and replay-buffer reuse introduce further approximations to the ideal population iteration. On LJ-13, WASABI underrepresents the low-energy shoulder and trails BMS on both configuration and energy Wasserstein distances. On alanine dipeptide, low Jensen–Shannon divergence can coexist with underrepresentation of a smaller conformational mode; increasing replay capacity and refresh count mitigates this behavior in the evaluated runs but does not ensure complete mode coverage.

ETHICS STATEMENT

This work develops methods for sampling Boltzmann distributions and evaluates them on simulated particle and molecular systems. It does not involve human participants, personal data, or animal experiments. We do not identify ethical concerns specific to the experiments reported here.

The experiments reported in this work were conducted on shared H100 GPU infrastructure at an in-house computing facility. We estimate that reproducing the reported experiments, including the main ablations and evaluation curves, requires approximately 1,000 H100 GPU-hours. Including exploratory experiments, repeated trials, and unsuccessful approaches, we estimate the total compute attributable to this project to be approximately 3,000–4,000 H100 GPU-hours. Because compute accounting is shared across concurrent projects, these figures are approximate.

At a nominal H100 power draw of up to 700 W, this corresponds to approximately 2.1–2.8 MWh of direct GPU energy for the project as a whole, excluding shared system and facility overhead. To acknowledge the associated environmental impact, one of the lead authors of this paper commits to retiring three tonnes of verified carbon credits through the Gold Standard Impact Registry. A public registry record documenting the retirement will be included in the camera-ready version.

REPRODUCIBILITY STATEMENT

To support reproducibility, Appendix A provides the theoretical derivations and the extended training algorithm, including replay, alignment, and annealing. Appendix B describes the benchmark potentials, training settings, evaluation protocols, and computational-cost estimates. We will publicly release the code after publication.

AI USE STATEMENT

The authors used AI assistants to support manuscript editing, reformatting author-written proofs, and coding, analysis, and metric computation at the level of individual functions. AI assistants also helped with figure and table preparation. The authors individually verified all reported results.

REFERENCES

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from boltzmann densities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 760–786. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/akhound-sadegh24a.html.

Tara Akhound-Sadegh, Jungyoon Lee, Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M. Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vf2GHcxzMV.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Denis Blessing, Lorenz Richter, Julius Berner, Egor Malitskiy, and Gerhard Neumann. Bridge matching sampler: Scalable sampling via generalized fixed-point diffusion matching. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gUbNUNuJbO. Code: https://github.com/DenisBless/BridgeMatchingSampler.

I Buch, M J Harvey, T Giorgino, D P Anderson, and G De Fabritiis. High-throughput all-atom molecular dynamics simulations using distributed computing. *J Chem Inf Model*, 50(3):397–403, Mar 2010. ISSN 1549-960X (Electronic); 1549-9596 (Linking). doi: 10.1021/ci900455r.

Mouyang Cheng, Denis Blessing, Botao Yu, Gerhard Neumann, Mingda Li, Carles Domingo-Enrich, and Yuanqi Du. Atlas: A foundation neural sampler for amorphous materials, 2026. URL https://arxiv.org/abs/2607.19198.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, CA, 2 edition, 2001.

Aaron J Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S. Levine, Brandon M Wood, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22204–22237. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/havens25a.html.

Aaron J Havens, Brian Karrer, and Neta Shaul. Flow sampling : Learning to sample from unnormalized densities via denoising conditional processes. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YlcyOMTPNl.

Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching. *Biometrika*, 112(4):asaf048, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

John David Jackson. *Classical Electrodynamics*. Wiley, New York, NY, 3rd edition, 1999. ISBN 9780471309321. URL https://cds.cern.ch/record/490457.

C. Jarzynski. Targeted free energy perturbation. *Physical Review E*, 65(4):046122, April 2002. doi: 10.1103/PhysRevE.65.046122. URL https://doi.org/10.1103/PhysRevE.65.046122.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359. URL https://doi.org/10.1137/S0036141096303359.

Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kohler20a.html.

Benedict Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Springer, Cham, 2015.

Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. doi: 10.1126/science.1208351. URL https://www.science.org/doi/abs/10.1126/science.1208351.

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

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky T. Q. Chen. Adjoint schrödinger bridge sampler. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rMhQBlhh4c.

Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, NY, 2001.

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.

Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R. Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106(45):19011–19016, 2009.

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. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71 (Volume 71, 2020):361–390, 2020. ISSN 1545-1593. doi: https://doi.org/10.1146/annurev-physchem-042018-052331. URL https://www.annualreviews.org/content/journals/10.1146/annurev-physchem-042018-052331.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. 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. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.

G.A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. Texts in Applied Mathematics. Springer New York, 2014. ISBN 9781493913237. URL https://books.google.com/books?id=8213BQAAQBAJ.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015.

F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Springer International Publishing, 2015. ISBN 9783319208282. URL https://books.google.com/books?id=UOHHCgAAQBAJ.

Maximilian Schebek, Frank Noé, and Jutta Rogal. Scalable boltzmann generators for equilibrium sampling of large-scale materials. *Nature Communications*, 17(1):5010, 2026. doi: 10.1038/s41467-026-73900-9. URL https://doi.org/10.1038/s41467-026-73900-9.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Charlie B. Tan, Joey Bose, Chen Lin, Leon Klein, Michael M. Bronstein, and Alexander Tong. Scalable equilibrium sampling with sequential boltzmann generators. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58467–58498. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/tan25a.html.

Luke Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1728, 1994. URL https://api.semanticscholar.org/CorpusID:13369978.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y.

APPENDIX CONTENTS

A Method **13**

A.1 Denoiser Response to Endpoint Transport 13

A.2 Kabsch Alignment and the Induced Coupling 15

A.3 Stein's Identity and the Mean Estimator 17

A.4 Covariance Estimators under Independent Coupling 18

A.5 Covariance Estimator under a General Coupling 21

A.6 Extended WASABI Algorithm 24

B Experiment Details **25**

B.1 DW-4 Evaluation 25

B.2 DW-4 and LJ-13 potentials 25

B.3 Alanine-dipeptide potential and training setup 25

B.4 Alanine-dipeptide ablations and computational cost 28

C Related work **31**

A METHOD

A.1 DENOISER RESPONSE TO ENDPOINT TRANSPORT

Proposition A.1 (Denoiser response to endpoint transport)**.** *Let $(X_0, X_1) \sim \gamma_n$ have marginals p_0, q_n, and let $v : \mathbb{R}^d \to \mathbb{R}^d$ be measurable, with $\mathbb{E}[\|X_0\|^2 + \|X_1\|^2 + \|v(X_1)\|^2] < \infty$. Define $X_1^\epsilon = X_1 + \epsilon v(X_1)$, whose distribution is $q_{n,\epsilon} = (\mathrm{id} + \epsilon v)_{\#} q_n$.*

For fixed $t \in (0,1)$, let $h_t = B\sqrt{t(1-t)}$, $B > 0$, and let $Z \sim \mathcal{N}(0, I_d)$ be independent of (X_0, X_1). Define $Y_t^\epsilon = (1-t)X_0 + tX_1^\epsilon + h_t Z$ and $m_{t,\epsilon}^{(n)}(y) = \mathbb{E}[X_1^\epsilon \mid Y_t^\epsilon = y]$. Then

$$\partial_\epsilon m_{t,\epsilon}^{(n)}(y)\Big|_{\epsilon=0} = \mathbb{E}[v(X_1) \mid Y_t = y] + \frac{t}{h_t}\,\mathrm{Cov}(X_1, Z \cdot v(X_1) \mid Y_t = y)\,, \tag{20}$$

where $Y_t = Y_t^0$.

Proof. Fix $t \in (0,1)$ and $y \in \mathbb{R}^d$. Conditional on $X_0 = x_0$ and $X_1 = x_1$, the perturbed interpolant has Gaussian density

$$K_\epsilon(y \mid x_0, x_1) = \frac{1}{(2\pi h_t^2)^{d/2}} \exp\!\left(-\frac{\|y - (1-t)x_0 - t[x_1 + \epsilon v(x_1)]\|^2}{2h_t^2}\right).$$

Consequently,

$$\begin{aligned} m_{t,\epsilon}^{(n)}(y) &= \mathbb{E}[X_1^\epsilon \mid Y_t^\epsilon = y] \\ &= \frac{\displaystyle\int [x_1 + \epsilon v(x_1)] K_\epsilon(y \mid x_0, x_1)\, d\gamma_n(x_0, x_1)}{\displaystyle\int K_\epsilon(y \mid x_0, x_1)\, d\gamma_n(x_0, x_1)} =: \frac{N(\epsilon)}{D(\epsilon)}. \end{aligned}$$

The denominator is strictly positive because the Gaussian kernel is positive.

