BRIDGING STOCHASTIC FLOW MAPS AND BOLTZMANN GENERATORS WITH NORMALIZING FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating independent, equilibrium samples of molecular systems at scale remains a central obstacle in computational statistical mechanics. Boltzmann Generators address this by pairing a generative model with importance sampling to obtain consistent samples from the target distribution. We introduce Normalizing Flow Flow Maps (NF^2M), which combines the strengths of recent stochastic flow maps with the tractability of classic normalizing flows to build a Boltzmann Generator. Unlike most methods, which correct the generative model only at the end, NF^2M reweighs each denoising transition as generation proceeds, avoiding wasted compute on trajectories that are ultimately discarded. At each denoising step, a conditional normalizing flow proposes clean configurations given the current noisy state (a simpler task than sampling directly from the target) and its exact likelihood enables correcting each proposal toward the true denoising transition of the target Boltzmann distribution. This is in contrast to most existing methods, whose likelihoods are approximate or expensive to evaluate, undermining the statistical reliability of the correction. We establish consistency of the corrected transitions and bound how approximation errors propagate through the sampling chain. We evaluate NF^2M on peptide systems, demonstrating improved sampling efficiency and sample quality.

1 INTRODUCTION

Sampling the Boltzmann distribution of molecular systems is central to statistical physics, underlying protein folding, ligand binding, conformational change, and molecular design (Liu, 2001; Frenkel & Smit, 2001; Noé et al., 2009; Lindorff-Larsen et al., 2011; Buch et al., 2011). This remains hard because molecular energy landscapes are high-dimensional and rugged, with metastable states separated by large barriers. Classical approaches like MD and MCMC are asymptotically exact (Leimkuhler & Matthews, 2015; Frenkel & Smit, 2001; Wirnsberger et al., 2020), but their local-update nature makes them mix slowly on such landscapes, producing correlated samples and requiring prohibitively long runs to capture transitions between metastable states.

Boltzmann Generators (BGs) take a different strategy (Noé et al., 2019; Müller et al., 2019): a learned generative model produces plausible configurations independently and in parallel, which are then statistically corrected toward the true equilibrium distribution, e.g. via importance sampling (IS). This hinges on two requirements: proposal-target overlap and a tractable correction procedure—that existing generative backbones trade off differently. Normalizing flows (NFs) (Rezende & Mohamed, 2015; Dinh et al., 2017; Noé et al., 2019; Papamakarios et al., 2021) dominate this role because their tractable likelihoods enable IS correction, but their invertibility constraints limit expressiveness and overlap. Bridging the NF to target gap via temperature-annealed interpolation (Tan et al., 2025) instead suffers from mass teleportation on multimodal targets (Woodard et al., 2009; Máté & Fleuret, 2023) and does not reliably outperform direct importance sampling (Tan et al., 2025). Continuous normalizing flows built from ODE trajectories (Chen et al., 2018; Lipman et al., 2023) offer more expressive architectures, but their likelihoods require integrating the vector-field divergence, making correction costly and discretization-sensitive. Distilling these into few-step flow maps (Boffi et al., 2025) sacrifices tractable likelihoods altogether. Subsequent work restores likelihoods for such few-step maps in two ways : FALCON (Rehman et al., 2026a) computes them from the full Jacobian of the map, at the cost of memory overhead and sensitivity to the map's numerical invertibility, while F2D2 (Ai et al., 2026) and SCALLOP (OuYang et al., 2026) instead distill a likelihood-prediction head, gaining speed but risking error propagation from the learned likelihood into the correction step.

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.

Diffusion models offer a different stochastic transport path, gradually perturbing clean samples via a tractable noising process and generating through learned, typically Gaussian, denoising transitions (Ho et al., 2020; Song et al., 2021). These paths handle multimodal targets better than temperature annealing and have shown promise for BGs (Grenioux & Noble, 2026): although the endpoint density is intractable, the path density can be evaluated, enabling correction in an extended space (Zhang et al., 2024; 2025). The Gaussian approximation is accurate only for fine discretizations, however, creating a tradeoff between statistical performance and inference cost (Grenioux & Noble, 2026). Distributional diffusion and stochastic flow maps address this by modeling the full conditional distribution of clean samples given a noisy state, enabling expressive transitions over larger stepsizes (De Bortoli et al., 2025; Potaptchik et al., 2026; Holderrieth et al., 2026) but the resulting models are typically implicit, lacking the likelihoods BG correction needs.

We close this gap with *Normalizing Flow Flow Maps* (NF^{2}M), a BG that generates samples along a diffusion transport path in a few steps using denoising kernels estimated via importance sampling. Like stochastic flow maps, NF^{2}M models the conditional distribution $q^\theta(x_0|x_t)$ of clean data given a noisy state, but it uses a conditional NF rather than an implicit model, yielding a tractable density.

This tractability lets us build an importance-sampling estimator of the true denoising kernel, exact as the number of particles grows: at each step, we propose candidates from $q^\theta(x_0|x_t)$ and reweight them against the tractable conditional target $p(x_0|x_t)$ (which is the product of the Boltzmann density and the diffusion noising kernel) yielding a Rao-Blackwellized mixture estimator that we sample to denoise the current state. Iterating this gradually transports noise toward the Boltzmann target, unlike NF- and CNF-based BGs that bridge the gap in one step. By stochastic localization (Grenioux et al., 2024), $p(x_0|x_t)$ loses its multimodality and concentrates as $t \to 0$ which makes it easier to sample from and simplifies the conditional NF's approximation target.

Our main contributions are:

- We propose *Normalizing Flow Flow Maps* (NF^{2}M), a Boltzmann Generator that parameterizes a stochastic flow map with a normalizing flow, combining tractable densities with few-step generation.
- We design novel conditional normalizing flow architectures for molecule generation, adapting state-of-the-art image and video normalizing flows to this setting (Zhai et al., 2025; Gu et al., 2026).
- We propose two inference strategies, NF^{2}M-IS and NF^{2}M-MIS, that yield asymptotically exact samples from the Boltzmann distribution, and support them with theoretical guarantees.
- We evaluate NF^{2}M against NF and flow-map baselines across molecular systems of varying sizes, demonstrating competitive or superior performance.

2 Background

Problem setup. We represent an N-atom molecule configuration with 3D coordinate $x \in \mathbb{R}^{N\times 3}$ and we seek equilibrium samples from a Boltzmann distribution

$$p(x) = \frac{1}{\mathcal{Z}} \exp\{-\mathcal{E}(x)\}, \qquad \mathcal{Z} = \int_{\mathbb{R}^d} \exp\{-\mathcal{E}(x)\}\, \mathrm{d}x, \tag{1}$$

where $\mathcal{E} : \mathbb{R}^{N\times 3} \to \mathbb{R}$ is an energy function and $\mathcal{Z}$ is generally intractable. We assume that $\mathcal{E}(x)$ can be evaluated for any x. Following prior work (Noé et al., 2019), we also assume access to a small, biased dataset $\mathcal{D} = \{x^i\}_{i=1}^D$, such as configurations obtained from a short, unmixed MD trajectory. We denote by q the biased data distribution underlying $\mathcal{D}$.

Boltzmann Generators. Boltzmann Generators (BGs) (Noé et al., 2019; Müller et al., 2019) use a learned model q^θ to propose configurations and then correct its mismatch with the Boltzmann target by *self-normalized importance sampling (SNIS)*. Given M independent proposals $x^m \sim q^\theta$, the unnormalized weights and estimator of an observable $\mathcal{O}$ are

$$\widetilde{w}^m = \frac{\exp\{-\mathcal{E}(x^m)\}}{q^\theta(x^m)}, \qquad \widehat{\mathbb{E}}_p[\mathcal{O}] = \frac{\sum_{m=1}^M \widetilde{w}^m \mathcal{O}(x^m)}{\sum_{m=1}^M \widetilde{w}^m}. \tag{2}$$

The estimator is generally biased at finite M but is consistent as $M \to \infty$, provided the proposal's support covers the target's. Its variance depends on the overlap between q^θ and p, and computing the importance weights requires evaluating the proposal density $q^\theta(x)$.

Normalizing flows. Normalizing flows (NFs) (Rezende & Mohamed, 2015; Dinh et al., 2017) are a natural choice for parameterizing q^θ because they support both sampling and exact density evaluation. A NF learns a diffeomorphism $\mathrm{T}_\theta : \mathbb{R}^d \to \mathbb{R}^d$ from data to a base density ρ, typically $\mathcal{N}(0, I)$, so that T_θ^{-1} maps base samples to data space. Its density follows from the change-of-variables formula,

$$\log q^\theta(x) = \log \rho(\mathrm{T}_\theta(x)) + \log \left| \det \frac{\partial \mathrm{T}_\theta(x)}{\partial x} \right|. \tag{3}$$

In practice, T_θ is a composition of invertible layers whose inverses and Jacobian determinants are tractable. These structural requirements enable exact likelihoods but constrain the neural-network architectures that can be used, thereby limiting the expressivity of the learned distribution.

Diffusion paths and denoising posteriors. Diffusion models (Ho et al., 2020; Song et al., 2021) transform a tractable reference p_T into a target p_0 through learned denoising transitions. To train them, consider a prescribed continuous-time forward process that progressively noises $X_0 \sim p_0$, yielding states $X_t \sim p_t$ for $t \in [0, T]$. We use a linear Gaussian noising kernel, yielding the marginal

$$p_{t|0}(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I), \tag{4}$$

$$p_t(x_t) = \int p_0(x_0) p_{t|0}(x_t \mid x_0)\, \mathrm{d}x_0. \tag{5}$$

We use the variance-exploding (VE) schedule of Karras et al. (2022), for which $\alpha_t = 1$ and $\sigma_t = t$. At sufficiently large T, the signal of p_0 in p_T is small relative to the added noise, so the terminal marginal is approximated by the tractable reference distribution $p_T \approx \mathcal{N}(0, \sigma_T^2 I)$.

Generation starts from $X_T \sim p_T$ and successively samples less noisy states. For $0 \le s < t \le T$, this requires the denoising transition $p_{s|t}(x_s|x_t)$. Given x_t, one first infers from the *denoising posterior*

$$p_{0|t}(x_0 \mid x_t) = p_0(x_0) p_{t|0}(x_t \mid x_0)/p_t(x_t). \tag{6}$$

Marginalizing over these possible clean states gives the exact decomposition

$$p_{s|t}(x_s \mid x_t) = \int p_{s|t,0}(x_s \mid x_t, x_0)\, p_{0|t}(x_0 \mid x_t)\, \mathrm{d}x_0, \tag{7}$$

where $p_{s|t,0}$ is an endpoint conditioned Gaussian given the clean state x_0 and the noisy state x_t,

$$p_{s|t,0}(x_s \mid x_t, x_0) = \mathcal{N}(x_s; a_{s,t} x_0 + b_{s,t} x_t, \Sigma_{s,t}), \tag{8}$$

with the parameters determined by the forward process:

$$(\alpha_{t|s}, \sigma_{t|s}^2) = \left(\alpha_t/\alpha_s,\ \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2\right),\ (a_{s,t}, b_{s,t}, \Sigma_{s,t}) = \left(\alpha_s \sigma_{t|s}^2/\sigma_t^2,\ \alpha_{t|s}\sigma_s^2/\sigma_t^2,\ \sigma_{t|s}^2 \sigma_s^2/\sigma_t^2 I\right).$$

Diffusion models approximate it by a point mass at a learned conditional mean $\mu_{0|t}^\theta(x_t) \approx \mathbb{E}[X_0|X_t = x_t]$, trained on paired data (x_0, x_t). The reverse transition is approximated as Gaussian

$$p_{s|t}(x_s \mid x_t) \approx \mathcal{N}\left(x_s; a_{s,t} \mu_{0|t}^\theta(x_t) + b_{s,t} x_t, \Sigma_{s,t}\right). \tag{9}$$

With an accurate conditional mean prediction, this approximation recovers the reverse diffusion dynamics in the continuous-time limit as the step size tends to zero (Song et al., 2021). With finite steps, however, this approximation can be poor for coarse step sizes.

3 Normalizing Flow Flow Maps (NF^{2}M)

While existing BG designs compose approximate transitions and correct accumulated error only at the end by reweighting the full composition, NF^{2}M instead reweighs each denoising transition directly along the diffusion path of Sec. 2, with target $p_0 = p$. Since each transition is individually asymptotically unbiased, so is their composition, with no need for a final correction. Concretely, NF^{2}M learns a conditional normalizing flow $q_{0|t}^\theta(x_0|x_t)$ approximating the denoising posterior at every noise level: the flow proposes clean configurations from the current noisy state, importance sampling corrects them toward the true posterior, and the result determines the transition via Eq. (7).

Sec. 3.1 describes the basic sampler, NF^{2}M-IS; Sec. 3.2 covers its training objective and flow architecture; Sec. 3.3 improves inference through a trajectory-sharing mechanism and a custom MCMC scheme; and Sec. 3.4 provides theoretical intuition and guarantees.

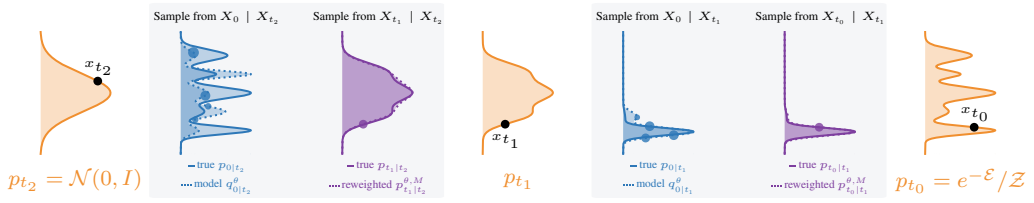


Figure 1: Conceptual illustration of the NF^{2}M mechanism. Given a sample x_t drawn from the marginal distribution at noise level t, we first draw M samples $\{x^m_{0|t}\}^M_{m=1}$ from the learned posterior $q^{\theta}_{0|t}(\cdot|x_t)$. We then reweight them via Eq. (12) to construct the posterior $p^{\theta,M}_{s|t}$, which proposes a sample x_s at the next noise level s. We repeat this procedure until reaching the target distribution.