To differentiate under these integrals, we establish integrable bounds for their derivatives. Differentiating the Gaussian gives

$$\partial_\epsilon K_\epsilon = \frac{t}{h_t^2}\big[y - (1-t)x_0 - t(x_1 + \epsilon v(x_1))\big] \cdot v(x_1)\, K_\epsilon.$$

By the product rule,

$$\begin{aligned}\partial_\epsilon([x_1+\epsilon v(x_1)]K_\epsilon) &= v(x_1)K_\epsilon + [x_1+\epsilon v(x_1)]\partial_\epsilon K_\epsilon \\ &= v(x_1)K_\epsilon + \frac{t}{h_t^2}[x_1+\epsilon v(x_1)] \\ &\quad \times \Big([y-(1-t)x_0 - t(x_1+\epsilon v(x_1))]\cdot v(x_1)\Big)K_\epsilon.\end{aligned}$$

Taking norms, applying Cauchy–Schwarz to the inner product, and using $|\epsilon| \le 1$, we obtain

$$\begin{aligned}&\|\partial_\epsilon([x_1+\epsilon v(x_1)]K_\epsilon)\| \\ &\le \|v(x_1)\|K_\epsilon + \frac{t}{h_t^2}\big(\|x_1\|+\|v(x_1)\|\big)\|v(x_1)\| \\ &\quad \times \|y-(1-t)x_0 - t(x_1+\epsilon v(x_1))\|K_\epsilon \\ &\le C_t\left(\|v(x_1)\| + \|x_1\|\|v(x_1)\| + \|v(x_1)\|^2\right).\end{aligned}$$

The last inequality follows because a Gaussian density is bounded, as is its product with the distance from its mean. Applying $ab \le (a^2+b^2)/2$ gives

$$\|v(x_1)\| \le \frac{1+\|v(x_1)\|^2}{2}, \qquad \|x_1\|\|v(x_1)\| \le \frac{\|x_1\|^2+\|v(x_1)\|^2}{2}.$$

Thus, after enlarging C_t,

$$\|\partial_\epsilon([x_1+\epsilon v(x_1)]K_\epsilon)\| \le C_t\big(1+\|x_1\|^2+\|v(x_1)\|^2\big).$$

Similarly, $|\partial_\epsilon K_\epsilon| \le C_t\|v(x_1)\|$. These bounds are integrable under the assumed second moments, so dominated convergence justifies differentiation under both integrals.

Evaluating the derivatives at $\epsilon = 0$, and abbreviating the kernel arguments and $d\gamma_n = d\gamma_n(x_0, x_1)$, gives

$$\begin{aligned}N'(0) &= \int v(x_1)K_0\, d\gamma_n \\ &\quad + \frac{t}{h_t^2}\int x_1\big([y-(1-t)x_0 - tx_1]\cdot v(x_1)\big)K_0\, d\gamma_n, \\ D'(0) &= \frac{t}{h_t^2}\int \big([y-(1-t)x_0 - tx_1]\cdot v(x_1)\big)K_0\, d\gamma_n.\end{aligned}$$

The conditional distribution of the endpoints given $Y_t = y$ is

$$\gamma_n(dx_0, dx_1 \mid Y_t = y) = \frac{K_0(y \mid x_0, x_1)}{D(0)}\, d\gamma_n(x_0, x_1).$$

Under this conditioning, $y-(1-t)X_0 - tX_1 = h_t Z$. Therefore,

$$\begin{aligned}\frac{N'(0)}{D(0)} &= \mathbb{E}[v(X_1) \mid Y_t = y] + \frac{t}{h_t}\mathbb{E}[X_1(Z\cdot v(X_1)) \mid Y_t = y], \\ \frac{D'(0)}{D(0)} &= \frac{t}{h_t}\mathbb{E}[Z\cdot v(X_1) \mid Y_t = y].\end{aligned}$$

Finally, applying the quotient rule and using $N(0)/D(0) = m_t^{(n)}(y)$,

$$\begin{aligned}\partial_\epsilon m_{t,\epsilon}^{(n)}(y)\Big|_0 &= \frac{N'(0)}{D(0)} - m_t^{(n)}(y)\frac{D'(0)}{D(0)} \\ &= \mathbb{E}[v(X_1) \mid Y_t = y] \\ &\quad + \frac{t}{h_t}\Big(\mathbb{E}[X_1(Z\cdot v(X_1)) \mid Y_t = y] \\ &\qquad - \mathbb{E}[X_1 \mid Y_t = y]\,\mathbb{E}[Z\cdot v(X_1) \mid Y_t = y]\Big) \\ &= \mathbb{E}[v(X_1) \mid Y_t = y] + \frac{t}{h_t}\operatorname{Cov}(X_1, Z\cdot v(X_1) \mid Y_t = y),\end{aligned}$$

which proves the claim. □

Taking $\gamma_n = p_0 \otimes q_n$ and $v = -\nabla_{W_2}\mathcal{F}[q_n]$ recovers the main-text theorem whenever the stated moment condition holds.

A.2 KABSCH ALIGNMENT AND THE INDUCED COUPLING

Let $g = (g_1, \dots, g_N)$ and $x = (x_1, \dots, x_N)$ be centered configurations in $\mathbb{R}^{3N}$. Kabsch alignment finds a rotation

$$R^\star(g, x) \in \operatorname*{arg\,min}_{R \in SO(3)} \frac{1}{2} \sum_{i=1}^{N} \|Rg_i - x_i\|^2,$$

where $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^\top R = I,\ \det R = 1\}$. We define the aligned configuration by $A(g, x) = (R^\star(g, x)g_i)_{i=1}^N$.

To construct the aligned coupling, we independently draw $G \sim p_0$ and $X_1 \sim q_n$, then set $X_0 = A(G, X_1)$. In our ALA2 experiments, p_0 is the empirical distribution of configurations obtained from high-temperature molecular dynamics. The resulting pair (X_0, X_1) defines the coupling γ_n, and the bridge observation is

$$Y_t = (1 - t)X_0 + tX_1 + h_t Z, \qquad Z \sim \mathcal{N}(0, I),$$

with Z independent of G, X_1. Here p_0 describes the unaligned seed G; the aligned sample X_0 need not have the same marginal distribution.

In the endpoint-transport derivative, the already aligned source X_0 is held fixed while X_1 is perturbed; alignment is not recomputed during that infinitesimal perturbation.

To derive the Jacobian $D_x A(g, x)$, we first review basic facts about the Lie algebra of $SO(3)$.

Lemma A.2 (Derivative of a rotation)**.** *Let $R : (-\delta, \delta) \to SO(3)$ be differentiable. Then there exists $\Omega \in T_I SO(3)$ such that*

$$R'(0) = \Omega R(0), \qquad T_I SO(3) = \{\Omega \in \mathbb{R}^{3\times 3} : \Omega^\top = -\Omega\}.$$

Proof. Differentiating $R(\epsilon)R(\epsilon)^\top = I$ at $\epsilon = 0$ gives

$$R'(0)R(0)^\top + R(0)R'(0)^\top = 0.$$

Thus $\Omega := R'(0)R(0)^\top$ is skew-symmetric and belongs to $T_I SO(3)$. Multiplying by $R(0)$ yields $R'(0) = \Omega R(0)$. □

Every $\Omega \in T_I SO(3)$ can be written uniquely as

$$\Omega = [\omega]_\times := \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \qquad \omega \in \mathbb{R}^3.$$

Direct multiplication shows that $\Omega a = \omega \times a$. Consequently, for $a_i(\epsilon) = R(\epsilon)g_i$, the preceding lemma gives

$$a_i'(0) = \Omega R(0) g_i = \omega \times a_i(0).$$

Lemma A.3 (Optimality condition for alignment)**.** *Let $R^\star$ minimize $\frac{1}{2}\sum_i \|Rg_i - x_i\|^2$ over $SO(3)$, and set $a_i = R^\star g_i$. Then*

$$\sum_i a_i \times x_i = 0.$$

Proof. Fix $u \in \mathbb{R}^3$ and consider the rotated positions $b_i(s) = \exp(s[u]_\times)a_i$, where exp denotes the matrix exponential. These satisfy $b_i(0) = a_i$ and $b_i'(0) = u \times a_i$.

Since a is optimally aligned, $E(s) := \frac{1}{2}\sum_i \|b_i(s) - x_i\|^2$ has a minimum at $s = 0$. Therefore,

$$\begin{aligned} 0 = E'(0) &= \sum_i (a_i - x_i) \cdot (u \times a_i) \\ &= -\sum_i x_i \cdot (u \times a_i) \\ &= -u \cdot \sum_i a_i \times x_i. \end{aligned}$$

As this holds for every $u \in \mathbb{R}^3$, we conclude that

$$\sum_i a_i \times x_i = 0.$$

□

Theorem A.4 (Jacobian of Kabsch alignment)**.** *Fix centered configurations g, x, and suppose the optimal rotation $R^\star(g,\cdot)$ is unique and differentiable in a neighborhood of x. Write $a_i = R^\star(g,x)g_i$, and assume that*

$$H := \sum_{i=1}^{N} \left[(a_i^\top x_i)I - a_i x_i^\top\right]$$

is invertible. Then, for any centered perturbation δx,

$$\left[D_x A(g,x)\,\delta x\right]_i = \left(H^{-1}\sum_{j=1}^{N} a_j \times \delta x_j\right) \times a_i.$$

Equivalently, defining $T\omega = (\omega \times a_i)_{i=1}^N$, we have $D_x A(g,x) = TH^{-1}T^\top$.

Proof. Let $R(\epsilon) = R^\star(g, x + \epsilon\delta x)$ and $a_i(\epsilon) = R(\epsilon)g_i$. By the rotation derivative lemma, there exists $\omega \in \mathbb{R}^3$ such that

$$a_i'(0) = \omega \times a_i.$$

For each sufficiently small ϵ, the configuration $a(\epsilon)$ is optimally aligned to $x + \epsilon\delta x$. The optimality condition therefore gives

$$\sum_i a_i(\epsilon) \times (x_i + \epsilon\delta x_i) = 0.$$

Differentiating at $\epsilon = 0$ yields

$$0 = \sum_i \left[(\omega \times a_i) \times x_i + a_i \times \delta x_i\right].$$

Using the BAC–CAB identity Jackson (1999),

$$\begin{aligned}(\omega \times a_i) \times x_i &= a_i(x_i^\top \omega) - (a_i^\top x_i)\omega \\ &= \left[a_i x_i^\top - (a_i^\top x_i)I\right]\omega.\end{aligned}$$

Substituting and rearranging, we obtain

$$H\omega = \sum_i a_i \times \delta x_i.$$

Since H is invertible,

$$\omega = H^{-1}\sum_i a_i \times \delta x_i.$$

Substituting into $[D_x A(g,x)\,\delta x]_i = a_i'(0) = \omega \times a_i$ proves the stated formula.