3.1 Boltzmann sampling along the diffusion path

An ideal sampler along the diffusion path would follow the exact denoising transitions $p_{s|t}$ in Eq. (7), starting from p_T and ending at the Boltzmann distribution $p_0 = p$. The intractable component in Eq. (7) is the Boltzmann denoising posterior $p_{0|t}$, which satisfies by Eq. (6)

$$p_{0|t}(x_0 \mid x_t) \propto \exp\{-\mathcal{E}(x_0)\}\, p_{t|0}(x_t \mid x_0). \tag{10}$$

While we cannot sample from it, the right hand size of Eq. (10) can be evaluated pointwise. For now, suppose we have a posterior sampler $q^{\theta}_{0|t}$ trained on the biased dataset q, we would like to treat it as a proposal and employ SNIS to sample from the target posterior distribution $p_{0|t}$. Although $q^{\theta}_{0|t}$ generally deviates from the target posterior $p_{0|t}$, we instead use it as a proposal in a SNIS scheme, which corrects for this mismatch before constructing the next denoising transition.

SNIS-corrected transition kernel. Fix $0 \le s < t \le T$ and a noisy state x_t. We draw M independent clean states $x^m_{0|t} \sim q^{\theta}_{0|t}(\cdot|x_t)$ and assign them self-normalized importance weights

$$w^m_t = \frac{\widetilde{w}^m_t}{\sum_{j=1}^M \widetilde{w}^j_t}, \qquad \widetilde{w}^m_t = \frac{\exp\{-\mathcal{E}(x^m_{0|t})\}\, p_{t|0}(x_t \mid x^m_{0|t})}{q^{\theta}_{0|t}(x^m_{0|t} \mid x_t)}. \tag{11}$$

As $M \to \infty$, the weighted empirical measure $\sum_{m=1}^M w^m_t \delta_{x^m_{0|t}}$ approximates the denoising posterior $p_{0|t}(\cdot|x_t)$. Substituting it for $p_{0|t}$ inside the integral of Eq. (7) and marginalizing analytically over the Gaussian bridge $p_{s|t,0}$ turns the true transition into a finite mixture, the SNIS-corrected transition

$$p_{s|t}(x_s \mid x_t) \approx \widehat{p}^{M,\theta}_{s|t}(x_s \mid x_t) := \sum_{m=1}^M w^m_t\, p_{s|t,0}(x_s \mid x_t, x^m_{0|t}), \tag{12}$$

in place of the true kernel's infinite mixture over $p_{0|t}$. This is a Rao-Blackwellized estimator: we retain the full conditional law $p_{s|t,0}$ and mix over it analytically, rather than drawing and reweighting a single noisy sample per proposal. The exact and approximate transitions share the same endpoint-conditioned Gaussian $p_{s|t,0}$ and differ only through $p_{0|t}$ versus its finite-mixture approximation. Since x_0 enters the Gaussian mean with coefficient $a_{s,t}$, errors in the denoising posterior are attenuated by $|a_{s,t}|$ when propagated to the next state, as formalized in Sec. 3.4.

Algorithm 1 NF^{2}M-IS for a single trajectory

Input: Grid $\{t_k\}^K_{k=0}$, particle count M, flow $q^{\theta}_{0|t}$, energy $\mathcal{E}$.
Output: x_0 approximately distributed as $p(\cdot) \propto \exp\{-\mathcal{E}(\cdot)\}$.
Sample $x_K \sim \mathcal{N}(0, \sigma_T^2 I)$
for $k = K, \dots, 1$ **do**
  Draw $x^m_{0|k} \overset{\text{iid}}{\sim} q^{\theta}_{0|t_k}(\cdot|x_k)$, $m = 1, \dots, M$
  Compute weights $\widetilde{w}^m_k = \dfrac{\exp\{-\mathcal{E}(x^m_{0|k})\}\, p_{t_k|0}(x_k|x^m_{0|k})}{q^{\theta}_{0|t_k}(x^m_{0|k}|x_k)}$
  Sample $x_{k-1} \sim \sum_{m=1}^M \dfrac{\widetilde{w}^m_k}{\sum_{j=1}^M \widetilde{w}^j_k}\, p_{t_{k-1}|t_k,0}(\cdot|x_k, x^m_{0|k})$
end
return x_0

Chaining the SNIS-corrected transition in Eq. (12) along a discretized diffusion path yields the full sampler, NF^{2}M-IS: given a time grid $0 = t_0 < t_1 < \cdots < t_K = T$, we initialize $x_K \sim \mathcal{N}(0, \sigma_T^2 I)$

and apply $\widehat{p}^{M,\theta}_{s|t}$ successively down to x_0, as summarized in Alg. 1. The approximation Eq. (12) converges to the corresponding true denoising kernel as $M \to \infty$. The practical error also includes the terminal approximation $p_T \approx \mathcal{N}(0, \sigma_T^2 I)$ and the finite-M errors accumulated along the chain. We study their propagation in Sec. 3.4 and App. A.1.

3.2 TRAINING

We now discuss how to train the conditional flow $q^\theta_{0|t}$. Although equilibrium samples from the Boltzmann target p are unavailable, we have access to a biased dataset $\mathcal{D}$ drawn from some distribution $q := q_0$. Applying the same forward noising process to q_0 induces a noisy marginal q_t and denoising posterior $q_{0|t}$, and we train $q^\theta_{0|t}$ to approximate $q_{0|t}$ from noisy–clean pairs generated from $\mathcal{D}$. The bias inherited from $\mathcal{D}$ is corrected only later, at inference time.

Training objective. For a positive time-weighting function $\lambda(t)$, we minimize the forward KL from $q_{0|t}$ to $q^\theta_{0|t}$, averaged over the noisy marginal q_t,

$$\mathcal{L}_{\mathrm{KL}}(\theta) = \int_0^T \lambda(t)\, \mathbb{E}_{X_t \sim q_t} \left[\mathbb{D}_{\mathrm{KL}} \left(q_{0|t}(\cdot \mid X_t) \,\|\, q^\theta_{0|t}(\cdot \mid X_t) \right) \right] \mathrm{d}t. \tag{13}$$

We note that the reverse KL $\mathbb{D}_{\mathrm{KL}}(q^\theta_{0|t} \| p_{0|t})$ would also be tractable up to a constant via Eq. (10). We leave this alternative to future work and focus here on the forward-KL formulation.

Since $q_t(x_t) q_{0|t}(x_0|x_t) = q_0(x_0) p_{t|0}(x_t|x_0)$ by Bayes rule, expanding the KL and dropping terms independent of θ reduces Eq. (13) to the tractable conditional maximum-likelihood objective

$$\mathcal{L}_{\mathrm{NF}^2\mathrm{M}}(\theta) = -\int_0^T \lambda(t)\, \mathbb{E}_{X_0 \sim q_0} \mathbb{E}_{X_t \sim p_{t|0}(\cdot|X_0)} \left[\log q^\theta_{0|t}(X_0 \mid X_t) \right] \mathrm{d}t. \tag{14}$$

Training thus requires only samples from biased data $\mathcal{D}$, noise levels t, and the forward noising kernel. In principle, NF^{2}M applies to any conditional family $q^\theta_{0|t}$ that supports sampling and exact likelihood evaluation. For example, an isotropic Gaussian with learned mean and fixed variance $v_t I$ gives

$$q^\theta_{0|t}(x_0 \mid x_t) = \mathcal{N}(x_0; \mu^\theta_{0|t}(x_t), v_t I), \tag{15}$$

then Eq. (14) becomes, up to terms independent of θ,

$$\int_0^T \frac{\lambda(t)}{2 v_t}\, \mathbb{E}_{X_0 \sim q_0} \mathbb{E}_{X_t \sim p_{t|0}(\cdot|X_0)} \left[\left\| X_0 - \mu^\theta_{0|t}(X_t) \right\|^2 \right] \mathrm{d}t. \tag{16}$$

This is the usual x_0-prediction denoising objective in diffusion models (Ho et al., 2020; Karras et al., 2022), where $\mu^\theta_{0|t}$ is optimized to predict the conditional mean $\mathbb{E}[X_0|X_t = x_t]$ induced by q_0.

In this work, we replace the Gaussian conditional with a flexible conditional NF architecture. This choice enables representing multimodal denoising posteriors while retaining the likelihood required for training in Eq. (14) and for inference in Eq. (11).

Conditional Normalizing Flow architecture. We develop conditional molecular variants of TarFlow (Zhai et al., 2025) and StarFlow (Gu et al., 2026). Both are autoregressive flows (Huang et al., 2018) that use causal Transformers for autoregressive affine coupling, giving a triangular Jacobian that enables parallel density evaluation with sequential sampling. TarFlow stacks several medium conditioned blocks while StarFlow uses one large conditional block followed by small unconditional ones. Following Tan et al. (2025), we represent an N-atom configuration as N tokens in $\mathbb{R}^3$, conditioning both backbones on noise level t (via DiT-style adaptive layer norm (Peebles & Xie, 2023)) and on x_t itself, either through cross-attention (features of x_0 as queries, x_t as keys/values) or a prefix mechanism (prepending x_t features to the autoregressive sequence). Details in App. A.5.

Molecular symmetries. Following Tan et al. (2025), we enforce translation and rotation invariance by centering and randomly rotating configurations rather than using equivariance. We lift them to $\mathbb{R}^{N\times 3}$ with a Gaussian centroid $c_0 \sim \mathcal{N}(0, N^{-1} I_3)$, tractable under a standard Gaussian base. To reduce centroid-induced importance-weight variance, a closed-form adjustment to $\log q^\theta_{0|t}(x_0|x_t)$ removes the rotationally irrelevant angular component of the centroid's offset from its conditional mean given x_t. This recovers Tan et al. (2025)'s unconditional correction at high noise (App. A.4).

3.3 IMPROVING THE INFERENCE

Multiple Importance Sampling. NF^{2}M-IS shares candidates only at initialization. Can this sharing be extended further along the chain? When p is a Gaussian $\mathcal{N}(\mu, \gamma^2 \mathrm{I}_d)$, one can show that

$$\mathbb{D}_{\mathrm{KL}}(p_{0|t}(\cdot \mid x) \parallel p_{0|t}(\cdot \mid y)) = \frac{\alpha_t^2 \gamma^2 \|x-y\|^2}{2\sigma_t^2(\alpha_t^2\gamma^2 + \sigma_t^2)}, \forall x, y \in \mathbb{R}^d, t \in [0,T]. \tag{17}$$

Posteriors conditioned on nearby states are themselves close, which suggests a simple idea: when x and y are close, candidates drawn from $q^{\theta}_{0|t}(\cdot|x)$ should also be informative proposals for $p_{0|t}(\cdot|y)$.

This is exactly the idea behind ***NF^{2}M-MIS***: each trajectory reuses candidates generated for *other* trajectories, rather than drawing its own (Fig. 3). Concretely, instead of keeping the M candidates from each of N proposals separate, we pool them into N groups of C shared proposal components, so that each trajectory draws on CM candidates rather than M, for the same total of NM flow samples as NF^{2}M-IS. This comes at the cost of evaluating additional cross-likelihoods for the importance weights, an overhead that remains comparatively inexpensive for our autoregressive flow architectures.

Such sharing is especially valuable at high noise levels, where noisy states are less informative about x_0 and the posterior across trajectories approach the common Boltzmann target. follows literature on multiple importance sampling (Veach & Guibas, 1995; Owen & Zhou, 2000; He & Owen, 2014; Bugallo et al., 2017; Elvira et al., 2019). See App. A.2 for algorithmic details.

Posterior-adapted MCMC rejuvenation. We can refine the IS or MIS sample with a few steps of MCMC targeting the posterior $p_{0|t}$ directly. Its score decomposes as $\nabla_{x_0} \log p_{0|t}(x_0|x_t) = \nabla \log p(x_0) - Ax_0 + b$, with $A = \alpha_t^2/\sigma_t^2$ and $b = (\alpha_t/\sigma_t^2)x_t$, giving the Langevin diffusion

$$\mathrm{d}Y_s = (-AY_s + b + \nabla \log p(Y_s))\,\mathrm{d}s + \sqrt{2}\,\mathrm{d}W_s. \tag{18}$$

Holding $\nabla \log p$ fixed and integrating the affine drift exactly yields the Gaussian kernel

$$Y_{s+h} \mid Y_s \;\sim\; \mathcal{N}\left(e^{-Ah}Y_s + \tfrac{1-e^{-Ah}}{A}\,(b + \nabla \log p(Y_s))\,,\; \tfrac{1-e^{-2Ah}}{A} I\right), \tag{19}$$

or, using the noise-to-signal ratio $\tau = (1-e^{-2Ah})/(2A) \in (0, \tau_{\max}]$, with $\tau_{\max} = 1/(2A)$,

$$Y_{s+h} \mid Y_s \;\sim\; \mathcal{N}\left(c(\tau)Y_s + \varphi(\tau)\,(b + \nabla \log p(X_s))\,,\; 2\tau I\right). \tag{20}$$

A Metropolis–Hastings accept/reject step makes the proposal kernel invariant under $p_{0|t}$, so rejuvenated particles retain their original importance weights. See App. A.3 for the full construction.

3.4 THEORETICAL GUARANTEES

Why approximate the denoising posterior at intermediate noise levels instead of applying SNIS directly to the final Boltzmann target? An intermediate denoising step need not inherit the full error of the clean-state approximation. The following proposition quantifies how the endpoint-conditioned Gaussian transition controls this error; its proof is given in App. A.1.

Proposition 1 *Let $\hat{p}_{0|t}$ be an approximation of $p_{0|t}$ and define the corresponding denoising kernel*

$$\hat{p}_{s|t}(x_s \mid x_t) = \int p_{s|t,0}(x_s \mid x_t, x_0)\, \hat{p}_{0|t}(x_0 \mid x_t)\, \mathrm{d}x_0.$$

Let $\ell \geq 1$, $x_t \in \mathbb{R}^d$, and assume that both $p_{0|t}(\cdot|x_t)$ and $\hat{p}_{0|t}(\cdot|x_t)$ have finite ℓ-th moment. Then

$$W_\ell\left(p_{s|t}(\cdot \mid x_t),\, \hat{p}_{s|t}(\cdot \mid x_t)\right) \leq a_{s,t}\, W_\ell\left(p_{0|t}(\cdot \mid x_t),\, \hat{p}_{0|t}(\cdot \mid x_t)\right).$$

Error attenuation and posterior concentration. Prop. 1 shows that errors in approximating $p_{0|t}$ are damped by $a_{s,t}$ when propagated to $p_{s|t}$. Under the VE schedule, $a_{s,t} = 1 - (s/t)^2$ grows with the relative step size $(t-s)/t$, so damping is strongest where the schedule takes relatively small steps (Fig. 2), which is why a poorly approximated posterior in Fig. 1 can still yield a good SNIS approximation $q^{\theta,M}_{s|t}$. Near $t \approx T$, where $p_{0|t}(\cdot|x_t) \approx p_0$ is broad and hard to learn, the Karras et al. (2022) discretization takes the smallest relative steps and so makes $a_{s,t}$ smallest, attenuating their propagation. Conversely, at low noise $p_{0|t}(\cdot|x_t)$ concentrates around x_t and loses its multimodality, which should make the flow easier to approximate but a sharply peaked target remains hard to represent with affine couplings. MAExL rejuvenation covers this regime, being most accurate at low noise (App. A.3), so the larger $a_{s,t}$ acts on an error that is already small.

Per-step error. Prop. 1 bounds a single step for an abstract $\hat{p}_{0|t}$. To obtain a guarantee for the full sampler, we substitute the SNIS estimator for $\hat{p}_{0|t}$ and track how per-step errors accumulate along the chain (proofs in App. A.1).

Proposition 2 (Per-step SNIS error, informal version of Thm. 7) *Let $0 < s < t$ and let $q_{s|t}^{\theta,M}$ use proposal $q_{0|t}$. Assume $q_{0|t}$ and its square-weighted tilt have sub-Gaussian tails of rate $\kappa_t(x_t)$. For any bounded ψ, the bias and MSE of $q_{s|t}^{\theta,M}(\psi|x_t)$ relative to $p_{s|t}(\psi \mid x_t)$ satisfy,*

$$\text{bias} = \mathcal{O}\left(\frac{\rho_t(x_t)\gamma_{t,s}\kappa_t(x_t)\sqrt{d}}{M}\right), \qquad \text{MSE} = \mathcal{O}\left(\frac{\rho_t(x_t)\gamma_{t,s}^2\kappa_t(x_t)^2 d}{M}\right),$$

up to $\mathcal{O}(M^{-2})$ terms and logarithmic factors in M, where $\rho_t(x_t) = 1 + \chi^2(p_{0|t}(\cdot|x_t)\|q_{0|t}(\cdot|x_t))$ measures proposal quality and $\gamma_{t,s}$ is the schedule-dependent bridge sensitivity Eq. (22).

As in Prop. 1, the bounds reflect proposal fit (ρ_t) and sensitivity to residual x_0-error ($\gamma_{t,s}\kappa_t$). SNIS adds the number of importance-sampling particles M, which controls the decay of bias and MSE.

Error propagation. Prop. 2 gives a per-step control statement about a single transition. The sampler, however, chains K such transitions to go from t_K down to τ, so what ultimately matters is how these per-step errors combine along the full path. We answer this next.

Proposition 3 (Path error, informal version of Cor. 16) *Let $T = t_K > \cdots > t_0 = \tau > 0$ and let $q_{t_0}^M$ denote the estimator obtained by chaining the M-particle SNIS kernels of Prop. 2 along the schedule, started from the exact law at t_K. For any bounded test function ψ, let $\Delta_M := q_{t_0}^M(\psi) - p_{t_0}(\psi)$. Then*

$$|\mathbb{E}[\Delta_M]| \le \sum_{k=1}^{K} \mathbb{E}_{p_{t_k}}[B_k(X_{t_k})], \qquad \sqrt{\mathbb{E}[\Delta_M^2]} \le \sum_{k=1}^{K} \sqrt{\mathbb{E}_{p_{t_k}}[V_k(X_{t_k})]},$$

where $B_k, V_k = O(1/M)$ are exactly the per-step bias and MSE rates of Prop. 2 at step k.

The bias thus accumulates only *additively*, and the root-MSE only through a *triangle inequality*, across the K denoising steps.

4 Related Works

BGs with tractable correction weights. Sequential Boltzmann Generators (SBG) use sequential Monte Carlo (SMC) to sample along a geometric annealing path between a trained TarFlow proposal and the Boltzmann target. However, their ablation results show that SMC does not consistently improve over SNIS on the same trained proposal across systems and metrics (Tan et al., 2025). NF^{2}M also deploys a TarFlow sequentially, but both inference and learning differ fundamentally: rather than annealing a fixed proposal toward the target, NF^{2}M performs progressive denoising transitions, and the flow itself is trained as a conditional model rather than an unconditional one.

The Autoregressive Boltzmann Generator (ArBG) models coordinate-bin probabilities with uniform dequantization (Rehman et al., 2026b), yielding an exactly evaluable but bin-wise coarse density. Its coordinate-wise factorization enables twisted SMC using temporarily capped peptide energies to resample partial structures. Stochastic normalizing flows interleave invertible maps with MCMC or Langevin transitions, weighting entire sampling paths (Wu et al., 2020). NF^{2}M instead corrects denoising proposals over complete molecular configurations at each noise level, avoiding both the binning artifacts of ArBG and the need to evaluate energies of artificially capped partial structures.

BGs with approximate correction. Free-form Flows jointly train a network and its pseudo-inverse by maximum likelihood (Draxler et al., 2024). Continuous normalizing flows keep maximum-likelihood training but replace the invertible network with a neural ODE (Köhler et al., 2020), whose likelihood requires integrating the costly probability-flow ODE through the network's divergence, with many steps needed for both accuracy and invertibility. Later work cuts training cost via flow matching but leaves this inference cost untouched (Klein et al., 2023b; Klein & Noé, 2024). FALCON instead distills into a few-step flow map, replacing the full ODE integration with a handful of Jacobian evaluations, at the price of only approximate invertibility (Rehman et al., 2026a). F2D2 drops

exact Jacobians entirely, jointly distilling the flow map and its density evolution via Hutchinson trace estimates (Ai et al., 2026; Hutchinson, 1990), and SCALLOP replaces this with a Hutchinson-free conditional divergence-matching loss (OuYang et al., 2026). Across these distillation-based methods, the learned density's approximation error enters the importance weights directly and persists regardless of proposal count. NF^{2}M instead evaluates its conditional proposal densities exactly.

Stochastic flow maps and intermediate correction. Distributional diffusion models learn the conditional distribution of clean states given noisy states, enabling larger denoising steps (Xiao et al., 2022; De Bortoli et al., 2025). Meta Flow Maps and Diamond Maps use clean-state candidates for intermediate correction toward reward-tilted distributions (Potaptchik et al., 2026; Holderrieth et al., 2026). Their implicit conditional generators do not directly provide the proposal densities needed for Boltzmann importance weights. NF^{2}M supplies them through a conditional normalizing flow.

5 Experiments

Benchmarks. We evaluate on the five all-atom peptide systems of Tan et al. (2025) reusing their MD training trajectories, splits, force fields, and evaluation protocol throughout so our numbers are directly comparable to theirs. We report the energy Wasserstein distance ($\mathcal{E}$-$\mathcal{W}_2$, sensitive to fine local structure) and the dihedral torus Wasserstein distance ($\mathbb{T}$-$\mathcal{W}_2$, sensitive to metastable-state occupancy), each against 10^4 held-out reference configurations over three seeds (App. B.1).

Baselines. We compare NF^{2}M (variants NF^{2}M-IS and NF^{2}M-MIS) against SNIS, the exact-Jacobian flow maps FALCON and FALCON-A (Rehman et al., 2026a), and the distilled-density flow maps F2D2 (Ai et al., 2026) and SCALLOP (OuYang et al., 2026). The flow-maps are distilled from the same per-system velocity teacher, which uses a DiT-style backbone (Peebles & Xie, 2023). Our SNIS baseline applies direct self-normalized importance sampling to an unconditional molecular flow proposal trained following Tan et al. (2025), without their annealed SMC procedure. We additionally compare against the NF-based SMC sampler from Tan et al. (2025) in Tab. 26: it does not consistently outperform the cheaper direct-SNIS baseline, which we retain in the main comparison. For each system, their total wall-clock training budget, including the teacher and flow-map pretraining, matches the training time of the selected NF^{2}M checkpoint (App. B.2).

Budgets. Proposal budgets otherwise follow each method's practical operating point: NF^{2}M draws its final 10^4 particles directly, while SNIS and the four flow-map baselines first draw 10^6 proposals and then resample 10^4 samples for evaluation (except FALCON/FALCON-A on chignolin, where exact per-step Jacobian costs cap the pool at 10^5). For NF^{2}M, we run 512 MAExL steps followed by 128 post-hoc MALA steps targeting the Boltzmann distribution. For all other methods, we apply enough post-hoc MALA steps to the final 10^4 samples to match NF^{2}M's total target-energy evaluations; all metrics use 10^4 final samples. Tab. 23 compares the cost of this refinement to the total cost of each method, and Tab. 25 reports the number of post-hoc MALA steps used for each method.

5.1 Main results

Tab. 1 reports the energy and dihedral Wasserstein distances, while Tab. 2 reports training and inference times. The systems are ordered by increasing size in both tables, from ALA-2 to chignolin.

Distributional accuracy across syste, scales. Tab. 1 shows that NF^{2}M-IS/MIS lead every system except ALA-6 on $\mathcal{E}$-$\mathcal{W}_2$. On the smallest systems ALA-2/ALA-3, SNIS achieves the lowest $\mathbb{T}$-$\mathcal{W}_2$, reflecting the closest match to the reference dihedral-angle distribution. However, SNIS's $\mathbb{T}$-$\mathcal{W}_2$ rises on the larger systems. In contrast, NF^{2}M's progressive correction maintains stronger performance, achieving the lowest $\mathbb{T}$-$\mathcal{W}_2$ on the largest systems, ALA-6 and chignolin, with FALCON and FALCON-A in between. FALCON achieves the lowest $\mathcal{E}$-$\mathcal{W}_2$ on ALA-6, where NF^{2}M does not lead on that metric. F2D2 and SCALLOP have substantially higher $\mathcal{E}$-$\mathcal{W}_2$ on ALA-4 and ALA-6.

We also ablate NF^{2}M's in-loop MAExL: it improves energy accuracy and stability with little change in torsion distances, while increasing from 512 to 1024 steps brings little further benefit for IS (Tab. 4). Separately, removing post-hoc MALA from all methods changes NF^{2}M's results relatively little but substantially worsens energy accuracy for several baselines (Tab. 24).

Table 1: Energy and torus Wasserstein distances on ALA-N and chignolin systems, from 3 random seeds, computed on the test split vs. 10^4 reference samples. Best **bold**, second underlined.

		SNIS	NF^{2}M-IS	NF^{2}M-MIS	FALCON	FALCON-A	F2D2	SCALLOP
ALA-2	$\mathcal{E}$-$\mathcal{W}_2$	0.768±0.031	**0.132**±0.013	0.242±0.080	0.759±0.047	0.595±0.329	0.890±0.030	0.874±0.033
	$\mathbb{T}$-$\mathcal{W}_2$	**0.242**±0.012	1.050±0.011	1.239±0.006	0.519±0.253	0.862±1.089	0.752±0.071	0.374±0.043
ALA-3	$\mathcal{E}$-$\mathcal{W}_2$	0.268±0.067	0.164±0.011	**0.157**±0.009	0.181±0.049	0.241±0.144	0.553±0.133	0.703±0.186
	$\mathbb{T}$-$\mathcal{W}_2$	**0.362**±0.019	0.490±0.016	0.465±0.007	0.463±0.170	1.100±0.706	1.049±0.476	0.847±0.196
ALA-4	$\mathcal{E}$-$\mathcal{W}_2$	1.515±0.351	**0.988**±0.027	1.107±0.013	1.763±0.192	1.985±0.737	2.748±2.096	2.981±0.970
	$\mathbb{T}$-$\mathcal{W}_2$	**0.777**±0.131	0.824±0.041	0.882±0.051	0.893±0.140	1.462±1.092	2.430±0.115	2.368±0.189
ALA-6	$\mathcal{E}$-$\mathcal{W}_2$	0.923±1.017	1.018±0.063	1.301±0.092	**0.610**±0.410	0.680±0.493	2.161±1.723	3.767±1.724
	$\mathbb{T}$-$\mathcal{W}_2$	1.767±0.675	**0.990**±0.026	1.027±0.021	1.716±0.332	1.557±0.025	2.927±0.288	3.126±0.462
Chignolin	$\mathcal{E}$-$\mathcal{W}_2$	7.748±5.290	**4.331**±0.262	6.112±0.129	4.807±1.066	5.028±3.624	4.793±2.335	4.770±1.076
	$\mathbb{T}$-$\mathcal{W}_2$	3.859±0.167	2.597±0.054	**2.581**±0.018	2.989±0.085	3.448±0.619	3.930±0.450	3.568±0.200

The accuracy-efficiency trade-off of flow-map methods. Tab. 2 shows that NF^{2}M-IS/MIS have comparable training and inference time to SNIS, although they require more training time on chignolin. FALCON, F2D2, and SCALLOP use different approximations of proposal densities, resulting in distinct accuracy-efficiency tradeoffs. F2D2 and SCALLOP predict density changes with a learned divergence head, with inference times comparable to NF^{2}M and SNIS on the ALA systems while roughly 1.6–2x slower on chignolin. FALCONs instead compute full flow-map Jacobians; without guaranteed invertibility, their determinants need not yield exact proposal densities.Their inference is roughly $11\times$ slower than SNIS on ALA-2 and 48–$129\times$ slower on ALA-3 through ALA-6. On chignolin, a full 10^6-proposal evaluation was projected at ~ 500 GPU-hours per seed (Sec. B.2.2), forcing a reduced pool of 10^5— still $\sim 60\times$ slower than SNIS or NF^{2}M-IS. We additionally show that under SNIS, distilled flow maps do not consistently outperform their velocity teacher (Tab. 22).By contrast, NF^{2}M computes its conditional proposal density exactly from the flow's tractable Jacobian.

Table 2: Inference time at the matched target-energy budget (sampling + post-hoc MALA ladder) and total training wall-clock across all required stages (mean ± std, 3 seeds). Chignolin FALCON/FALCON-A ran at a reduced 10^5-sample budget (vs. 10^6 elsewhere; 10^4 for NF^{2}M).