Finally, the scalar triple-product identity gives

$$\langle T\omega, \delta x\rangle = \sum_i (\omega \times a_i) \cdot \delta x_i = \omega \cdot \sum_i a_i \times \delta x_i.$$

Thus $T^\top \delta x = \sum_i a_i \times \delta x_i$, and we conclude that

$$D_x A(g,x) = TH^{-1}T^\top.$$

□

Corollary A.5 (Positivity of the bridge Jacobian)**.** *Under the assumptions of the preceding theorem, $H \succ 0$ and $D_x A(g,x) = TH^{-1}T^\top \succeq 0$. Consequently, for $0 < t < 1$,*

$$J_t = tI + (1-t)D_x A(g,x) \succeq tI \succ 0.$$

Proof. The optimality condition implies that $\sum_i a_i x_i^\top$ is symmetric, so H is symmetric. For any $u \in \mathbb{R}^3$, let $\Omega v = u \times v$ and consider $E(s) = \frac{1}{2}\sum_i \|e^{s\Omega}a_i - x_i\|^2$. The alignment minimum at $s = 0$ gives

$$0 \le E''(0) = \sum_i \left[(a_i^\top x_i)\|u\|^2 - (u^\top a_i)(x_i^\top u)\right] = u^\top H u.$$

Thus $H \succeq 0$; its assumed invertibility gives $H \succ 0$. The remaining claims follow from $D_x A = TH^{-1}T^\top$. □

A.3 Stein's Identity and the Mean Estimator

Proposition A.6 (Mean estimator under the aligned coupling)**.** *Let $G \sim p_0$, $X_1 \sim q_n$, and $Z \sim \mathcal{N}(0, I)$ be independent. For fixed $t \in (0,1)$ and $h_t > 0$, define*

$$Y_t = (1-t)A(G, X_1) + tX_1 + h_t Z, \qquad J_t = tI + (1-t)D_{X_1}A(G, X_1).$$

Assume q_n has a differentiable positive density, $A(G,\cdot)$ is differentiable almost surely, and the integrability and vanishing-boundary conditions required for the following applications of Stein's identity hold. Then

$$\begin{aligned}\mathcal{M}_t[q_n](y) &:= \mathbb{E}[\nabla \log \pi(X_1) - \nabla \log q_n(X_1) \,|\, Y_t = y] \\ &= \mathbb{E}\left[\nabla \log \pi(X_1) + \frac{J_t^\top Z}{h_t} \,\middle|\, Y_t = y\right]\end{aligned}$$

for almost every y under the distribution of Y_t.

Proof. Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be a smooth, compactly supported test function. Holding G and Z fixed, we apply Stein's identity in X_1 to

$$f(x) = \varphi((1-t)A(G,x) + tx + h_t Z)\,.$$

Stein's identity states that

$$\mathbb{E}_{X_1}[f(X_1)\nabla \log q_n(X_1) + \nabla_x f(X_1)] = 0.$$

By the chain rule,

$$\nabla_x f(X_1) = \underbrace{[tI + (1-t)D_{X_1}A(G, X_1)]^\top}_{J_t^\top} \nabla\varphi(Y_t).$$

Substituting and averaging over G, Z, we obtain

$$\mathbb{E}\big[\varphi(Y_t)\nabla \log q_n(X_1) + J_t^\top \nabla\varphi(Y_t)\big] = 0.$$

Next, applying Gaussian Stein's identity in Z, with G, X_1 fixed, gives

$$\mathbb{E}_Z[Z\varphi(Y_t) \mid G, X_1] = h_t\, \mathbb{E}_Z[\nabla\varphi(Y_t) \mid G, X_1]\,.$$

Since J_t does not depend on Z, multiplying by $J_t^\top / h_t$ and averaging yields

$$\mathbb{E}[J_t^\top \nabla\varphi(Y_t)] = \frac{1}{h_t}\mathbb{E}[\varphi(Y_t)J_t^\top Z].$$

Combining the two identities,

$$\mathbb{E}\left[\varphi(Y_t)\left(\nabla \log q_n(X_1) + \frac{J_t^\top Z}{h_t}\right)\right] = 0.$$

As this holds for every such φ, we conclude that

$$\mathbb{E}[\nabla \log q_n(X_1) \mid Y_t = y] = -\frac{1}{h_t}\mathbb{E}[J_t^\top Z \mid Y_t = y].$$

Substitution into the definition of $\mathcal{M}_t[q_n]$ proves the result. □

For independent endpoints, $A(G,x) = G$, so $J_t = tI$, recovering the estimator $\nabla \log \pi(X_1) + tZ/h_t$. For Kabsch alignment, the preceding Jacobian theorem gives $J_t = tI + (1-t)TH^{-1}T^\top$.

A.4 Covariance Estimators under Independent Coupling

Fix $t \in (0,1)$ with $h_t > 0$. Suppose $X_1 \sim q_n$, $X_0 \sim p_0$ and $Z \sim \mathcal{N}(0, I_d)$ are independent, and write $s_{q_n} = \nabla \log q_n$ and $s_\pi = \nabla \log \pi$. Let ϕ be the standard Gaussian density on $\mathbb{R}^d$, and set

$$Y = tX_1 + (1-t)X_0 + h_t Z, \qquad m(y) = \mathbb{E}[X_1 \mid Y = y], \tag{21}$$

$$F(y,x) = f_\theta(y) \cdot \big(x - m(y)\big), \qquad a(x) = \frac{h_t}{2t} s_\pi(x). \tag{22}$$

Here $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is the vector field being fitted. Using the main-text definition of $\mathcal{C}_t[q_n]$, we estimate its expected inner product with f_θ:

$$\begin{aligned}\mathcal{C}_t[f_\theta] &:= \mathbb{E}[f_\theta(Y) \cdot \mathcal{C}_t[q_n](Y)] \\ &= \frac{t}{h_t}\mathbb{E}\big[f_\theta(Y) \cdot \mathrm{Cov}\big(X_1, Z \cdot \big(s_\pi(X_1) - s_{q_n}(X_1)\big) \mid Y\big)\big]\,. \end{aligned} \tag{23}$$

All expectations below average over the sampled variables unless conditioning is shown. We assume sufficient differentiability, absolute integrability of the expectations involved, and vanishing boundary terms in integration by parts. The mean m and the covariance are defined under the original Gaussian bridge law.

Theorem A.7 (Unbiased covariance estimator)**.** *The function*

$$\widetilde{\phi}(z \mid x) = \frac{\|z + a(x)\|^2}{d + \|a(x)\|^2}\,\phi(z) \tag{24}$$

is a probability density. For any coupling of $(Z, \widetilde{Z})$ with conditional marginals ϕ and $\widetilde{\phi}(\cdot \mid X_1)$ given (X_0, X_1), set $\widetilde{Y} = tX_1 + (1-t)X_0 + h_t\widetilde{Z}$. Then

$$\begin{aligned}\widehat{\mathcal{C}}_t[f_\theta] = \frac{t}{h_t} f_\theta(Y) \cdot Z \\ + \frac{t^2}{h_t^2}\big(d + \|a(X_1)\|^2\big)\big[F(\widetilde{Y}, X_1) - F(Y, X_1)\big]\end{aligned} \tag{25}$$

satisfies $\mathbb{E}[\widehat{\mathcal{C}}_t[f_\theta]] = \mathcal{C}_t[f_\theta]$.

Proof. Expanding out the covariance term we get

$$\begin{aligned}\mathcal{C}_t[f_\theta] &= \frac{t}{h_t}\mathbb{E}\left[\underbrace{f_\theta(Y) \cdot (X_1 - m(Y))}_{F(Y,X_1)}\big(Z \cdot (s_\pi(X_1) - s_{q_n}(X_1))\big)\right] \\ &= \frac{t}{h_t}\mathbb{E}[FZ \cdot s_\pi(X_1)] - \frac{t}{h_t}\mathbb{E}[FZ \cdot s_{q_n}(X_1)].\end{aligned} \tag{26}$$

For the term involving the unknown endpoint score, perform integration by parts with X_0 and Z fixed when differentiating in X_1. Here ∇_{X_1} includes the dependence through Y, while $\partial_{X_1} F(Y, X_1)$ holds Y fixed.

$$\begin{aligned}\mathbb{E}[F(Y,X_1)Z \cdot s_{q_n}(X_1)] &= -\mathbb{E}[\nabla_{X_1} \cdot (F(Y,X_1)Z)] && (27)\\ &= -\mathbb{E}[\langle \nabla_{X_1} F(Y,X_1), Z\rangle] && (28)\\ &= -\mathbb{E}[\langle \partial_{X_1} F(Y,X_1) + t\partial_Y F(Y,X_1), Z\rangle] && (29)\\ &= -\mathbb{E}[\langle f_\theta(Y) + t\partial_Y[f_\theta(Y) \cdot (X_1 - m(Y))], Z\rangle] && (30)\\ &= -\mathbb{E}[\langle f_\theta(Y), Z\rangle && (31)\\ &\quad + t\langle \nabla_Y[f_\theta(Y) \cdot (X_1 - m(Y))], Z\rangle] && (32)\\ &= -\mathbb{E}[\langle f_\theta(Y), Z\rangle && (33)\\ &\quad + \frac{t}{h_t}\langle \nabla_Z[F(Y,X_1)], Z\rangle] \qquad \text{using } \nabla_Z F = h_t \nabla_Y F && (34)\end{aligned}$$

For the next step, hold X_0, X_1 fixed and write $F(Z)$ for $F(Y(t, X_1, X_0, Z), X_1)$. Recall Gaussian integration by parts tells us that $\mathbb{E}[Z_i F(Z)] = \mathbb{E}[\nabla_{Z_i} F(Z)]$ so

$$\mathbb{E}[Z_i^2 F(Z)] = \mathbb{E}[\nabla_{Z_i}[Z_i F(Z)]] \tag{35}$$
$$= \mathbb{E}[F(Z) + Z_i \nabla_{Z_i} F(Z)] \tag{36}$$

thus

$$\mathbb{E}[\langle Z, \nabla_Z F\rangle] = \mathbb{E}[(\|Z\|^2 - d)F(Z)] \tag{37}$$

Substituting this above we get

$$\mathbb{E}[F(Y, X_1) Z \cdot s_{q_n}(X_1)] = -\mathbb{E}[\langle f_\theta(Y), Z\rangle \tag{38}$$
$$+ \frac{t}{h_t}(\|Z\|^2 - d)F(Y, X_1)] \tag{39}$$

Therefore, combining with the target-score term gives

$$\mathcal{C}_t[f_\theta] = \frac{t}{h_t}\mathbb{E}[f_\theta(Y) \cdot Z]$$
$$+ \frac{t}{h_t}\mathbb{E}\left[F(Y, X_1) \underbrace{\left(Z \cdot s_\pi(X_1) + \frac{t}{h_t}\left(\|Z\|^2 - d\right)\right)}_{\text{High variance term}}\right].$$

For $t > 0$ and $h_t > 0$, write $s = s_\pi(X_1)$. Completing the square gives

$$Z \cdot s + \frac{t}{h_t}(\|Z\|^2 - d) = \frac{t}{h_t}\left\|Z + \frac{h_t}{2t}s\right\|^2 - \frac{h_t}{4t}\|s\|^2 - \frac{t}{h_t}d \tag{40}$$

Substituting this above gives

$$\mathcal{C}_t[f_\theta] = \frac{t}{h_t}\mathbb{E}[f_\theta(Y) \cdot Z]$$
$$+ \frac{t^2}{h_t^2}\mathbb{E}\left[F(Y, X_1)\left\|Z + \frac{h_t}{2t}s_\pi(X_1)\right\|^2\right]$$
$$- \frac{1}{4}\mathbb{E}\left[F(Y, X_1)\|s_\pi(X_1)\|^2\right] - \frac{t^2 d}{h_t^2}\mathbb{E}[F(Y, X_1)]\,. \tag{41}$$

Change of measure step Let ϕ be the standard Gaussian density on $\mathbb{R}^d$. For a fixed vector a,

$$\mathbb{E}_{Z\sim\phi}\left[\|Z + a\|^2\right] = d + \|a\|^2. \tag{42}$$

Now fix $X_0 = x_0$ and $X_1 = x_1$, and set

$$a(x_1) = \frac{h_t}{2t}s_\pi(x_1), \qquad \widetilde{\phi}(z \mid x_1) = \frac{\|z + a(x_1)\|^2}{d + \|a(x_1)\|^2}\,\phi(z). \tag{43}$$

The preceding Gaussian identity shows that $\widetilde{\phi}(\cdot \mid x_1)$ is a normalized probability density. Draw $\widetilde{Z} \mid X_1 \sim \widetilde{\phi}(\cdot \mid X_1)$ and define

$$\widetilde{Y} = tX_1 + (1 - t)X_0 + h_t\widetilde{Z}. \tag{44}$$

The endpoint variables retain their original independent laws $X_1 \sim q_n$ and $X_0 \sim p_0$; only the conditional noise law changes. The change of measure and the law of iterated expectation give

$$\mathbb{E}\left[F(Y, X_1)\left\|Z + \frac{h_t}{2t}s_\pi(X_1)\right\|^2\right]$$
$$= \mathbb{E}_{X_0, X_1}\left[\left(d + \|a(X_1)\|^2\right)\mathbb{E}_{\widetilde{Z}\sim\widetilde{\phi}(\cdot|X_1)}\left[F(\widetilde{Y}, X_1)\right]\right]$$
$$= \mathbb{E}\left[\left(d + \frac{h_t^2}{4t^2}\|s_\pi(X_1)\|^2\right)F(\widetilde{Y}, X_1)\right]. \tag{45}$$

The two subtracted terms from completing the square combine as

$$-\frac{1}{4}\mathbb{E}\big[F(Y,X_1)\|s_\pi(X_1)\|^2\big]-\frac{t^2 d}{h_t^2}\mathbb{E}[F(Y,X_1)]$$
$$=-\frac{t^2}{h_t^2}\mathbb{E}\big[\big(d+\|a(X_1)\|^2\big)F(Y,X_1)\big]\,. \tag{46}$$

Substituting both identities into the covariance expression therefore yields

$$\begin{aligned}\mathcal{C}_t[f_\theta] &= \frac{t}{h_t}\mathbb{E}[f_\theta(Y)\cdot Z]\\ &\quad+\frac{t^2}{h_t^2}\mathbb{E}\Big[\big(d+\|a(X_1)\|^2\big)F(\widetilde{Y},X_1)\Big]-\frac{t^2}{h_t^2}\mathbb{E}\big[\big(d+\|a(X_1)\|^2\big)F(Y,X_1)\big]\\ &= \frac{t}{h_t}\mathbb{E}[f_\theta(Y)\cdot Z]\\ &\quad+\frac{t^2}{h_t^2}\mathbb{E}\Big[\big(d+\|a(X_1)\|^2\big)\big(F(\widetilde{Y},X_1)-F(Y,X_1)\big)\Big]\,.\end{aligned} \tag{47}$$

This is exactly $\mathbb{E}[\widehat{\mathcal{C}}_t[f_\theta]]$ for the estimator in equation 25. Only the conditional noise marginals enter the two expectations, so the identity holds for any coupling with those marginals. □

Their coupling affects variance but not the expectation. It remains to draw from the tilted density. The following lemma gives an exact sampler.

Lemma A.8 (Sampling the tilted density)**.** *Fix $a\in\mathbb{R}^d$. Draw independent $G\sim\mathcal{N}(0,I_d)$ and $H_1,H_2\sim\mathcal{N}(0,1)$. Independently choose*

$$V=\begin{cases}G, & \text{with probability } \dfrac{\|a\|^2}{d+\|a\|^2},\\[2mm] \sqrt{1+\dfrac{H_1^2+H_2^2}{\|G\|^2}}\,G, & \text{otherwise},\end{cases} \tag{48}$$

then set

$$\widetilde{Z}=\begin{cases}V, & \text{with probability } \dfrac{1}{2}+\dfrac{V\cdot a}{\|V\|^2+\|a\|^2},\\[2mm] -V, & \text{otherwise}.\end{cases} \tag{49}$$

Then $\widetilde{Z}$ has density $\widetilde{\phi}(z\mid a)=\|z+a\|^2\phi(z)/(d+\|a\|^2)$.

Proof. Start by choosing a pair $\{z,-z\}$ using the symmetrized density below. Then conditional on $\{z,-z\}$ choose one of these, this corresponds to:

$$P(\text{choose } z\mid\pm z)=\frac{\widetilde{\phi}(z)}{\widetilde{\phi}(z)+\widetilde{\phi}(-z)} \tag{50}$$
$$=\frac{\|z+a\|^2}{\|z+a\|^2+\|z-a\|^2} \tag{51}$$
$$=\frac{1}{2}+\frac{z\cdot a}{\|z\|^2+\|a\|^2} \tag{52}$$

$$\frac{1}{2}\Big(\widetilde{\phi}(z\mid a)+\widetilde{\phi}(-z\mid a)\Big)=\frac{\|z\|^2+\|a\|^2}{d+\|a\|^2}\,\phi(z) \tag{53}$$

Now consider the symmetrized density $s(z\mid a)=\frac{1}{2}\Big(\widetilde{\phi}(z\mid a)+\widetilde{\phi}(-z\mid a)\Big)$. Note that

$$\begin{aligned}s(z\mid a)P(\text{choose } z\mid\pm z)&=\frac{\widetilde{\phi}(z\mid a)}{\widetilde{\phi}(z\mid a)+\widetilde{\phi}(-z\mid a)}\cdot\frac{1}{2}\Big(\widetilde{\phi}(z\mid a)+\widetilde{\phi}(-z\mid a)\Big)\\ &=\frac{1}{2}\widetilde{\phi}(z\mid a).\end{aligned} \tag{54}$$

But we can also output z by first drawing $-z$ and choosing its opposite. Adding both contributions, the output density at z is

$$\begin{aligned} &s(z \mid a)P(\text{choose } z \mid \pm z) + s(-z \mid a)P(\text{choose } z \mid \pm z) \\ &\quad = \frac{1}{2}\widetilde{\phi}(z \mid a) + \frac{1}{2}\widetilde{\phi}(z \mid a) = \widetilde{\phi}(z \mid a). \end{aligned} \tag{55}$$

Thus, it suffices to sample from the symmetrized density and then choose the sign with the probability above. For fixed a, write $g(z) = s(z \mid a)$, so

$$g(z) = \frac{\|z\|^2 + \|a\|^2}{d + \|a\|^2}\phi(z) \tag{56}$$

$$= \frac{\|a\|^2}{\|a\|^2 + d}\phi(z) + \frac{d}{\|a\|^2 + d}\frac{\|z\|^2}{d}\phi(z) \tag{57}$$

This is just a mixture of two densities $\phi(z)$ and $d^{-1}\|z\|^2\phi(z)$. The second is normalized since $\mathbb{E}_\phi[\|Z\|^2] = d$. Multiplying the Gaussian density by $\|z\|^2$ changes its radial density from one proportional to $r^{d-1}e^{-r^2/2}$ to one proportional to $r^{d+1}e^{-r^2/2}$. This is just the Gaussian radius in $d + 2$ dimensions.