		SNIS	NF^{2}M-IS	NF^{2}M-MIS	FALCON	FALCON-A	F2D2	SCALLOP
ALA-2	Inf. (m)	2.0±0.0	2.3±0.2	2.1±0.2	24.2±0.0	24.1±0.1	0.8±0.0	0.8±0.0
	Train (h)	3.4	4.3	4.3	4.8	5.0	6.3	5.6
ALA-3	Inf. (m)	2.1±0.0	2.7±0.0	2.5±0.2	$\gtrsim 129$	$\gtrsim 129$	2.0±0.0	2.0±0.0
	Train (h)	4.7	6.0	6.0	7.1	7.2	6.5	5.8
ALA-4	Inf. (m)	2.8±0.1	3.1±0.2	2.7±0.2	$\gtrsim 202$	$\gtrsim 201$	2.4±0.0	2.4±0.0
	Train (h)	5.8	5.3	5.3	7.1	7.1	5.6	5.1
ALA-6	Inf. (m)	6.2±0.0	7.5±0.2	6.8±0.2	$\gtrsim 991$	$\gtrsim 992$	6.7±0.0	6.7±0.1
	Train (h)	11.7	12.8	10.2	15.1	15.1	10.3	9.8
Chig.	Inf. (m)	40.7±0.0	47.2±0.3	37.7±0.0	$\gtrsim 2820$	$\gtrsim 2818$	73.0±0.1	73.1±0.0
	Train (h)	13.3	21.7	21.7	24.4	24.0	22.2	23.4

NF^{2}M: IS versus MIS. NF^{2}M-IS and NF^{2}M-MIS perform similarly, with MIS using a smaller candidate budget. MIS reduces inference time most clearly on chignolin, with smaller differences on ALA systems. At matched candidate budgets, IS and MIS achieve comparable inner ESS, although both exhibit substantial weight concentration at low noise, especially on chignolin (Fig. 4).

Overall, NF^{2}M delivers strong performance at comparable training cost. Its inference time is broadly comparable to SNIS, F2D2, and SCALLOP and much shorter than FALCON and FALCON-A. Its performance advantage is most pronounced on the larger systems.

6 Conclusion

We presented NF^{2}M, a Boltzmann Generator that parameterizes a stochastic flow map with a conditional normalizing flow, correcting each denoising transition via tractable SNIS. This yields consistent samples with theoretical error control and strong empirical performance across peptide systems, with no approximate likelihood. Currently, NF^{2}M is trained per system; a natural next step is transferability across peptides as in Tan et al. (2025), learning a single conditional flow that generalizes across sequences and lengths. One other promising directions would be to do training with the reverse KL in Sec. 3.2 to further refining the model.

AI USE STATEMENT

We used generative AI tools to improve the writing quality of this manuscript (grammar, phrasing, and clarity), but not to generate full sections of the paper from scratch. All technical content, claims, and structure originate from the authors. We also used generative AI tools to assist in writing code for our experiments. All AI-assisted text and code were reviewed by the authors, and we take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide substantial implementation and experimental details in the appendix, including the conditional flow architectures (App. A.5), the molecular symmetry correction (App. A.4), the multiple importance sampling construction (App. A.2, the MCMC rejuvenation scheme (App. A.3), the proofs of our theoretical guarantees (App. A.1), and the full experimental setup and evaluation protocol (Apps. B.1 and B.2). We additionally attach our code as supplementary material to this submission.

REFERENCES

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. ISSN 08834237, 21688745. URL http://www.jstor.org/stable/26408299.

Xinyue Ai, Yutong He, Albert Gu, Ruslan Salakhutdinov, J Zico Kolter, Nicholas Matthew Boffi, and Max Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8uZ5UdIul2.

M. Albergo, N. M. Boffi, and E. 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.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Di5apl8HSH.

Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching, 2024. URL https://arxiv.org/abs/2402.08667.

Ivet Buch, Toni Giorgino, and Gianni De Fabritiis. High-throughput all-atom molecular dynamics simulations using distributed computing. *Journal of Chemical Information and Modeling*, 51(8): 1824–1830, 2011.

Monica F. Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M. Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017. doi: 10.1109/MSP.2017.2699226.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Patrick Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. 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. 12632–12676. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/de-bortoli25b.html.

Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and Boltzmann generators. *Phys. Rev. Res.*, 4:L042005, Oct 2022. doi: 10.1103/PhysRevResearch.4.L042005.

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.

Felix Draxler, Peter Sorrenson, Lea Zimmermann, Armand Rousselot, and Ullrich Köthe. Free-form flows: Make any architecture a normalizing flow. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2197–2205. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/draxler24a.html.

Alain Durmus and Éric Moulines. Quantitative bounds of convergence for geometrically ergodic markov chain in the wasserstein distance with application to the metropolis adjusted langevin algorithm. *Statistics and Computing*, 25(1):5–19, Jan 2015. ISSN 1573-1375. doi: 10.1007/s11222-014-9511-z. URL https://doi.org/10.1007/s11222-014-9511-z.

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

Víctor Elvira, Luca Martino, David Luengo, and Mónica F. Bugallo. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129 – 155, 2019. doi: 10.1214/18-STS668. URL https://doi.org/10.1214/18-STS668.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

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

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uWj4s7rMnR.

Louis Grenioux and Maxence Noble. Diffusion-based annealed boltzmann generators : benefits, pitfalls and hopes. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=la4FDaeIbw.

Louis Grenioux, Maxence Noble, Marylou Gabrié, and Alain Oliviero Durmus. Stochastic localization via iterative posterior sampling. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16337–16376. PMLR, 2024. URL https://proceedings.mlr.press/v235/grenioux24a.html.

Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Joshua Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *Advances in Neural Information Processing Systems*, 38:120986–121022, 2026.

Hera Y. He and Art B. Owen. Optimal mixture weights in multiple importance sampling, 2014. URL https://arxiv.org/abs/1411.3954.

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

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tAtpSwjsCB.

Shinya Honda, Kazuhiko Yamasaki, Yoshito Sawada, and Hisayuki Morii. 10 residue folded peptide designed by segment statistics. *Structure*, 12(8):1507–1518, 2004.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2078–2087. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/huang18d.html.

M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866. URL https://doi.org/10.1080/03610919008812866.

Khaled Kahouli, Romuald Elie, Klaus-Robert Müller, Quentin Berthet, Oliver T. Unke, and Arnaud Doucet. Control variate score matching for diffusion models, 2025. URL https://arxiv.org/abs/2512.20003.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

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

Leon Klein, Andrew Y. K. Foong, Tor Erlend Fjelde, Bruno Kacper Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=EjMLpTgvKH.

Leon Klein, Andreas Krämer, and Frank Noe. Equivariant flow matching. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 59886–59910. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/bc827452450356f9f558f4e4568d553b-Paper-Conference.pdf.

Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In Hal Daumé III and Aarti Singh (eds.), *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.

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Bálint Máté and François Fleuret. Learning interpolations between boltzmann densities. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=TH6YrEcbth.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics*, 38(5):145:1–145:19, 2019.

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.

RuiKang OuYang, Hanlin Yu, Xinyue Ai, Yutong He, Nicholas Matthew Boffi, Pradeep Ravikumar, José Miguel Hernández-Lobato, Max Simchowitz, Benjamin Kurt Miller, and Omar Chehab. Few-step boltzmann generators via scalable likelihood flow maps. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026. URL https://openreview.net/forum?id=mfdTHHRIid.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2669533.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K5gV8Yptne.

Danyal Rehman, Tara Akhound-Sadegh, Artem Gazizov, Yoshua Bengio, and Alexander Tong. FALCON: Few-step accurate likelihoods for continuous flows. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=FbssShlI4N.

Danyal Rehman, Charlie B. Tan, Yoshua Bengio, Joey Bose, and Alexander Tong. Autoregressive boltzmann generators. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=75AYDsndHP.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998. doi: https://doi.org/10.1111/1467-9868.00123. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00123.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth 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.

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pp. 419–428, New York, NY, USA, 1995. Association for Computing Machinery. ISBN 0897917014. doi: 10.1145/218380.218498. URL https://doi.org/10.1145/218380.218498.

Peter Wirnsberger, Jonas Adler, Jonas Köhler, and Frank Noé. Targeted free energy perturbation. *Proceedings of the National Academy of Sciences*, 117(18):9902–9908, 2020.

Dawn B. Woodard, Scott C. Schmidler, and Mark Huber. Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electronic Journal of Probability*, 14:780–804, 2009.

Hao Wu, Jonas Köhler, and Frank Noe. Stochastic normalizing flows. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5933–5944. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/41d80bfc327ef980528426fc810a6d7a-Paper.pdf.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JprM0p-q0Co.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

James Matthew Young, Paula Cordero-Encinar, Sebastian Reich, Andrew Duncan, and O. Deniz Akyildiz. Diffusion path samplers via sequential monte carlo, 2026. URL https://arxiv.org/abs/2601.21951.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. 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. 74348–74369. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhai25d.html.

Fengzhe Zhang, Jiajun He, Laurence I Midgley, Javier Antorán, and José Miguel Hernández-Lobato. Efficient and unbiased sampling of boltzmann distributions via consistency models. *arXiv preprint arXiv:2409.07323*, 2024.

Fengzhe Zhang, Laurence Illing Midgley, and José Miguel Hernández-Lobato. Efficient and unbiased sampling from boltzmann distributions via variance-tuned diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Jq2dcMCS5R. J2C Certification.

Bridging Stochastic Flow Maps and Boltzmann Generators with Normalizing Flows Appendix

A Details on the method **16**

A.1 Theoretical details and proofs 16

A.1.1 Controlling the per-step error 17

A.1.2 Adding an MCMC rejuvenation step 19

A.1.3 From per-step bounds to a path-error bound 24

A.2 Multiple Importance Sampling 26

A.3 Metropolis-Adjusted Exponential Langevin 31

A.4 Centre-of-mass adjustment for the conditional posterior 35

A.4.1 Setup and notation 35

A.4.2 Relation to SBG's unconditional adjustment 36

A.4.3 The conditional Gaussian lift 36

A.4.4 Importance weights and effective log-likelihood 37

A.4.5 When centroid matching improves ESS 37

A.5 Conditional TarFlow and StarFlow 38

A.5.1 TarFlow and StarFlow 38

A.5.2 Making the flow conditional on (x_t, t) 39

A.5.3 The conditioning cache 40

A.5.4 Computational complexity 40

B Experimental details **42**

B.1 Datasets and Metrics 42

B.2 Hyperparameters 44

B.2.1 Training 44

B.2.2 Inference 48

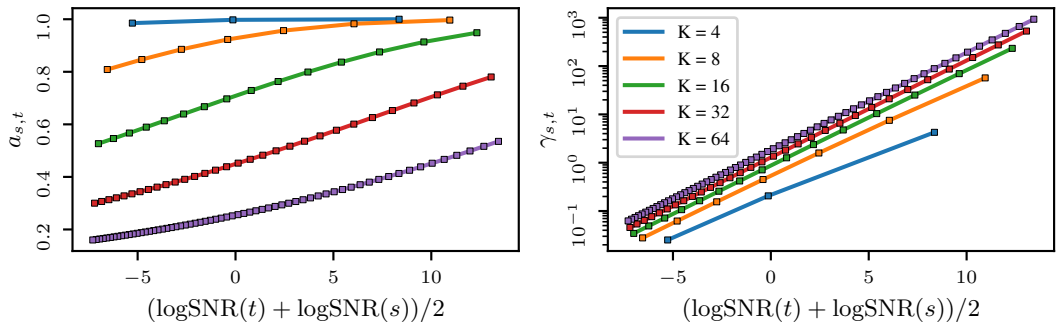


Figure 2: Values of $a_{s,t}$ and $\gamma_{s,t}$ along the time grid of Karras et al. (2022).

A Details on the method

A.1 Theoretical details and proofs

Notation. For a probability measure π and a π-measurable function f, we write $\pi(f) = \mathbb{E}_\pi[f(X)] = \int f(x)\,\pi(\mathrm{d}x)$. For any bounded function f and any $B \subset \mathbb{R}^d$, we set

$$[f]_B = \sup_{x,x' \in B} |f(x') - f(x)|.$$

We say that a probability distribution π on $\mathbb{R}^d$ has *sub-Gaussian tails with center m and rate κ* if for all $u > 0$,

$$\pi\left(\|Z - m\| > \kappa(\sqrt{d} + u)\right) \le \exp\left(-u^2\right). \tag{21}$$

Bridges and the operator $G_{s|t,x_t}$. Throughout, $p_{s|t,0}(x_s|x_t, x_0)$ denotes the Gaussian bridge density. We write $\alpha_{s,t} = a_{s,t}$ for the clean-state coefficient used in the main text and use the standard parametrization

$$\mathcal{L}\left(X_s \mid X_t = x_t, X_0 = x_0\right) = \mathcal{N}(\alpha_{s,t}x_0 + b_{s,t}x_t,\, \Sigma_{s,t}), \qquad \Sigma_{s,t} = \sigma^2_{s,t}\,\mathrm{I}_d,$$

and define the *bridge sensitivity* (see Fig. 2)

$$\gamma_{t,s} = \frac{\alpha_s \sigma_{t|s}}{\sigma_t \sigma_s} = \frac{\alpha_{s,t}}{\sigma_{s,t}}. \tag{22}$$

Writing $A_t = \alpha_t^2/\sigma_t^2$ for the signal-to-noise ratio at time t and using $\sigma^2_{t|s} = \sigma_t^2 - (\alpha_t/\alpha_s)^2\sigma_s^2$, the bridge sensitivity is the square root of an SNR increment,

$$\gamma^2_{t,s} = \frac{\alpha_s^2}{\sigma_s^2} - \frac{\alpha_t^2}{\sigma_t^2} = A_s - A_t. \tag{23}$$

Since $t \mapsto A_t$ is nonincreasing, $\gamma_{t,s} \le \sqrt{A_s} = \alpha_s/\sigma_s$, which is finite for every $s > 0$ and diverges as $s \to 0$, the bridge degenerating to a point mass. All schedules below therefore stop at a strictly positive time, see Remark 18. For any p_s-measurable ϕ, we let

$$(G_{s|t,x_t}\phi)(x_0) = \int \phi(x_s)\, p_{s|t,0}(x_s \mid x_t, x_0)\, \mathrm{d}x_s,$$

so that

$$p_{s|t}(\phi \mid x_t) = \int \phi(x_s)\, p_{s|t,0}(x_s \mid x_t, x_0)\, p_{0|t}(x_0 \mid x_t)\, \mathrm{d}x_s\, \mathrm{d}x_0 = p_{0|t}(G_{s|t,x_t}\phi \mid x_t).$$

SNIS approximations. We assume access to a proposal $q_{0|t}$ that approximates $p_{0|t}$. Drawing $x^m_{0|t} \overset{\text{iid}}{\sim} q_{0|t}(\cdot|x_t)$ for $m = 1, \dots, M$ and forming unnormalized importance weights

$$\tilde{w}^m_{0|t} = \frac{p_{t|0}(x_t \mid x^m_{0|t})\, p(x^m_{0|t})}{q_{0|t}(x^m_{0|t} \mid x_t)}, \qquad w^m_{0|t} = \frac{\tilde{w}^m_{0|t}}{\sum_{i=1}^M \tilde{w}^i_{0|t}},$$

the SNIS approximation of $p_{0|t}$ is

$$p^M_{0|t}(x_0 \mid x_t) = \sum_{m=1}^{M} w^m_{0|t}\, \delta_{x^m_{0|t}}(x_0),$$

and the induced SNIS approximation of $p_{s|t}$ is

$$p^M_{s|t}(x_s \mid x_t) = \sum_{m=1}^{M} w^m_{0|t}\, p_{s|t,0}(x_s \mid x_t, x^m_{0|t}).$$

By the identity $p_{s|t}(\phi|x_t) = p_{0|t}(G_{s|t,x_t}\phi|x_t)$, we have $p^M_{s|t}(\phi|x_t) = p^M_{0|t}(G_{s|t,x_t}\phi|x_t)$.

Standing assumption (per time t). For a fixed $0 \le s < t \le T$ and $x_t \in \mathbb{R}^d$, we say that the *sub-Gaussian tail assumption* holds at (t, x_t) if both $q_{0|t}(\cdot|x_t)$ and the normalized square-weight measure proportional to $\tilde{w}^2_{0|t}\, q_{0|t}(\cdot|x_t)$ have sub-Gaussian tails (in the sense of (21)) with common center $m_t(x_t)$ and rate $\kappa_t(x_t)$. We further denote the chi-squared divergence

$$\rho_t(x_t) = 1 + \chi^2\left(p_{0|t}(\cdot \mid x_t) \,\middle\|\, q_{0|t}(\cdot \mid x_t)\right).$$

A.1.1 Controlling the per-step error

We begin with a standard Wasserstein contraction property of the bridge.

Proposition 4 *Let $p \ge 1$, $x_t \in \mathbb{R}^d$, and assume that both $p_{0|t}(\cdot|x_t)$ and $q_{0|t}(\cdot|x_t)$ have finite p-th moment. Then*

$$W_p\left(p_{s|t}(\cdot \mid x_t),\, q_{s|t}(\cdot \mid x_t)\right) \le \alpha_{s,t}\, W_p\left(p_{0|t}(\cdot \mid x_t),\, q_{0|t}(\cdot \mid x_t)\right).$$

Proof. Let Π be an optimal W_p-coupling of $p_{0|t}(\cdot|x_t)$ and $q_{0|t}(\cdot|x_t)$, and draw $(X_0, X'_0) \sim \Pi$. Independently, draw $\xi \sim \mathcal{N}(0, \Sigma_{s,t})$, and set

$$X_s = \alpha_{s,t} X_0 + b_{s,t} x_t + \xi, \qquad X'_s = \alpha_{s,t} X'_0 + b_{s,t} x_t + \xi.$$

Then (X_s, X'_s) is a coupling of $p_{s|t}(\cdot|x_t)$ and $q_{s|t}(\cdot|x_t)$, and

$$\|X_s - X'_s\| = \alpha_{s,t} \|X_0 - X'_0\|.$$

Taking L^p-norms gives the result. □

We next sharpen the SNIS bias and MSE bound of (Agapiou et al., 2017, Theorem 2.1) by localizing the test function to a set B where the proposal places most of its mass.

Lemma 5 *Let X be a Polish space and let μ and ν be probability distributions on X with $\mu \ll \nu$, and let $w = \mathrm{d}\mu/\mathrm{d}\nu$. Set $\rho = 1 + \chi^2(\mu \,\|\, \nu) = \nu(w^2)$, and for any measurable $B \subset \mathsf{X}$ define*

$$\eta_{B^c} = \int_{B^c} w(x)^2\, \nu(\mathrm{d}x).$$

For $N \ge 1$, let μ^N be the SNIS approximation of μ with proposal ν,

$$\mu^N = \sum_{i=1}^{N} \frac{w(X^i)}{\sum_{j=1}^{N} w(X^j)}\, \delta_{X^i}, \qquad X^i \overset{\text{iid}}{\sim} \nu.$$

Then, for any bounded μ-measurable function f,

$$\left|\mathbb{E}\left[\mu^N(f) - \mu(f)\right]\right| \le \frac{6\rho}{N}\, [f]_B + 2\|f\|_\infty \sqrt{\eta_{B^c}\, \nu(B^c)} + 4\|f\|_\infty\, N\, \nu(B^c), \tag{24}$$

$$\mathbb{E}\left[\left(\mu^N(f) - \mu(f)\right)^2\right] \le \frac{2\rho}{N}\, [f]^2_B + 8\|f\|^2_\infty\, \eta_{B^c}\, \nu(B^c) + 32\|f\|^2_\infty\, N\, \nu(B^c). \tag{25}$$

Proof. Set $c_B = (\sup_B f - \inf_B f)/2$ and define the truncation

$$f_B(x) = \begin{cases} f(x) & \text{if } x \in B, \\ c_B & \text{otherwise,} \end{cases}$$

so that $\|f_B - c_B\|_\infty = [f]_B/2$. Applying (Agapiou et al., 2017, Theorem 2.1) to $(f_B - c_B)/([f]_B/2)$ and using linearity yields

$$\left|\mathbb{E}\left[\mu^N(f_B) - \mu(f_B)\right]\right| \le \frac{6\rho}{N}\,[f]_B, \qquad \mathbb{E}\left[\left(\mu^N(f_B) - \mu(f_B)\right)^2\right] \le \frac{2\rho}{N}\,[f]_B^2.$$

Now let $h_B = f - f_B$; then $h_B = 0$ on B and $\|h_B\|_\infty \le 2\|f\|_\infty$. Denote the event $E_B = \bigcap_{i=1}^N \{X^i \in B\}$. On E_B, all atoms of μ^N lie in B, so $\mu^N(h_B) = 0$ and only the deterministic term $\mu(h_B)$ contributes. We bound the latter using Cauchy–Schwarz:

$$\begin{aligned} |\mu(h_B)| &\le 2\|f\|_\infty \int_{B^\mathsf{c}} \mu(\mathrm{d}x) \\ &\le 2\|f\|_\infty \sqrt{\int_{B^\mathsf{c}} w(x)^2\,\nu(\mathrm{d}x)}\sqrt{\int_{B^\mathsf{c}} \nu(\mathrm{d}x)} \\ &= 2\|f\|_\infty \sqrt{\eta_{B^\mathsf{c}}\,\nu(B^\mathsf{c})}. \end{aligned}$$

On E_B^c, we use the crude bound $\left|\mu^N(h_B) - \mu(h_B)\right| \le 4\|f\|_\infty$. By the union bound,

$$\mathbb{P}(E_B^\mathsf{c}) \le N\,\nu(B^\mathsf{c}).$$

Combining,

$$\mathbb{E}\left[\left|\mu^N(h_B) - \mu(h_B)\right|\right] \le 2\|f\|_\infty \sqrt{\eta_{B^\mathsf{c}}\nu(B^\mathsf{c})} + 4\|f\|_\infty\, N\,\nu(B^\mathsf{c}),$$

and similarly $\mathbb{E}\left[\left(\mu^N(h_B) - \mu(h_B)\right)^2\right] \le 4\|f\|_\infty^2\,\eta_{B^\mathsf{c}}\nu(B^\mathsf{c}) + 16\|f\|_\infty^2\,N\,\nu(B^\mathsf{c})$. The decomposition $\mu^N(f) - \mu(f) = \mu^N(f_B) - \mu(f_B) + \mu^N(h_B) - \mu(h_B)$ together with $(a+b)^2 \le 2(a^2+b^2)$ then yields (24)–(25). □

We now control the oscillation of $G_{s|t,x_t}\psi$ on a ball.

Lemma 6 *Let $x_t \in \mathbb{R}^d$ and $0 \le s < t \le T$. For any bounded test function ψ, $\left\|G_{s|t,x_t}\psi\right\|_\infty \le \|\psi\|_\infty$, and for all $x, x' \in \mathbb{R}^d$,*

$$\left|G_{s|t,x_t}\psi(x) - G_{s|t,x_t}\psi(x')\right| \le 2\|\psi\|_\infty \left[2\Phi\left(\frac{\gamma_{t,s}\|x - x'\|}{2}\right) - 1\right], \tag{26}$$

where Φ is the standard normal cumulative distribution function and $\gamma_{t,s}$ is defined in (22)*. Consequently, for $B_u = \mathcal{B}(m, \kappa(\sqrt{d} + u))$ with $m \in \mathbb{R}^d$, $\kappa > 0$ and $u > 0$,*

$$[G_{s|t,x_t}\psi]_{B_u} \le 2\|\psi\|_\infty \min\left\{1,\ \sqrt{\tfrac{2}{\pi}}\,\gamma_{t,s}\,\kappa\left(\sqrt{d} + u\right)\right\}. \tag{27}$$

Proof. The sup-norm bound is immediate since $G_{s|t,x_t}$ is an expectation operator. For (26), let $x, x' \in \mathbb{R}^d$. Then

$$\left|G_{s|t,x_t}\psi(x) - G_{s|t,x_t}\psi(x')\right| \le \|\psi\|_\infty \int \left|p_{s|t,0}(x_s \mid x_t, x) - p_{s|t,0}(x_s \mid x_t, x')\right| \mathrm{d}x_s.$$

The right-hand side equals $2\|\psi\|_\infty\ \mathrm{TV}(p_{s|t,0}(\cdot|x_t, x),\, p_{s|t,0}(\cdot|x_t, x'))$. The bridge densities are Gaussians with the same covariance $\Sigma_{s,t} = \sigma_{s,t}^2 \mathrm{I}_d$ and means differing by $\alpha_{s,t}(x - x')$, so by the standard identity

$$\mathrm{TV}\left(\mathcal{N}(m_1, \sigma^2\mathrm{I}_d), \mathcal{N}(m_2, \sigma^2\mathrm{I}_d)\right) = 2\Phi\left(\frac{\|m_1 - m_2\|}{2\sigma}\right) - 1,$$

together with $\alpha_{s,t}/\sigma_{s,t} = \gamma_{t,s}$, we obtain (26).

For (27), let $x, x' \in B_u$, so that $\|x - x'\| \le \operatorname{diam}(B_u) = 2\kappa(\sqrt{d} + u)$. Since Φ is nondecreasing, (26) gives

$$[G_{s|t,x_t}\psi]_{B_u} \le 2\|\psi\|_\infty \left[2\Phi\left(\gamma_{t,s}\,\kappa(\sqrt{d}+u)\right) - 1\right].$$

Finally $2\Phi(r) - 1 \le 1$ for every $r \ge 0$, and, with φ the standard normal density,

$$2\Phi(r) - 1 = \int_{-r}^{r} \varphi(z)\,\mathrm{d}z \le 2r\,\varphi(0) = \sqrt{\tfrac{2}{\pi}}\,r,$$

so that $2\Phi(r) - 1 \le \min\{1, \sqrt{2/\pi}\,r\}$, which yields (27). □

Combining the previous two lemmas yields the main per-step bound.

Theorem 7 *Fix $0 < s < t \le T$ and $x_t \in \mathbb{R}^d$, and assume the sub-Gaussian tail assumption holds at (t, x_t) with center $m_t(x_t)$ and rate $\kappa_t(x_t)$. Write $u_M = \sqrt{3\log M}$ and*

$$\Theta_{t,s}(u) = \min\left\{1,\ \sqrt{\tfrac{2}{\pi}}\,\gamma_{t,s}\,\kappa_t(x_t)\left(\sqrt{d}+u\right)\right\}. \tag{28}$$

Then, for any bounded test function ψ,

$$\|\psi\|_\infty^{-1}\left|\mathbb{E}\left[p^M_{s|t}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \le \frac{12\,\rho_t(x_t)}{M}\,\Theta_{t,s}(u_M) + \frac{2\sqrt{\rho_t(x_t)}}{M^3} + \frac{4}{M^2},$$
$$\|\psi\|_\infty^{-2}\,\mathbb{E}\left[\left(p^M_{s|t}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right)^2\right] \le \frac{8\,\rho_t(x_t)}{M}\,\Theta_{t,s}(u_M)^2 + \frac{8\,\rho_t(x_t)}{M^6} + \frac{32}{M^2}.$$

In particular, bounding $\Theta_{t,s}$ by its linear branch,

$$\|\psi\|_\infty^{-1}\left|\mathbb{E}\left[p^M_{s|t}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \le \frac{24\,\rho_t(x_t)}{M\sqrt{2\pi}}\,\gamma_{t,s}\,\kappa_t(x_t)\left(\sqrt{d}+u_M\right) + \frac{2\sqrt{\rho_t(x_t)}}{M^3} + \frac{4}{M^2},$$
$$\|\psi\|_\infty^{-2}\,\mathbb{E}\left[\left(p^M_{s|t}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right)^2\right] \le \frac{16\,\rho_t(x_t)}{\pi M}\,\gamma_{t,s}^2\,\kappa_t(x_t)^2\left(\sqrt{d}+u_M\right)^2 + \frac{8\,\rho_t(x_t)}{M^6} + \frac{32}{M^2},$$

while bounding it by its constant branch gives an $O(\rho_t(x_t)/M)$ bound valid uniformly in $\gamma_{t,s}$.