Recall that sampling from a Gaussian may be done by sampling the radius with density proportional to $r^{d-1}e^{-r^2/2}$ and independently drawing a direction from $\mathcal{U}(S^{d-1})$, namely

$$G = RU, \qquad U \sim \mathcal{U}(S^{d-1}), \qquad R \perp U, \tag{58}$$

$$p_R(r) \propto r^{d-1}e^{-r^2/2}, \qquad r > 0. \tag{59}$$

Equivalently, given $G \sim \mathcal{N}(0, I_d)$, take $R = \|G\|$ and $U = G/\|G\|$. The Gaussian density depends only on the radius, and the volume element in polar coordinates contributes the factor r^{d-1}. This is why the direction is uniform and independent of the radius.

Multiplying the Gaussian density by $\|z\|^2$ changes only its radial density. So to sample from a density proportional to $\|z\|^2\phi(z)$, draw two independent one dimensional standard normals H_1, H_2, also independent of G, and do

$$\widetilde{R} = \sqrt{G_1^2 + \cdots + G_d^2 + H_1^2 + H_2^2}, \qquad H_1, H_2 \sim \mathcal{N}(0, 1), \tag{60}$$

$$\widetilde{G} = \widetilde{R}\,U = \sqrt{1 + \frac{H_1^2 + H_2^2}{\|G\|^2}}\,G. \tag{61}$$

Now $\widetilde{R}^2$ is the sum of $d + 2$ independent squared standard normals, so $\widetilde{R}$ has the required radial density. Also, U is independent of (R, H_1, H_2) and hence of $\widetilde{R}$. Thus $\widetilde{G}$ has density $d^{-1}\|z\|^2\phi(z)$. It is still a vector in $\mathbb{R}^d$; the extra two scalars change only its length.

Thus equation 48 draws V from $s(\cdot \mid a)$. The sign selection in equation 49 then gives a draw from $\widetilde{\phi}(\cdot \mid a)$, as shown above. □

For the estimator, take $a = a(X_1)$. Using $Z = G$ for the original Gaussian noise and this $\widetilde{Z}$ for the tilted noise gives a coupling with the required marginals. We can therefore substitute $Y = tX_1 + (1-t)X_0 + h_tG$ and $\widetilde{Y} = tX_1 + (1-t)X_0 + h_t\widetilde{Z}$ into equation 25. The construction is exact; it does not by itself guarantee lower variance.

A.5 Covariance Estimator under a General Coupling

Stein's identity under a general coupling. Let $G \sim p_0$, $X_1 \sim q_n$, and $Z \sim \mathcal{N}(0, I_d)$ be independent. For fixed $t \in (0, 1)$ and $h_t > 0$, define

$$Y_t = (1-t)A(G, X_1) + tX_1 + h_tZ, \qquad J_t = tI + (1-t)D_{X_1}A(G, X_1).$$

Write $s_{q_n} = \nabla \log q_n$ and $F(y, x) = f_\theta(y) \cdot (x - m(y))$, where $m(y) = \mathbb{E}[X_1 \mid Y_t = y]$. We assume sufficient smoothness and integrability for the following integrations by parts, with vanishing boundary terms.

Holding G, Z fixed, we apply Stein's identity in X_1 to the vector field $F(Y_t, X_1)Z$. The chain rule gives

$$\nabla_{X_1} F(Y_t, X_1) = f_\theta(Y_t) + J_t^\top \nabla_y F(Y_t, X_1),$$

where the first term comes from differentiating the explicit X_1 in F, and the second accounts for its dependence through Y_t. Therefore,

$$\begin{aligned}\mathbb{E}[F(Y_t, X_1)Z \cdot s_{q_n}(X_1)] &= -\mathbb{E}[Z \cdot \nabla_{X_1} F(Y_t, X_1)] \\ &= -\mathbb{E}[f_\theta(Y_t) \cdot Z] - \mathbb{E}[\nabla_y F(Y_t, X_1) \cdot J_t Z] \\ &= -\mathbb{E}[f_\theta(Y_t) \cdot Z] - \frac{1}{h_t}\mathbb{E}[\nabla_Z F(Y_t, X_1) \cdot J_t Z],\end{aligned}$$

using $\nabla_Z F(Y_t, X_1) = h_t \nabla_y F(Y_t, X_1)$.

Next, hold G, X_1 fixed and apply Gaussian Stein's identity to $F(Y_t, X_1)J_t Z$. Since J_t does not depend on Z, the product rule yields

$$\begin{aligned}\mathbb{E}[F(Y_t, X_1)Z^\top J_t Z] &= \mathbb{E}\big[\nabla_Z \cdot \big(F(Y_t, X_1)J_t Z\big)\big] \\ &= \mathbb{E}[\nabla_Z F(Y_t, X_1) \cdot J_t Z + F(Y_t, X_1) \operatorname{tr} J_t]\,.\end{aligned}$$

Rearranging and substituting into the preceding expression, we obtain

$$\begin{aligned}\mathbb{E}[F(Y_t, X_1)Z \cdot s_{q_n}(X_1)] = &-\mathbb{E}[f_\theta(Y_t) \cdot Z] \\ &- \frac{1}{h_t}\mathbb{E}\big[F(Y_t, X_1)\big(Z^\top J_t Z - \operatorname{tr} J_t\big)\big]\,.\end{aligned} \tag{62}$$

For independent coupling, $A(G, x) = G$ and $J_t = tI$, recovering the factor $t(\|Z\|^2 - d)$ from the preceding proof.

Completing the square and changing measure. We retain the covariance convention from the main text:

$$\mathcal{C}_t[f_\theta] := \frac{t}{h_t}\mathbb{E}\big[F(Y_t, X_1)Z \cdot \big(s_\pi(X_1) - s_{q_n}(X_1)\big)\big]\,. \tag{63}$$

The following construction requires the symmetric part $S_t = (J_t + J_t^\top)/2$ to be positive definite almost surely, in addition to the integration-by-parts and integrability assumptions above. This holds for independent coupling and for nondegenerate Kabsch alignment (Corollary A.5). The mean m remains the conditional mean under the original Gaussian bridge.

Theorem A.9 (Covariance estimator under a general coupling)**.** *Define*

$$a = \frac{h_t}{2} S_t^{-1} s_\pi(X_1), \qquad \kappa = \operatorname{tr} S_t + a^\top S_t a.$$

Let ϕ denote the standard Gaussian density. Conditional on G, X_1, the function

$$\widetilde{\phi}(z \mid G, X_1) = \frac{(z + a)^\top S_t (z + a)}{\kappa}\,\phi(z) \tag{64}$$

is a probability density. For any coupling of $Z, \widetilde{Z}$ with conditional marginals $\phi, \widetilde{\phi}$ given G, X_1, set $\widetilde{Y}_t = (1 - t)A(G, X_1) + tX_1 + h_t\widetilde{Z}$. Then

$$\widehat{\mathcal{C}}_t[f_\theta] = \frac{t}{h_t} f_\theta(Y_t) \cdot Z + \frac{t\kappa}{h_t^2}\big[F(\widetilde{Y}_t, X_1) - F(Y_t, X_1)\big] \tag{65}$$

satisfies $\mathbb{E}[\widehat{\mathcal{C}}_t[f_\theta]] = \mathcal{C}_t[f_\theta]$.

Proof. Substituting equation 62 into equation 63, and using $Z^\top J_t Z = Z^\top S_t Z$ and $\operatorname{tr} J_t = \operatorname{tr} S_t$, gives

$$\begin{aligned}\mathcal{C}_t[f_\theta] = &\frac{t}{h_t}\mathbb{E}[f_\theta(Y_t) \cdot Z] \\ &+ \frac{t}{h_t^2}\mathbb{E}\big[F(Y_t, X_1)\,\big(Z^\top S_t Z + h_t Z \cdot s_\pi(X_1) - \operatorname{tr} S_t\big)\big]\,.\end{aligned} \tag{66}$$

Since $2S_t a = h_t s_\pi(X_1)$, completing the square yields

$$Z^\top S_t Z + h_t Z \cdot s_\pi(X_1) - \operatorname{tr} S_t = (Z+a)^\top S_t (Z+a) - \kappa.$$

Conditional on G, X_1, the matrix S_t and vector a are fixed, and

$$\mathbb{E}_Z[(Z+a)^\top S_t (Z+a)] = \operatorname{tr} S_t + a^\top S_t a = \kappa.$$

Together with $S_t \succ 0$, this establishes that equation 64 is a probability density. Changing measure conditional on G, X_1, then averaging, gives

$$\mathbb{E}\big[F(Y_t, X_1)(Z+a)^\top S_t (Z+a)\big] = \mathbb{E}[\kappa F(\widetilde{Y}_t, X_1)].$$

Combining these identities,

$$\begin{aligned}\mathcal{C}_t[f_\theta] = \frac{t}{h_t}\mathbb{E}[f_\theta(Y_t)\cdot Z] \\ + \frac{t}{h_t^2}\mathbb{E}\Big[\kappa\big(F(\widetilde{Y}_t, X_1) - F(Y_t, X_1)\big)\Big],\end{aligned}$$

which is the expectation of equation 65. □

Lemma A.10 (Sampling a quadratic Gaussian tilt)**.** *Fix $S \succ 0$ and $a \in \mathbb{R}^d$, and write $S = Q\operatorname{diag}(\lambda_1, \dots, \lambda_d)Q^\top$, $b = Q^\top a$, and $\kappa = \operatorname{tr} S + a^\top S a$. Choose an index $I = i$ with probability $w_i = \lambda_i(1+b_i^2)/\kappa$. Conditional on $I = i$, draw U_i from the one-dimensional density*

$$\frac{(u+b_i)^2}{1+b_i^2}\,\phi_1(u),$$

using Lemma A.8 in dimension one. Draw the remaining coordinates independently from $\mathcal{N}(0,1)$ and independently of U_i. Then $\widetilde{Z} = QU$ has density $\kappa^{-1}(z+a)^\top S(z+a)\phi(z)$.

Proof. The weights sum to one because $\sum_i \lambda_i(1+b_i^2) = \kappa$. The mixture density of U is

$$\sum_i w_i \frac{(u_i+b_i)^2}{1+b_i^2}\prod_j \phi_1(u_j) = \frac{\sum_i \lambda_i (u_i+b_i)^2}{\kappa}\,\phi(u).$$

The orthogonal transformation $z = Qu$ preserves the standard Gaussian density and turns the numerator into $(z+a)^\top S(z+a)$. □

For independent coupling, $S_t = tI$, so $a = h_t s_\pi(X_1)/(2t)$ and $\kappa = t(d + \|a\|^2)$, recovering equation 25. For Kabsch alignment, $S_t = J_t = tI + (1-t)TH^{-1}T^\top \succ 0$. The Stein identity above also holds for other differentiable couplings, but the positive-definiteness condition is required for this particular completed-square tilt.

A.6 EXTENDED WASABI ALGORITHM

Algorithm 2 WASABI with replay, optional alignment, and annealing

Inputs: Energy U; sampleable prior p_0; K initial endpoints; refresh count $R \le K$; noise scale B; iterations N; schedules $\eta_t, \lambda_t, \beta_n$; coupling map A.

1: Initialize denoiser parameters α and correction parameters θ.
2: Initialize the ring buffer $\mathcal{D}_0 = \{(x^{(k)}, \nabla U(x^{(k)}))\}_{k=1}^{K}$.
3: Set $A(g, x) = g$ without alignment and $\beta_n = 1$ without annealing.
4: **for** $n = 0, \dots, N-1$ **do**
5: **Set target score:** $s_{\pi_n}(x) = -\beta_n \nabla U(x)$.
6: **Fit denoiser:** sample minibatches of $(X_1, \nabla U(X_1)) \sim \mathcal{D}_n$, independent $G \sim p_0$, bridge times t, and $Z \sim \mathcal{N}(0, I)$. Construct

$$X_0 = A(G, X_1), \quad h_t = B\sqrt{t(1-t)},$$
$$Y_t = (1-t)X_0 + tX_1 + h_t Z, \text{ and fit}$$
$$\alpha_n \approx \arg\min_{\alpha} \mathbb{E}_{\mathcal{D}_n}\left[\|m_\alpha(Y_t, t) - X_1\|^2\right].$$

7: **Fit correction:** freeze α_n; construct training pairs as above and compute

$$J_t = tI + (1-t)D_{X_1}A(G, X_1).$$

Using paired Gaussian and tilted noise, fit

$$\theta_n \approx \arg\min_{\theta} \mathbb{E}_{\mathcal{D}_n}\Bigg[\frac{1}{2}\|f_\theta(Y_t, t)\|^2 - \lambda_t \widehat{\mathcal{C}}_{t,n}[f_\theta(\cdot, t)] - f_\theta(Y_t, t) \cdot \left(s_{\pi_n}(X_1) + \frac{J_t^\top Z}{h_t}\right)\Bigg].$$

8: **Update sampler:** $\widetilde{m}_t^{(n+1)}(y) \leftarrow m_{\alpha_n}(y, t) + \eta_t f_{\theta_n}(y, t)$.
9: **Generate fresh endpoints:** simulate the updated sampler to obtain R configurations x and evaluate their gradients $\nabla U(x)$.
10: **Refresh buffer:** replace the oldest R entries of $\mathcal{D}_n$ with the fresh pairs $(x, \nabla U(x))$ to obtain $\mathcal{D}_{n+1}$.
11: **end for**
12: **return** The sampler driven by $\widetilde{m}^{(N)}$.

Here $\widehat{\mathcal{C}}_{t,n}$ denotes the general-coupling covariance estimator from Appendix A.5, evaluated using the current target score s_{π_n} and the frozen denoiser m_{α_n}. The subscript $\mathcal{D}_n$ indicates that endpoints are drawn uniformly from the replay buffer; the expectations also average over prior configurations, bridge times, and noise.

The prior p_0 need only support sampling: prior configurations may be generated on demand or drawn from a stored sample bank, without evaluating the prior density or score. In the ALA2 experiments, we use a bank of configurations generated by a molecular dynamics simulation of the ALA2 Hamiltonian (Appendix B.3) at $1{,}000\,\mathrm{K}$. This prior bank is distinct from the replay buffer, which stores generated endpoints together with their energy gradients and is partially refreshed at each outer iteration. When annealing, the cached gradients are rescaled by the current factor β_n.

Retaining endpoints from earlier samplers makes replay-buffer training an approximation to the population iteration. This reuse follows prior work (Liu et al., 2025; Blessing et al., 2026), and empirically we find that it produces samples that closely match the target distributions defined by the potentials in Section 4.

B EXPERIMENT DETAILS

B.1 DW-4 EVALUATION

We compare ASBS, damped ASBS, BMS, damped BMS, and WASABI using a 16×4 EGNN under matched training settings. Training stops at 1,000 outer rounds for every method. For each seed, we pool 128 fresh endpoints from each of 16 checkpoints, saved every 25 rounds from 625 through 1,000, giving 2,048 endpoints. Each run is evaluated against one random 2,048-frame subsample of the public test split. Configuration W_2 uses the released assignment convention (Blessing et al., 2026). Energy W_2 uses every endpoint, without an energy cutoff.

Table 1 reports means and standard deviations across six seeds, except for damped BMS, where seed 1 is excluded from both metrics after divergence in this window (energy $W_2 = 2{,}449$). The retained five seeds give configuration $W_2 = 1.48 \pm 0.21$ and energy $W_2 = 2.72 \pm 0.58$. Including all six seeds gives 1.46 ± 0.20 and 410 ± 999, respectively.

The pair-distance plot in Figure 2 pools all six pairwise distances from 10,000 WASABI and 10,000 reference configurations. It is a separate illustrative sample set, not the pooled checkpoint samples used for the table. The dashed lines mark the minima of the pairwise potential.

B.2 DW-4 AND LJ-13 POTENTIALS

For both systems, $r_{ij} = \|x_i - x_j\|$ denotes the distance between particles i and j.

Double-well system (DW-4). The four particles lie in two dimensions and interact through

$$U_{\mathrm{DW-4}}(x) = \sum_{i<j} \left[0.9(r_{ij} - 4)^4 - 4(r_{ij} - 4)^2\right]. \tag{67}$$

Lennard–Jones cluster (LJ-13). The 13 particles lie in three dimensions. Their pairwise interactions are supplemented by quadratic confinement around the center of mass:

$$U_{\mathrm{LJ-13}}(x) = 2\sum_{i<j}\left(r_{ij}^{-12} - 2r_{ij}^{-6}\right) + \frac{1}{2}\sum_{i=1}^{13}\|x_i - x_{\mathrm{COM}}\|^2, \qquad x_{\mathrm{COM}} = \frac{1}{13}\sum_{i=1}^{13} x_i. \tag{68}$$

B.3 ALANINE-DIPEPTIDE POTENTIAL AND TRAINING SETUP

Following the alanine-dipeptide benchmark in (Blessing et al., 2026), we use ff96 parameters with OBC1 implicit solvent and an additional chirality restraint. Let $x = (r_1, \ldots, r_n) \in (\mathbb{R}^3)^n$ denote the atomic positions. The configurational potential is

$$\begin{aligned} U(x) &= U_{\mathrm{ff96}}(x) + U_{\mathrm{OBC1}}(x) + U_{\mathrm{chirality}}(x), \\ U_{\mathrm{ff96}}(x) &= U_{\mathrm{bond}}(x) + U_{\mathrm{angle}}(x) + U_{\mathrm{torsion}}(x) + U_{\mathrm{LJ}}(x) + U_{\mathrm{Coulomb}}(x), \\ U_{\mathrm{OBC1}}(x) &= U_{\mathrm{GB}}(x) + U_{\mathrm{SA}}(x). \end{aligned} \tag{69}$$

The nonzero bonded coefficients of this Hamiltonian define the index sets

$$\begin{aligned} E &= \{e = (i,j) : k_e^{\mathrm{b}} \neq 0\}, \\ A &= \{a = (i,j,k) : k_a^{\mathrm{a}} \neq 0\}, \\ T &= \{\tau = (i,j,k,l) : V_{\tau n} \neq 0 \text{ for some } n\}. \end{aligned} \tag{70}$$

Here j is the central atom of an angle triple, and n indexes the Fourier components of a dihedral. For this alanine-dipeptide topology, A coincides with the triples formed by pairs of bonds in E sharing a central atom.

Table 4: Energy terms in the alanine-dipeptide Hamiltonian. The formulas display functional forms; ff96 exclusions and special 1–4 parameters are included in the starred nonbonded sums. Coefficients include any thermal rescaling used to define U.