Proof. Setting $f = G_{s|t,x_t}\psi$, the identity $p^M_{s|t}(\psi|x_t) = p^M_{0|t}(f|x_t)$ allows us to apply Lemma 5 with $\mu = p_{0|t}(\cdot|x_t)$ and $\nu = q_{0|t}(\cdot|x_t)$. For $u > 0$, take $B_u = \mathcal{B}(m_t(x_t),\, \kappa_t(x_t)(\sqrt{d}+u))$. By the sub-Gaussian tail assumption,

$$q_{0|t}(B_u^{\mathsf{c}} \mid x_t) \le e^{-u^2}, \qquad \eta_{B_u^{\mathsf{c}}} \le \rho_t(x_t)\,e^{-u^2}, \qquad \eta_{B_u^{\mathsf{c}}}\,q_{0|t}(B_u^{\mathsf{c}} \mid x_t) \le \rho_t(x_t)\,e^{-2u^2}.$$

Using (27) to control $[f]_{B_u}$ and $\|f\|_\infty \le \|\psi\|_\infty$ in Lemma 5 gives

$$\left|\mathbb{E}\left[p^M_{s|t}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \le \frac{12\,\rho_t(x_t)}{M}\,\|\psi\|_\infty\,\Theta_{t,s}(u) + 2\|\psi\|_\infty\sqrt{\rho_t(x_t)}\,e^{-u^2} + 4\|\psi\|_\infty\,M\,e^{-u^2},$$

and similarly for the MSE. Choosing $u = u_M = \sqrt{3\log M}$ (so $e^{-u^2} = M^{-3}$) yields the stated bounds. □

A.1.2 Adding an MCMC rejuvenation step

We now study the same SNIS estimator equipped with an additional rejuvenation move. Let R_{t,x_t} be a Markov kernel leaving $p_{0|t}(\cdot|x_t)$ invariant. We define the *rejuvenated* estimator

$$p^{M,L}_{s|t}(x_s \mid x_t) = \sum_{m=1}^{M} w^m_{0|t}\,p_{s|t,0}(x_s \mid x_t, y^m_{0|t}),$$
$$y^m_{0|t} \sim R^L_{t,x_t}(x^m_{0|t}, \cdot), \quad x^m_{0|t} \overset{\text{iid}}{\sim} q_{0|t}(\cdot \mid x_t),$$

where $L \geq 0$ is the number of MCMC steps. The bridge is integrated out in $p^{M,L}_{s|t}$, but the MCMC randomness is not: the $y^m_{0|t}$ enter through their realized values. Alongside it we introduce the *MCMC-integrated* estimator, defined for any bounded ψ by

$$\bar{p}^{M,L}_{s|t}(\psi \mid x_t) = p^M_{0|t}\left(R^L_{t,x_t} G_{s|t,x_t}\psi \mid x_t\right) = \sum_{m=1}^{M} w^m_{0|t}\, \left(R^L_{t,x_t} G_{s|t,x_t}\psi\right)(x^m_{0|t}), \tag{29}$$

in which the MCMC move is averaged rather than sampled. The two are related as follows: since $w^m_{0|t}$ is $\sigma(x^{1:M}_{0|t})$-measurable and the $y^m_{0|t}$ are conditionally independent given $x^{1:M}_{0|t}$,

$$\mathbb{E}\left[p^{M,L}_{s|t}(\psi \mid x_t) \mid x^{1:M}_{0|t}\right] = \bar{p}^{M,L}_{s|t}(\psi \mid x_t), \tag{30}$$

the identity holding in conditional expectation and *not* pathwise. Together with the invariance identity $p_{s|t}(\psi|x_t) = p_{0|t}(R^L_{t,x_t} G_{s|t,x_t}\psi|x_t)$, (30) shows that the two estimators have the same bias, so every bias bound below applies verbatim to the sampled estimator $p^{M,L}_{s|t}$. Their mean squared errors differ by the conditional variance

$$\mathrm{Var}\left[p^{M,L}_{s|t}(\psi \mid x_t) \mid x^{1:M}_{0|t}\right] = \sum_{m=1}^{M} \left(w^m_{0|t}\right)^2 \mathrm{Var}_{R^L_{t,x_t}(x^m_{0|t},\cdot)}\left[G_{s|t,x_t}\psi\right], \tag{31}$$

which does not vanish as $L \to \infty$. We therefore state the MSE bounds of this subsection for $\bar{p}^{M,L}_{s|t}$ and treat the sampled estimator separately in Remark 12.

Remark 8 (Scope of the invariance assumption) *Our results assume R_{t,x_t} leaves $p_{0|t}(\cdot|x_t)$ invariant, which corresponds to MAExL at a* fixed *step size. The production sampler of App. A.3 adapts the step size online from the acceptance history, and such a chain need not be exactly invariant at finite L. The analysis is however insensitive to how the step size is chosen as long as it is not a function of the current state: if $(h_\ell)_{\ell \leq L}$ is generated independently of the chain (frozen, or adapted on a pilot run), then conditionally on $(h_\ell)_{\ell \leq L}$ the composition $R^{(h_1)}_{t,x_t} \cdots R^{(h_L)}_{t,x_t}$ is again invariant, Prop. 9 applies with $\Delta_{t,L}$ replaced by its conditional counterpart, and averaging over $(h_\ell)_{\ell \leq L}$ preserves both bounds. Quantifying the invariance defect of the fully state-dependent scheme at finite L is beyond our scope.*

Proposition 9 (SNIS with μ-invariant MCMC rejuvenation) *In the setting of Lemma 5, let R be a Markov kernel on $\mathbb{R}^d$ leaving μ invariant, and for $L \geq 0$ define the L-step rejuvenated SNIS estimator*

$$\mu^{N,L}(f) = \sum_{i=1}^{N} \frac{w(X^i)}{\sum_{j=1}^{N} w(X^j)} (R^L f)(X^i), \qquad X^i \overset{\text{iid}}{\sim} \nu.$$

Then, for any bounded μ-measurable function f and any $B \subset \mathbb{R}^d$,

$$\left|\mathbb{E}\left[\mu^{N,L}(f) - \mu(f)\right]\right| \leq \frac{6\rho}{N}\,[R^L f]_B + 2\|f\|_\infty \sqrt{\eta_{B^c}\nu(B^c)} + 4\|f\|_\infty\, N\,\nu(B^c), \tag{32}$$

$$\mathbb{E}\left[\left(\mu^{N,L}(f) - \mu(f)\right)^2\right] \leq \frac{2\rho}{N}\,[R^L f]_B^2 + 8\|f\|_\infty^2\, \eta_{B^c}\nu(B^c) + 32\|f\|_\infty^2\, N\,\nu(B^c). \tag{33}$$

Moreover, defining the local Wasserstein diameter

$$\Delta_L(B) = \sup_{x,x' \in B} W_1\left(R^L(x,\cdot),\, R^L(x',\cdot)\right),$$

if f is Lipschitz then $[R^L f]_B \leq \mathrm{Lip}(f)\,\Delta_L(B)$, and (32)–(33) *give*

$$\left|\mathbb{E}\left[\mu^{N,L}(f) - \mu(f)\right]\right| \leq \frac{6\rho}{N}\,\mathrm{Lip}(f)\Delta_L(B) + 2\|f\|_\infty \sqrt{\eta_{B^c}\nu(B^c)} + 4\|f\|_\infty\, N\,\nu(B^c), \tag{34}$$

$$\mathbb{E}\left[\left(\mu^{N,L}(f) - \mu(f)\right)^2\right] \leq \frac{2\rho}{N}\,\mathrm{Lip}(f)^2\Delta_L(B)^2 + 8\|f\|_\infty^2\, \eta_{B^c}\nu(B^c) + 32\|f\|_\infty^2\, N\,\nu(B^c). \tag{35}$$

Proof. Set $f_L = R^L f$. Since R leaves μ invariant, $\mu(f_L) = \mu(R^L f) = \mu(f)$, hence

$$\mu^{N,L}(f) - \mu(f) = \mu^N(f_L) - \mu(f_L),$$

where μ^N is the standard SNIS approximation of μ with proposal ν applied to f_L. Applying Lemma 5 to f_L (and using $\left\|R^L f\right\|_\infty \le \|f\|_\infty$) gives (32)–(33).

For the Lipschitz refinement, fix $x, x' \in B$ and let Π be any coupling of $R^L(x,\cdot)$ and $R^L(x',\cdot)$. Then

$$\left|R^L f(x) - R^L f(x')\right| \le \int |f(y) - f(y')|\,\Pi(\mathrm{d}y,\mathrm{d}y') \le \mathrm{Lip}(f)\int \|y-y'\|\,\Pi(\mathrm{d}y,\mathrm{d}y').$$

Optimizing over Π and taking the supremum over $x, x' \in B$ gives $[R^L f]_B \le \mathrm{Lip}(f)\,\Delta_L(B)$. □

The next proposition refines Lemma 6 by showing that the MCMC step contracts the effective x_0-diameter seen by the bridge.

Proposition 10 (MCMC-smoothed kernel sensitivity) *Let $x_t \in \mathbb{R}^d$, $0 < s < t \le T$, and let R_t be a Markov kernel on $\mathbb{R}^d$. For $L \ge 0$ and $B \subset \mathbb{R}^d$, set*

$$\Delta_{t,L}(B) = \sup_{x,x'\in B} W_1\left(R_t^L(x,\cdot),\, R_t^L(x',\cdot)\right).$$

Then, for any bounded test function ψ,

$$\left\|R_t^L G_{s|t,x_t}\psi\right\|_\infty \le \|\psi\|_\infty,$$

and

$$[R_t^L G_{s|t,x_t}\psi]_B \le 2\|\psi\|_\infty\left[2\Phi\left(\frac{\gamma_{t,s}\,\Delta_{t,L}(B)}{2}\right) - 1\right] \tag{36}$$

$$\le \|\psi\|_\infty \min\left\{2,\ \sqrt{\tfrac{2}{\pi}}\,\gamma_{t,s}\,\Delta_{t,L}(B)\right\}. \tag{37}$$

In particular, if $\Delta_{t,L}(B) \le a_{t,L}(B)\ \mathrm{diam}(B)$ with $a_{t,L}(B) \in [0,1]$, then

$$[R_t^L G_{s|t,x_t}\psi]_B \le \sqrt{\tfrac{2}{\pi}}\,\|\psi\|_\infty\,\gamma_{t,s}\,a_{t,L}(B)\ \mathrm{diam}(B);$$

and if there exist $C_t < \infty$ and $r_t \in (0,1)$ such that $W_1(R_t^L(x,\cdot), R_t^L(x',\cdot)) \le C_t\, r_t^L\, \|x - x'\|$ for all $x, x' \in B$, then

$$[R_t^L G_{s|t,x_t}\psi]_B \le \sqrt{\tfrac{2}{\pi}}\,\|\psi\|_\infty\,\gamma_{t,s}\,C_t\,r_t^L\ \mathrm{diam}(B).$$

Proof. The sup-norm bound follows because both R_t^L and $G_{s|t,x_t}$ are expectation operators. For the oscillation, let $x, x' \in B$ and let Π be any coupling of $R_t^L(x,\cdot)$ and $R_t^L(x',\cdot)$. Then

$$\left|R_t^L G_{s|t,x_t}\psi(x) - R_t^L G_{s|t,x_t}\psi(x')\right| \le \int \left|G_{s|t,x_t}\psi(y) - G_{s|t,x_t}\psi(y')\right|\Pi(\mathrm{d}y,\mathrm{d}y'),$$

and by (26) the integrand is at most $2\|\psi\|_\infty[2\Phi(\gamma_{t,s}\|y-y'\|/2)-1]$. The map $r \mapsto 2\Phi(\gamma_{t,s}r/2)-1$ is nondecreasing and concave on $\mathbb{R}_+$, so Jensen's inequality gives

$$\int\left[2\Phi\left(\frac{\gamma_{t,s}\|y-y'\|}{2}\right)-1\right]\Pi(\mathrm{d}y,\mathrm{d}y') \le 2\Phi\left(\frac{\gamma_{t,s}}{2}\int\|y-y'\|\Pi(\mathrm{d}y,\mathrm{d}y')\right)-1.$$

Optimizing over Π and taking the supremum over $x, x' \in B$ yields (36); the inequality $2\Phi(r)-1 \le \min\{1, \sqrt{2/\pi}\,r\}$ for $r \ge 0$ then gives (37). □

Theorem 11 (Per-step error with MCMC rejuvenation) *Fix $0 < s < t \le T$ and $x_t \in \mathbb{R}^d$, and assume the sub-Gaussian tail assumption holds at (t, x_t) with center $m_t(x_t)$ and rate $\kappa_t(x_t)$. For $u > 0$, set*

$$B_u = \mathcal{B}\left(m_t(x_t),\, \kappa_t(x_t)(\sqrt{d}+u)\right), \qquad \Delta_{t,L}(u) = \Delta_{t,L}(B_u).$$

Then, for any bounded test function ψ, the bias of the sampled estimator $p_{s|t}^{M,L}$ and the mean squared error of the MCMC-integrated estimator $\bar{p}_{s|t}^{M,L}$ of (29) *satisfy*

$$\|\psi\|_\infty^{-1}\left|\mathbb{E}\left[p_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \le \frac{12\,\rho_t(x_t)}{M}\left[2\Phi\left(\frac{\gamma_{t,s}\,\Delta_{t,L}(u)}{2}\right)-1\right] + 2\sqrt{\rho_t(x_t)}\,e^{-u^2} + 4M\,e^{-u^2}, \tag{38}$$

$$\|\psi\|_\infty^{-2}\,\mathbb{E}\left[\left(\bar{p}_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right)^2\right] \le \frac{8\,\rho_t(x_t)}{M}\left[2\Phi\left(\frac{\gamma_{t,s}\,\Delta_{t,L}(u)}{2}\right)-1\right]^2 + 8\,\rho_t(x_t)\,e^{-2u^2} + 32M\,e^{-u^2}. \tag{39}$$

Bounding $2\Phi(r) - 1 \le \min\{1, \sqrt{2/\pi}\, r\}$ *and writing*

$$\Lambda_{t,L}(u) = \min\left\{1,\ \tfrac{1}{\sqrt{2\pi}}\, \gamma_{t,s}\, \Delta_{t,L}(u)\right\}, \tag{40}$$

$$\|\psi\|_\infty^{-1} \left| \mathbb{E}\left[p_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t) \right] \right| \le \frac{12\, \rho_t(x_t)}{M}\, \Lambda_{t,L}(u) + 2\sqrt{\rho_t(x_t)}\, e^{-u^2} + 4M\, e^{-u^2}, \tag{41}$$

$$\|\psi\|_\infty^{-2}\, \mathbb{E}\left[\left(\bar{p}_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t) \right)^2 \right] \le \frac{8\, \rho_t(x_t)}{M}\, \Lambda_{t,L}(u)^2 + 8\, \rho_t(x_t)\, e^{-2u^2} + 32M\, e^{-u^2}. \tag{42}$$

Proof. Set $f = G_{s|t,x_t}\psi$. Since R_t is $p_{0|t}(\cdot|x_t)$-invariant, $p_{0|t}(R_t^L f|x_t) = p_{0|t}(f|x_t) = p_{s|t}(\psi|x_t)$. Apply Prop. 9 with $\mu = p_{0|t}(\cdot|x_t)$, $\nu = q_{0|t}(\cdot|x_t)$, $N = M$, and $f = G_{s|t,x_t}\psi$; the resulting estimator $\mu^{M,L}(f)$ is exactly $\bar{p}_{s|t}^{M,L}(\psi|x_t)$, and by (30) its bias coincides with that of $p_{s|t}^{M,L}(\psi|x_t)$. The sub-Gaussian tail assumption gives, exactly as in the proof of Thm. 7,

$$q_{0|t}(B_u^{\mathsf{c}} \mid x_t) \le e^{-u^2}, \qquad \eta_{B_u^{\mathsf{c}}}\, q_{0|t}(B_u^{\mathsf{c}} \mid x_t) \le \rho_t(x_t)\, e^{-2u^2}.$$