Term	Energy expression	Physics
U_{bond}	$\frac{1}{2}\sum_{e\in E} k_e^{\mathrm{b}}(\ell_e-\ell_e^0)^2$	Covalent bond stretching.
U_{angle}	$\frac{1}{2}\sum_{a\in A} k_a^{\mathrm{a}}(\alpha_a-\alpha_a^0)^2$	Valence-angle bending.
U_{torsion}	$\sum_{\tau\in T}\sum_n V_{\tau n}[1+\cos(n\phi_\tau-\gamma_{\tau n})]$	Rotation barriers about bonds.
U_{LJ}	$4\sum_{i<j}^{\star}\epsilon_{ij}(\rho_{ij}^{12}-\rho_{ij}^{6})$	Steric repulsion and dispersion.
U_{Coulomb}	$k_e\sum_{i<j}^{\star}\frac{q_iq_j}{r_{ij}}$	Direct charge interactions.
U_{GB}	$-\frac{k_e\Delta_\epsilon}{2}\sum_{i,j}\frac{q_iq_j}{f_{ij}^{\mathrm{GB}}}$	Implicit-solvent electrostatics.
U_{SA}	$4\pi\gamma_{\mathrm{SA}}\sum_i(a_i+a_s)^2(a_i/R_i)^6$	Nonpolar solvation cost.
$U_{\mathrm{chirality}}$	$\frac{k_\chi}{2}\sum_s[\max(d_s-\tau,0)]^2$	Molecular handedness restraint.

Here $r_{ij}=\|r_i-r_j\|$, $\rho_{ij}=\sigma_{ij}/r_{ij}$, and $\Delta_\epsilon=\epsilon_{\mathrm{in}}^{-1}-\epsilon_{\mathrm{out}}^{-1}$. The generalized-Born pair distance is

$$f_{ij}^{\mathrm{GB}}=\left[r_{ij}^2+R_iR_j\exp\left(-\frac{r_{ij}^2}{4R_iR_j}\right)\right]^{1/2}.$$

The R_i are OBC1 effective Born radii, a_i are atomic radii, and a_s is the solvent probe radius. For each restrained dihedral s, $d_s=|\operatorname{wrap}(\chi_s-\chi_s^0)|$ is its wrapped angular distance from the preferred chirality, with tolerance τ.

Bond–angle split readout. The sets E and A defined by the nonzero bond and angle terms index the bond lengths and angle cosines used in our readout. We define the local bonded-coordinate map

$$\begin{aligned} c(x) &= \big(\{\ell_{ij}(x)\}_{(i,j)\in E},\{\cos\alpha_{ijk}(x)\}_{(i,j,k)\in A}\big),\\ \ell_{ij}(x) &= \|r_i-r_j\|,\\ \cos\alpha_{ijk}(x) &= \frac{(r_i-r_j)\cdot(r_k-r_j)}{\|r_i-r_j\|\,\|r_k-r_j\|}. \end{aligned} \tag{71}$$

All Jacobians below are taken in the orthonormal zero-centroid coordinate basis used by the model. Thus $c(x)$ uses the known connectivity and current positions, but neither evaluates the Hamiltonian nor uses its force constants or preferred bond lengths. With $J(x)=Dc(x)$, a small displacement satisfies $c(x+\varepsilon v)=c(x)+\varepsilon J(x)v+O(\varepsilon^2)$. Directions in $\ker J(x)$ leave the selected bond lengths and angles unchanged to first order.

To distinguish these motions in the corrector, we use the regularized map

$$P_\delta(x)=J(x)^\top\big(J(x)J(x)^\top+\delta I\big)^{-1}J(x),\qquad \delta=10^{-4}, \tag{72}$$

and combine the two PaiNN vector heads as

$$f_\theta(x,t)=\sigma_{\mathrm{stiff}}P_\delta(x)H_{\mathrm{stiff}}(x,t)+\sigma_{\mathrm{soft}}\big(I-P_\delta(x)\big)H_{\mathrm{soft}}(x,t). \tag{73}$$

We call this the Bond–Angle Split Heads (BASH) readout. The reported ALA2 runs use a separate PaiNN backbone for each head, trained through their sum. The split is evaluated at the current noisy

configuration. Because P_δ is regularized, this is not an exact orthogonal decomposition and does not fix bond lengths or angles. In particular, $c(x)$ does not encode handedness; the separate chirality restraint acts through the target energy and score.

Training setup and prior preparation. The reported sweeps share a 4,096-endpoint replay buffer and 256 fresh samples per outer iteration (6.25%). We linearly interpolate inverse temperature from 1,000 K to 300 K over the first 1,500 of 4,000 outer iterations:

$$\beta_n = \frac{1-\alpha_n}{k_{\mathrm{B}}(1{,}000\,\mathrm{K})} + \frac{\alpha_n}{k_{\mathrm{B}}(300\,\mathrm{K})}, \qquad \alpha_n = \min\left\{\frac{n}{1{,}500}, 1\right\}, \tag{74}$$

where $n \geq 0$ is the outer-iteration index and $\beta_n = (k_{\mathrm{B}}T_n)^{-1}$. The target remains at 300 K thereafter. Replay samples are reweighted as the temperature changes. The fixed prior bank is separate from this evolving replay buffer. It contains 20,000 configurations from the 1,000 K state of a seven-temperature CUDA REMD run; the full run, including all temperatures and equilibration, took approximately 62.5 minutes. An independent chain supplies the 20,000-frame reference in Figure 4. A separate local benchmark generated 25,000 configurations at 1,000 K in approximately five minutes on an Apple M5 Max using OpenMM's CPU platform with one thread. This timing includes system setup, minimization, 100,000 equilibration steps and 2,500,000 production steps at 1 fs, recording every 100 steps; plotting is excluded. The 20k-frame local-prior budget in Table 3 uses 2,000,000 production steps and the same equilibration length.

We compared this locally generated 20k-frame bank with the 1,000 K states of two independent CUDA REMD chains, using 20,000 frames per bank and the same 50×50 angle grid. Local MD versus the two REMD banks gave $D_{\mathrm{JS}} = 0.0309$ and 0.0307, compared with 0.0296 between the two REMD banks. The discrepancy between the local MD and REMD backbone-angle distributions is comparable to the observed chain-to-chain sampling variability. Based on this agreement, we do not expect replacing the REMD-derived prior with the locally generated high-temperature MD prior to materially affect the reported results. The reported training runs used the REMD-derived bank.

Ramachandran maps. The maps in Figure 4 show relative free energy $\Delta F/(k_{\mathrm{B}}T) = -\log(p/p_{\max})$, where p is the histogram-bin probability and $p_{\max}$ is its maximum within each panel. Free-energy zeros are therefore set separately, and gray bins contain no observed samples. The positive-ϕ region contains 10 of 20,000 WASABI samples (0.05%) and 78 of 20,000 reference frames (0.39%).

Figure 4 compares 20,000 WASABI samples with 20,000 frames from an independent REMD simulation. We show seed 1 for $s_\eta = 0.10$, $\lambda_{\mathrm{cov}} = 0.25$, which had the lowest JS divergence against this reference among the three seeds.

B.4 ALANINE-DIPEPTIDE ABLATIONS AND COMPUTATIONAL COST

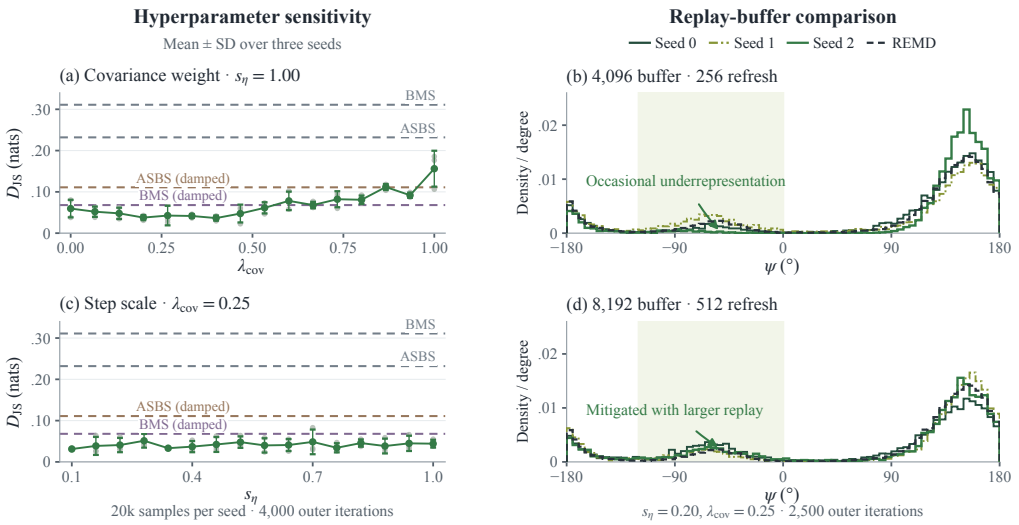


Figure 5: **ALA2 ablations and replay comparison. (a,c)** JS mean ± SD over three seeds; gray dots show individual scores and dashed lines published BMS/ASBS means (Blessing et al., 2026). **(b,d)** Matched seeds with smaller and larger replay settings; dashed curves show REMD. Arrows highlight mitigation of seed 2's central-peak underrepresentation. Both capacity and refresh increase. The sweeps and replay comparison use different checkpoints and sample counts; evaluation protocols are described below.

Influence of hyperparameters. We separately vary the covariance weight and update step scale, holding $s_\eta = 1$ fixed in the covariance sweep and $\lambda_{\text{cov}} = 0.25$ fixed in the step-scale sweep. Each sweep includes 16 settings evaluated over three training seeds, for 96 runs in total (Figure 5(a,c), Table 5). Covariance weights below 0.5 generally give lower JS divergence, with the lowest mean at $\lambda_{\text{cov}} = 0.4$. Performance varies less across the tested step scales, with the lowest mean at $s_\eta = 0.10$.

Replay-buffer size and mode coverage. Low JS divergence can coexist with underrepresentation of a smaller peak in the backbone-angle distribution. Figure 5(b,d) illustrates this behavior in the central peak of the ψ distribution. Increasing the replay capacity from 4,096 to 8,192 and the number of fresh endpoints per iteration from 256 to 512 improves coverage in the highlighted seed, although peak populations remain variable across the five runs evaluated with the larger replay setting. Both replay capacity and refresh count change in this comparison.

Evaluation. Following Blessing et al. (2026), we compute JS divergence in nats using a 50×50 grid of backbone angles. The hyperparameter sweeps use 20,000 generated samples per seed at outer iteration 4,000, evaluated against the one-million-frame validation set released with FAB (reference dataset). Values are reported as mean ± standard deviation over three seeds. Evaluating the same selected checkpoints with 1,024,000 generated samples per seed reduces JS from 0.0312 ± 0.0023 to 0.0254 ± 0.0033 (Table 3), consistent with reduced sampling noise in the empirical histograms. The replay comparison uses matched seeds 0–2 at iteration 2,500, with $s_\eta = 0.20$, $\lambda_{\text{cov}} = 0.25$, 4,096 generated samples per seed, and an independent 20,000-frame REMD reference. The plotted ψ marginals use unsmoothed 5° bins.

Computational cost. We count matrix multiplications in PaiNN's message, update, and readout blocks, including backpropagation, with each multiply-add counted as two FLOPs. Applying a linear map from d_{in} to d_{out} channels to K feature vectors therefore costs approximately

$$2K\, d_{\text{in}} d_{\text{out}} \text{ FLOPs.}$$

We sum these costs using the tensor dimensions in the implementation. Other operations, including activations, geometric calculations, optimizer updates, and molecular-energy evaluations, are excluded.

Table 5: Covariance-weight and step-scale sweeps. Values are mean ± SD over three seeds; bold marks the lowest mean in each sweep.

Covariance sweep $s_\eta = 1.00$ fixed		**Step-scale sweep** $\lambda_{\text{cov}} = 0.25$ fixed	
λ_{cov}	$D_{\text{JS}} \downarrow$	s_η	$D_{\text{JS}} \downarrow$
0.0000	0.0595 ± 0.0212	0.10	$\mathbf{0.0312 \pm 0.0023}$
0.0667	0.0520 ± 0.0125	0.16	0.0387 ± 0.0218
0.1333	0.0479 ± 0.0137	0.22	0.0408 ± 0.0173
0.2000	0.0377 ± 0.0061	0.28	0.0512 ± 0.0167
0.2667	0.0426 ± 0.0237	0.34	0.0332 ± 0.0020
0.3333	0.0416 ± 0.0047	0.40	0.0370 ± 0.0135
0.4000	$\mathbf{0.0365 \pm 0.0061}$	0.46	0.0422 ± 0.0180
0.4667	0.0475 ± 0.0216	0.52	0.0479 ± 0.0139
0.5333	0.0613 ± 0.0135	0.58	0.0400 ± 0.0173
0.6000	0.0782 ± 0.0224	0.64	0.0409 ± 0.0145
0.6667	0.0681 ± 0.0079	0.70	0.0485 ± 0.0299
0.7333	0.0822 ± 0.0194	0.76	0.0342 ± 0.0111
0.8000	0.0808 ± 0.0102	0.82	0.0456 ± 0.0089
0.8667	0.1117 ± 0.0073	0.88	0.0384 ± 0.0179
0.9333	0.0918 ± 0.0062	0.94	0.0449 ± 0.0185
1.0000	0.1562 ± 0.0435	1.00	0.0441 ± 0.0095

For ALA2, one evaluation of our four-layer, 64-channel PaiNN costs 76.6 MFLOPs per molecule. Sampling evaluates two correction networks and one denoiser, totaling 229.7 MFLOPs per integration step. Each outer iteration performs 100 correction updates and 200 denoiser updates with batch size 64, then generates 256 fresh endpoints using 256 integration steps. Including the antithetic and tilted queries used during training gives 31.7 TFLOPs per outer iteration.[4]

For BMS, we assume eight GPUs based on the paper's 16,384-sample replay buffer and the released configuration's 2,048 samples per GPU (Blessing et al., 2026). This implies a global training batch of 512 and 2,048 fresh endpoints per outer iteration. Under these settings, we estimate 112.9 TFLOPs per outer iteration without damping and 190.2 TFLOPs with damping. The paper does not explicitly report the GPU count. We do not estimate ASBS training FLOPs because its corrector and adjoint calculations require separate accounting.

Molecular energy and force evaluations. Table 3 reports estimated molecular-potential calls in millions per run, including force evaluations. One endpoint scored for both energy and force counts once. Neural sampler evaluations and optimizer updates are not molecular-potential calls. Diagnostic and evaluation samples, minimization, and additional energy-only reporting queries are excluded from these nominal force-call budgets.

WASABI scores the 4,096 initial replay endpoints and 256 fresh endpoints at each of 4,000 outer iterations, totaling $4,096 + 4,000 \times 256 = 1,028,096$ calls. Inner updates reuse cached scores; annealing rescales cached physical forces and reweights cached energies without new potential evaluations. Charging each run for a local 20k-frame hot bank adds 2,000,000 production and 100,000 equilibration MD steps, yielding $3,128,096 \approx 3.13 \times 10^6$ calls. This is the budget for the local-prior procedure, rather than a measured end-to-end cost of the reported REMD-prior training runs.

For ASBS and BMS, we use the paper's 5,000 ALA2 outer iterations and the inferred global refresh of 2,048 endpoints, yielding $5,000 \times 2,048 = 10.24 \times 10^6$ calls. The released BMS configuration

[4]Counts retain repeated frozen-denoiser evaluations before compiler optimization. Including the 10,000-update denoiser warmup and the skipped first denoiser refit gives 31.79 TFLOPs per outer iteration averaged over all 4,000 rounds.

initializes replay with the same number of endpoints as a refresh, counted in its first outer iteration. ASBS's released upstream implementation scores fresh adjoint-buffer endpoints and reuses cached terminal costs; corrector fitting uses the analytic reference-diffusion score and does not query the molecular potential. Damping adds neural evaluations, but no additional molecular-potential calls under the same refresh schedule. These counts estimate the published setting, rather than measure its unreleased execution logs; the released BMS default of 7,000 iterations would instead imply 14.336×10^6 calls.

The REMD row uses seven temperatures, 20,000 production cycles and 1,000 equilibration cycles, with 500 MD steps per cycle at 1 fs. Across replicas, this gives $7 \times (20,000 + 1,000) \times 500 = 73.5 \times 10^6$ force evaluations. There are additionally 147,000 energy queries for exchange decisions and initial minimization work. This row records the cost of the reference protocol; it is not an accuracy-matched comparison with trained samplers, and no self-comparison JS score is reported.

C Related work

Normalizing flow-based samplers Normalizing flows (Rezende & Mohamed, 2015; Papamakarios et al., 2017; Dinh et al., 2017) underpin Boltzmann Generators (BG, Noé et al., 2019), which learn an invertible map from a tractable base distribution to target Boltzmann distributions. Early BGs combine energy-based objectives with training on simulation data. Sequential Boltzmann Generators (SBG) (Tan et al., 2025) train normalizing flows on biased MD datasets and correct their samples using Sequential Monte Carlo along a temperature-annealing path. In comparison, WASABI learns from target energy gradients and self-generated samples. Energy-based training is also possible with normalizing flows: FAB (Midgley et al., 2023) combines flows with annealed importance sampling to improve mode coverage without pre-generated training samples, while its implementation uses internal molecular coordinates. In comparison, WASABI operates in Cartesian coordinates. Recent BG architectures exploit locality and size-transferable flow models to enable energy-based training and equilibrium sampling of larger many-particle systems (Schebek et al., 2026). However, the invertibility requirement can make normalizing flows less expressive than diffusion models.

Diffusion-based samplers Diffusion based sampling methods learn amortized samplers for unnormalized targets using access to the target energy and its derivatives, without requiring a pre-generated target dataset. PIS (Zhang & Chen, 2022) formulates sampling as stochastic optimal control and trains a controlled diffusion with a path-space objective. DDS (Vargas et al., 2023) constructs a noising process from the target to a tractable reference and learns the corresponding time reversal. iDEM (Akhound-Sadegh et al., 2024) estimates scores of noised target distributions from energy evaluations and iteratively refines a diffusion model using its own samples. PITA (Akhound-Sadegh et al., 2025) progressively trains lower temperature diffusion models using Feynman–Kac weighting and Sequential Monte Carlo along an annealing path.

More recent methods use bridge-based regression to iteratively refine the sampler. Adjoint Sampling (AS) (Havens et al., 2025) uses reciprocal adjoint matching to regress diffusion drifts while reusing endpoint energy gradients across reference bridges. Adjoint Schrödinger Bridge Sampler (ASBS) (Liu et al., 2025) extends this construction to general source distributions through a Schrodinger bridge formulation with a learned terminal corrector. BMS (Blessing et al., 2026) introduces a generalized fixed-point matching framework that recovers AS and ASBS, with an independent-coupling construction yielding explicit regression targets without a separate corrector. Flow Sampling (Havens et al., 2026) similarly learns denoising drifts from source-conditioned probability paths using regression targets constructed from generated endpoints and endpoint energy gradients. WASABI likewise learns through regression using generated samples and target energy gradients. Its update is derived from Wasserstein descent of the sampling distribution's KL divergence to the target, accounting for both sample displacement and the resulting changes in the conditional weights defining the denoiser.