The oscillation term is controlled by Prop. 10:

$$[R_t^L G_{s|t,x_t}\psi]_{B_u} \le 2\|\psi\|_\infty \left[2\Phi\left(\frac{\gamma_{t,s}\, \Delta_{t,L}(u)}{2} \right) - 1 \right].$$

Plugging in yields (38)–(39); linearization gives (41)–(42). □

Remark 12 (MSE of the sampled estimator) *The MSE bounds above are stated for* $\bar{p}_{s|t}^{M,L}$*; by* (31) *the sampled estimator* $p_{s|t}^{M,L}$ *carries an extra conditional variance that does not vanish as* $L \to \infty$*. Controlling it needs no new argument. On the extended space* $\mathbb{R}^d \times \mathbb{R}^d$*, set*

$$\bar{\nu}(\mathrm{d}x_0, \mathrm{d}y) = q_{0|t}(\mathrm{d}x_0 \mid x_t) R_{t,x_t}^L(x_0, \mathrm{d}y), \qquad \bar{\mu}(\mathrm{d}x_0, \mathrm{d}y) = p_{0|t}(\mathrm{d}x_0 \mid x_t) R_{t,x_t}^L(x_0, \mathrm{d}y).$$

Then $\mathrm{d}\bar{\mu}/\mathrm{d}\bar{\nu}(x_0, y) = \tilde{w}_{0|t}(x_0)/p_t(x_t)$*, so* $1 + \chi^2(\bar{\mu}\|\bar{\nu}) = \rho_t(x_t)$ *and the SNIS weights built from* $\bar{\nu}$ *coincide with* $w_{0|t}^{1:M}$*. By invariance the* y*-marginal of* $\bar{\mu}$ *is* $p_{0|t}(\cdot|x_t)$*, so* $\bar{\mu}(\bar{f}) = p_{s|t}(\psi|x_t)$ *for* $\bar{f}(x_0, y) = G_{s|t,x_t}\psi(y)$*. Thus* $p_{s|t}^{M,L}(\psi|x_t)$ *is the SNIS approximation of* $\bar{\mu}$ *applied to* $\bar{f}$*, and Lemma 5 with* $B = \mathbb{R}^d \times B_u^y$*, combined with* (27)*, gives the MSE bound of Thm. 7 with* $\kappa_t(x_t)$ *replaced by the sub-Gaussian rate of the* y*-marginal of* $\bar{\nu}$*, i.e. of the* L*-step pushforwards of* $q_{0|t}(\cdot|x_t)$ *and of the normalized square-weight measure through* R_{t,x_t}^L*. Rejuvenation therefore improves the bias by the mixing factor* r_t^L *while leaving the MSE of the same order as without it; averaging* J *conditionally independent moves per particle divides the extra term* (31) *by* J*.*

Corollary 13 (Convenient choice of radius) *Under the assumptions of Thm. 11, taking* $u_M = \sqrt{3 \log M}$ *gives*

$$\|\psi\|_\infty^{-1} \left| \mathbb{E}\left[p_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t) \right] \right| \le \frac{12\, \rho_t(x_t)}{M}\, \Lambda_{t,L}(u_M) + \frac{2\sqrt{\rho_t(x_t)}}{M^3} + \frac{4}{M^2},$$

$$\|\psi\|_\infty^{-2}\, \mathbb{E}\left[\left(\bar{p}_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t) \right)^2 \right] \le \frac{8\, \rho_t(x_t)}{M}\, \Lambda_{t,L}(u_M)^2 + \frac{8\, \rho_t(x_t)}{M^6} + \frac{32}{M^2}.$$

Specialization to MAExL. We now specialize to the case where R_t is the MAExL Markov kernel of Durmus & Moulines (2015) targeting $\pi_t = p_{0|t}(\cdot|x_t)$. Recall $A_t = \alpha_t^2/\sigma_t^2$ from (23) and the auxiliary quantities

$$\bar{m}_t(x_t) = \frac{x_t}{\alpha_t}, \qquad Q_t = A_t\, \mathrm{I}_d.$$

Corollary 14 (Per-step error with MAExL-rejuvenated SNIS) *Fix* $0 < s < t \le T$ *and* $x_t \in \mathbb{R}^d$*. Assume that, after centering at* $\bar{m}_t(x_t)$*, the MAExL kernel* R_t *satisfies the EI-MALA assumptions M1–M2 of Durmus & Moulines (2015). Then by (Durmus & Moulines, 2015, Proposition 3, Theorem 1,*

Theorem 2), there exist constants $C_t < \infty$, $r_t \in (0,1)$, $\eta_t > 0$, $\varepsilon_t > 0$, and a bounded distance $d_t = d_{\eta_t,\varepsilon_t}$ such that for all $L \geq 0$ and $x, y \in \mathbb{R}^d$,

$$W_{d_t}\left(R_t^L(x,\cdot),\, R_t^L(y,\cdot)\right) \leq C_t\, r_t^L\, \{V_t(x) + V_t(y)\}, \qquad V_t(x) = 1 \vee \|x - \bar{m}_t(x_t)\|_{Q_t}.$$

Assume the sub-Gaussian tail assumption holds at (t, x_t) with center $m_t(x_t)$ and rate $\kappa_t(x_t)$. For $u > 0$, set $B_u = \mathcal{B}(m_t(x_t),\, \kappa_t(x_t)(\sqrt{d} + u))$ and $V_{t,u} = \sup_{x \in B_u} V_t(x)$. Define the normalized d_t-Lipschitz constant of $G_{s|t,x_t}\psi$,

$$\mathfrak{L}_{t,s} = \sup_{x \neq y} \frac{\left|G_{s|t,x_t}\psi(x) - G_{s|t,x_t}\psi(y)\right|}{\|\psi\|_\infty\, d_t(x,y)};$$

using (26) *together with the lower comparison of $d_{\eta,\varepsilon}$ with the Q_t-norm (Durmus & Moulines, 2015, Lemma 8), one may take*

$$\mathfrak{L}_{t,s} \leq 2 \vee \left(\sqrt{\tfrac{2}{\pi}}\, \gamma_{t,s}\, \frac{\varepsilon_t}{\sqrt{A_t}}\right). \tag{43}$$

Then, for any bounded test function ψ, with the bias referring to $p_{s|t}^{M,L}$ and the MSE to $\bar{p}_{s|t}^{M,L}$,

$$\|\psi\|_\infty^{-1} \left|\mathbb{E}\left[p_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \leq \frac{12\,\rho_t(x_t)}{M}\, \mathfrak{L}_{t,s}\, C_t\, r_t^L\, V_{t,u} + 2\sqrt{\rho_t(x_t)}\, e^{-u^2} + 4M\, e^{-u^2}, \tag{44}$$

$$\|\psi\|_\infty^{-2}\, \mathbb{E}\left[\left(\bar{p}_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right)^2\right] \leq \frac{8\,\rho_t(x_t)}{M}\, \mathfrak{L}_{t,s}^2\, C_t^2\, r_t^{2L}\, V_{t,u}^2 + 8\,\rho_t(x_t)\, e^{-2u^2} + 32M\, e^{-u^2}. \tag{45}$$

Furthermore, $V_{t,u} \leq 1 \vee \sqrt{A_t}\left(\|m_t(x_t) - \bar{m}_t(x_t)\| + \kappa_t(x_t)(\sqrt{d} + u)\right)$.

Proof. Set $f = G_{s|t,x_t}\psi$. Since R_t is $p_{0|t}(\cdot|x_t)$-invariant, the MCMC-integrated estimator $\bar{p}_{s|t}^{M,L}$ coincides with the SNIS estimator applied to $R_t^L f$. Applying Prop. 9 with $\mu = p_{0|t}(\cdot|x_t)$, $\nu = q_{0|t}(\cdot|x_t)$, $N = M$, and $B = B_u$, we get

$$\left|\mathbb{E}\left[\bar{p}_{s|t}^{M,L}(\psi \mid x_t) - p_{s|t}(\psi \mid x_t)\right]\right| \leq \frac{6\,\rho_t(x_t)}{M}\, [R_t^L f]_{B_u} + 2\|\psi\|_\infty \sqrt{\eta_{B_u^{\mathsf{c}}}\, q_{0|t}(B_u^{\mathsf{c}} \mid x_t)} + 4\|\psi\|_\infty\, M\, q_{0|t}(B_u^{\mathsf{c}} \mid x_t),$$

and analogously for the MSE; by (30) the left-hand side is unchanged if $\bar{p}_{s|t}^{M,L}$ is replaced by $p_{s|t}^{M,L}$.

For the oscillation term, note that f is $\|\psi\|_\infty\, \mathfrak{L}_{t,s}$-Lipschitz with respect to d_t, so the quantitative Wasserstein control of MAExL gives

$$[R_t^L f]_{B_u} \leq \|\psi\|_\infty\, \mathfrak{L}_{t,s} \sup_{x,y \in B_u} W_{d_t}\left(R_t^L(x,\cdot),\, R_t^L(y,\cdot)\right) \leq 2\|\psi\|_\infty\, \mathfrak{L}_{t,s}\, C_t\, r_t^L\, V_{t,u}.$$

The sub-Gaussian tail assumption gives $q_{0|t}(B_u^{\mathsf{c}}|x_t) \leq e^{-u^2}$ and $\eta_{B_u^{\mathsf{c}}}\, q_{0|t}(B_u^{\mathsf{c}}|x_t) \leq \rho_t(x_t)\, e^{-2u^2}$, yielding (44)–(45).

The bound on $V_{t,u}$ follows from the triangle inequality and the definition of the Q_t-norm. □

Interpretation. Compared with the non-rejuvenated SNIS bound of Thm. 7, the leading oscillation term is multiplied by the MAExL mixing factor $C_t\, r_t^L\, V_{t,u}$. Thus the bridge sensitivity $\gamma_{t,s}$ is unaltered by the MCMC step; rather, MAExL contracts the effective x_0-diameter seen by the bridge. The constants C_t, r_t and the adapted metric $d_t = d_{\eta_t,\varepsilon_t}$ are those of the EI-MALA Wasserstein analysis of Durmus & Moulines (2015). The analysis never requires the rejuvenation output to be an exact draw from $p_{0|t}(\cdot|x_t)$: only the invariance identity $p_{0|t}(R_{t,x_t}^L G_{s|t,x_t}\psi|x_t) = p_{s|t}(\psi|x_t)$ is used, and the quality of the finite-L chain, together with its dependence on the initialization supplied by the proposal, enters solely through $\Delta_{t,L}(B)$, which is finite and in general nonzero for every finite L.

A.1.3 FROM PER-STEP BOUNDS TO A PATH-ERROR BOUND

We now propagate the per-step bias and MSE bounds along the diffusion schedule.

Proposition 15 (Telescoping) *Let $t_K > t_{K-1} > \cdots > t_0$, and let P_k be the exact Markov kernel from time t_k to t_{k-1}. Let $\widehat{P}_1, \ldots, \widehat{P}_K$ be independent random Markov kernels approximating $P_1, \ldots, P_K$. Given a probability measure η_K on $\mathbb{R}^d$, define the exact marginals*

$$\eta_{k-1} = \eta_k P_k \quad (k = K, \ldots, 1),$$

so that for any bounded ψ, $\eta_0(\psi) = \eta_K P_K \cdots P_1 \psi$. Define the chained approximation

$$\widehat{\eta}_0(\psi) = \eta_K \widehat{P}_K \widehat{P}_{K-1} \cdots \widehat{P}_1 \psi.$$

Suppose that for every bounded measurable f, every $x \in \mathbb{R}^d$, and every k, there exist deterministic functions b_k and v_k such that

$$\left|\mathbb{E}\left[\widehat{P}_k f(x) - P_k f(x)\right]\right| \le b_k(x, \|f\|_\infty), \qquad \mathbb{E}\left[\left(\widehat{P}_k f(x) - P_k f(x)\right)^2\right] \le v_k(x, \|f\|_\infty)^2.$$

Then, for every bounded measurable ψ,

$$|\mathbb{E}\left[\widehat{\eta}_0(\psi) - \eta_0(\psi)\right]| \le \sum_{k=1}^{K} \eta_k \left[b_k(\cdot, \|\psi\|_\infty)\right], \tag{46}$$

$$\mathbb{E}\left[\left(\widehat{\eta}_0(\psi) - \eta_0(\psi)\right)^2\right] \le \left(\sum_{k=1}^{K} \sqrt{\eta_k \left[v_k(\cdot, \|\psi\|_\infty)^2\right]}\right)^2. \tag{47}$$

Proof. For $k = 1, \ldots, K$, define the random backward test functions $\widehat{\psi}_0 = \psi$ and $\widehat{\psi}_{k-1} = \widehat{P}_{k-1} \cdots \widehat{P}_1 \psi$. Each $\widehat{P}_j$ is a Markov kernel, so $\left\|\widehat{\psi}_{k-1}\right\|_\infty \le \|\psi\|_\infty$. The telescoping identity

$$\widehat{P}_K \cdots \widehat{P}_1 - P_K \cdots P_1 = \sum_{k=1}^{K} P_K \cdots P_{k+1} \left(\widehat{P}_k - P_k\right) \widehat{P}_{k-1} \cdots \widehat{P}_1$$

gives

$$\widehat{\eta}_0(\psi) - \eta_0(\psi) = \sum_{k=1}^{K} \eta_k \left[\left(\widehat{P}_k - P_k\right) \widehat{\psi}_{k-1}\right].$$

Bias. Conditioning on $\widehat{P}_1, \ldots, \widehat{P}_{k-1}$ fixes $\widehat{\psi}_{k-1}$, hence

$$\left|\mathbb{E}\left[\eta_k \left[\left(\widehat{P}_k - P_k\right) \widehat{\psi}_{k-1}\right]\right]\right| \le \eta_k \left[b_k(\cdot, \|\psi\|_\infty)\right].$$

Summing over k gives (46).

MSE. By Minkowski's inequality and Jensen's inequality, conditioning on $\widehat{P}_1, \ldots, \widehat{P}_{k-1}$,

$$\begin{aligned}\left\{\mathbb{E}\left[\left(\widehat{\eta}_0(\psi) - \eta_0(\psi)\right)^2\right]\right\}^{1/2} &\le \sum_{k=1}^{K} \left\{\mathbb{E}\left[\left(\eta_k \left[\left(\widehat{P}_k - P_k\right)\widehat{\psi}_{k-1}\right]\right)^2\right]\right\}^{1/2} \\ &\le \sum_{k=1}^{K} \sqrt{\eta_k \left[v_k(\cdot, \|\psi\|_\infty)^2\right]},\end{aligned}$$

which gives (47). □

Specializing to the chained SNIS estimators yields the path-error bounds.

Corollary 16 (Path error without MCMC) *Let $T = t_K > t_{K-1} > \cdots > t_0 = \tau > 0$, and for $k = 1, \dots, K$ set $s_k = t_{k-1}$ and $\gamma_k = \gamma_{t_k, s_k}$. Let $P_k(x, \mathrm{d}x') = p_{s_k|t_k}(\mathrm{d}x'|x)$ be the exact denoising kernel and let Q_k^M be its SNIS approximation. Assume $Q_1^M, \dots, Q_K^M$ are independent and the initial law is exact, $q_{t_K} = p_{t_K}$, so that $\eta_k = p_{t_k}$. Define the chained approximation*

$$q_\tau^M(\psi) = p_{t_K} Q_K^M \cdots Q_1^M \psi,$$

and write $\rho_k(x) = 1 + \chi^2(p_{0|t_k}(\cdot|x) \,\|\, q_{0|t_k}(\cdot|x))$ and $\kappa_k(x) = \kappa_{t_k}(x)$. Assume that for each k, the sub-Gaussian tail assumption holds at (t_k, x) for every x in the support of p_{t_k}, with center $m_k(x)$ and rate $\kappa_k(x)$. With $u_M = \sqrt{3 \log M}$ and

$$\Theta_k(x) = \min\left\{1,\ \sqrt{\tfrac{2}{\pi}}\, \gamma_k\, \kappa_k(x)\, \left(\sqrt{d} + u_M\right)\right\},$$

set

$$B_k(x) = \frac{12\, \rho_k(x)}{M}\, \Theta_k(x) + \frac{2\sqrt{\rho_k(x)}}{M^3} + \frac{4}{M^2},$$
$$V_k(x) = \frac{8\, \rho_k(x)}{M}\, \Theta_k(x)^2 + \frac{8\, \rho_k(x)}{M^6} + \frac{32}{M^2}.$$

Then, for every bounded measurable ψ,

$$\left|\mathbb{E}\left[q_\tau^M(\psi) - p_\tau(\psi)\right]\right| \le \|\psi\|_\infty \sum_{k=1}^K \eta_k(B_k), \tag{48}$$

$$\mathbb{E}\left[\left(q_\tau^M(\psi) - p_\tau(\psi)\right)^2\right] \le \|\psi\|_\infty^2 \left(\sum_{k=1}^K \sqrt{\eta_k(V_k)}\right)^2. \tag{49}$$

Proof. Apply Prop. 15 to $\widehat{P}_k = Q_k^M$, with the per-step bias and MSE bounds of Thm. 7 at $u = u_M$; these are legitimate since $s_k \ge \tau > 0$ for every k. □

Corollary 17 (Path error with MCMC rejuvenation) *In the setting of Cor. 16, let $Q_k^{M,L}$ be the MAExL-rejuvenated SNIS approximation of P_k with L MCMC steps, and assume $Q_1^{M,L}, \dots, Q_K^{M,L}$ are independent, the independence covering their MCMC randomness as well as their proposal draws. Assume the MAExL constants $C_k(x) < \infty$, $r_k(x) \in (0,1)$, $\varepsilon_{t_k} > 0$ of Cor. 14 are well-defined at every (t_k, x), and set*

$$\mathfrak{L}_k = 2 \vee \left(\sqrt{\tfrac{2}{\pi}}\, \gamma_k\, \frac{\varepsilon_{t_k}}{\sqrt{A_{t_k}}}\right),$$
$$\mathfrak{V}_k(x) = 1 \vee \sqrt{A_{t_k}}\left(\left\|m_k(x) - \tfrac{x}{\alpha_{t_k}}\right\| + \kappa_k(x)\left(\sqrt{d} + u_M\right)\right),$$

as in (43) *and the last display of Cor. 14. Assume moreover that the sub-Gaussian tail assumption holds at (t_k, x) for the L-step pushforwards of $q_{0|t_k}(\cdot|x)$ and of the normalized square-weight measure through $R_{t_k,x}^L$, with common center $\tilde{m}_{k,L}(x)$ and rate $\tilde{\kappa}_{k,L}(x)$, and write*

$$\tilde{\Theta}_{k,L}(x) = \min\left\{1,\ \sqrt{\tfrac{2}{\pi}}\, \gamma_k\, \tilde{\kappa}_{k,L}(x)\left(\sqrt{d} + u_M\right)\right\}.$$

Define the chained approximation

$$q_\tau^{M,L}(\psi) = p_{t_K} Q_K^{M,L} \cdots Q_1^{M,L} \psi,$$

and set

$$B_k(x) = \frac{12\, \rho_k(x)}{M}\, \mathfrak{L}_k\, C_k(x)\, r_k(x)^L\, \mathfrak{V}_k(x) + \frac{2\sqrt{\rho_k(x)}}{M^3} + \frac{4}{M^2},$$
$$V_k(x) = \frac{8\, \rho_k(x)}{M}\, \tilde{\Theta}_{k,L}(x)^2 + \frac{8\, \rho_k(x)}{M^6} + \frac{32}{M^2}.$$

Then, for every bounded measurable ψ,

$$\left|\mathbb{E}\left[q_\tau^{M,L}(\psi) - p_\tau(\psi)\right]\right| \le \|\psi\|_\infty \sum_{k=1}^{K} \eta_k(B_k), \tag{50}$$

$$\mathbb{E}\left[\left(q_\tau^{M,L}(\psi) - p_\tau(\psi)\right)^2\right] \le \|\psi\|_\infty^2 \left(\sum_{k=1}^{K} \sqrt{\eta_k(V_k)}\right)^2. \tag{51}$$

The bias bound therefore contracts geometrically in L, while the MSE bound is of the same order as in Cor. 16; see Remark 12.

Proof. Each $Q_k^{M,L}$ is a random Markov kernel in x_{t_k}: the bridge remains integrated in

$$Q_k^{M,L}(x, \mathrm{d}x') = \sum_{m=1}^{M} w_{0|t_k}^m \, p_{s_k|t_k,0}(\mathrm{d}x' \mid x, y_{0|t_k}^m),$$

the MCMC draws $y_{0|t_k}^m$ being part of its internal randomness. Prop. 15 therefore applies with $\widehat{P}_k = Q_k^{M,L}$, the assumed independence across k covering both sources of randomness.

It remains to supply b_k and v_k. Fix k and a bounded measurable f, and apply the per-step results at (t_k, x) with $s = s_k$, $\psi = f$ and $u = u_M$, which is legitimate since $s_k \ge \tau > 0$. For the bias, (44) together with (43) and $V_{t_k,u_M} \le \mathfrak{V}_k(x)$ gives

$$\left|\mathbb{E}\left[Q_k^{M,L} f(x) - P_k f(x)\right]\right| \le \|f\|_\infty B_k(x),$$

this being a bound on the sampled estimator by (30). For the MSE, Remark 12 gives

$$\mathbb{E}\left[\left(Q_k^{M,L} f(x) - P_k f(x)\right)^2\right] \le \|f\|_\infty^2 V_k(x),$$

with $\tilde{\kappa}_{k,L}(x)$ the sub-Gaussian rate of the L-step pushforwards assumed above. Both bounds are deterministic and of the form required by Prop. 15, and since $\left\|\widehat{\psi}_{k-1}\right\|_\infty \le \|\psi\|_\infty$ for every k, (46)–(47) yield the claim. □

Remark 18 (Early stopping and terminal initialization) *The grid terminates at $t_0 = \tau > 0$ ($\tau = 10^{-4}$ in the experiments), so by* (23) *every step has $\gamma_k \le \sqrt{A_{s_k}} \le \sqrt{A_\tau} = \alpha_\tau/\sigma_\tau < \infty$ and the constants above are finite, the final-step constant scaling as σ_τ^{-1}. The truncated form* (27) *keeps the per-step bound $O(\rho_k(x)/M)$ uniformly in τ. The target controlled by these bounds is thus p_τ, not p_0. Likewise, Cors. 16 and 17 assume $q_{t_K} = p_{t_K}$, whereas the sampler starts from $\nu_T = \mathcal{N}(0, \sigma_T^2 \mathrm{I}_d)$: decomposing along $\nu_T P_K \cdots P_1 \psi$ adds $2\|\psi\|_\infty \, \mathrm{TV}(\nu_T, p_{t_K})$ to the bias bound and to the square root of the MSE bound, with η_k then the marginals of the exact backward chain started at ν_T. Under the VE schedule $\alpha_T = 1$, so $p_{t_K} = p_0 * \mathcal{N}(0, \sigma_T^2 \mathrm{I}_d)$ and joint convexity of the Kullback–Leibler divergence together with Pinsker's inequality give*

$$\mathrm{TV}(\nu_T, p_{t_K}) \le \frac{\sqrt{\mathbb{E}_{p_0}\|X_0\|^2}}{2\sigma_T}.$$

Neither term depends on M or L, so consistency in M holds for the (τ, T)-truncated target only. Finally, bounding the bias uniformly over $\|\psi\|_\infty \le 1$ in (48) *controls* TV *between the law of the sampler output and p_τ. The MSE bounds concern instead the across-run variability of expectation estimates.*

A.2 Multiple Importance Sampling

This appendix details the NF^{2}M-MIS sampler introduced in Sec. 3.3, which builds on the *multiple importance sampling* (MIS) literature (Veach & Guibas, 1995; Owen & Zhou, 2000; He & Owen, 2014; Bugallo et al., 2017; Elvira et al., 2019). Recall that NF^{2}M-IS shares candidates only at initialization; the KL identity (17) is what lets us extend this sharing further along the chain, since it shows that posteriors conditioned on nearby noisy states are themselves close. We specify the mixture proposal, component selection, and candidate sharing; derive the self-normalized importance-sampling correction for the reverse kernel; and give the full algorithm together with its computational cost.

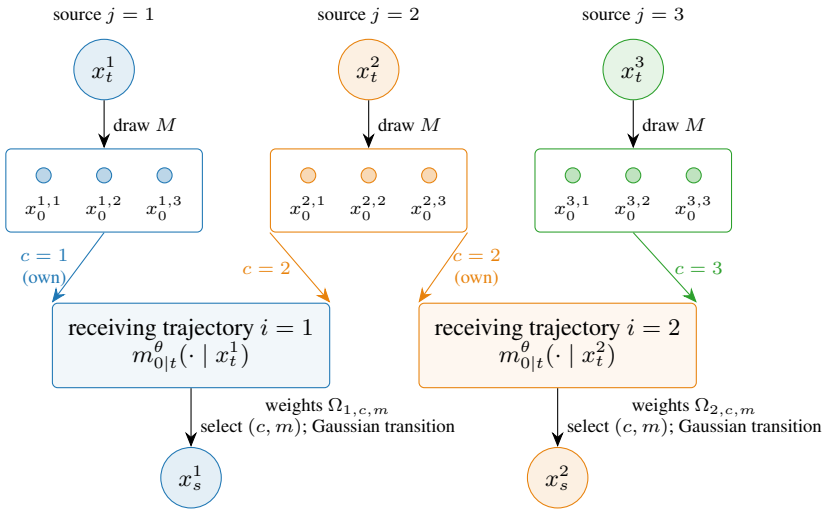


Figure 3: **Candidate sharing and indexing in NF^{2}M-MIS.** The example uses $N = 3$ sources, $M = 3$ candidates per source, and $C = 2$ components per mixture; two receiving trajectories are shown. Candidate $x_0^{j,m}$ is indexed by its source j and position m within that source's batch. For receiving trajectory i, a component $c \in \mathcal{C}(i)$ is itself a source index: $c = i$ always appears (the trajectory's "own" component), and the remaining slots are drawn by posterior proximity. The middle batch, drawn once from source $j = 2$, is reused by both trajectories: it serves as component $c = 2$ of trajectory 1's mixture and as the own component $c = 2$ of trajectory 2's mixture, receiving weight $\Omega_{1,2,m}$ and $\Omega_{2,2,m}$ respectively — the same M flow draws inform two different reverse steps. Each trajectory samples (c, m) from its own categorical weights and applies the endpoint-conditioned Gaussian transition. When a source is selected more than once within the same mixture, the repeat instead draws a fresh, independent batch, indexed by the occurrence counter $r(i, c)$ introduced below.

Mixture proposal. At noise level t, let $\{x_t^i\}_{i=1}^N$ denote the current trajectories. For each trajectory i, we build a mixture proposal by combining C selected learned posteriors,

$$m_{0|t}^{\theta}(x_0 \mid x_t^i) = \sum_{c \in \mathcal{C}(i)} \omega_{i,c}\, q_{0|t}^{\theta}(x_0 \mid x_t^c), \qquad \sum_{c \in \mathcal{C}(i)} \omega_{i,c} = 1, \tag{52}$$

while keeping the target unchanged at $p_{0|t}(\cdot|x_t^i)$. The component set $\mathcal{C}(i)$ is a multiset of trajectory indices that always contains i itself, so trajectory i's own posterior is always a summand; the other $C - 1$ slots are drawn from elsewhere in the ensemble. Fig. 3 works through a small example of this construction and of the candidate sharing described below.

Choosing components by posterior proximity. By (17), for a Gaussian target $\mathcal{N}(\mu, \gamma^2 \mathrm{I}_d)$ the denoising posteriors satisfy

$$\mathrm{KL}\left(p_{0|t}(\cdot \mid x) \,\|\, p_{0|t}(\cdot \mid y)\right) = \beta_t \|x - y\|^2, \qquad \beta_t = \frac{\alpha_t^2 \gamma^2}{2\sigma_t^2(\alpha_t^2 \gamma^2 + \sigma_t^2)},$$

so state-space distance provides a schedule-dependent measure of posterior similarity in this Gaussian case; for molecular targets we use it as a heuristic with $\gamma^2 = 1$ in normalized coordinates. This is also what makes sharing most valuable at high noise levels: as t grows, $\beta_t \to 0$ and the bound above shows the posteriors across *all* trajectories converge toward the common Boltzmann target, so nearly any component is an informative proposal for any trajectory. Given β_t, define for every pair (i, j) the proximity score

$$\lambda_j(i) = \exp\left\{-\beta_t \left\| x_t^i - x_t^j \right\|^2\right\}. \tag{53}$$

The component set $\mathcal{C}(i)$ always retains the slot i itself, and the remaining $C - 1$ slots are drawn independently, with replacement, from the multinomial over $\{1, \ldots, N\}$ with weights $\lambda_1(i), \ldots, \lambda_N(i)$.

The mixture weights are then set proportionally to these same scores, restricted to the selected slots,

$$\omega_{i,c} = \frac{\lambda_c(i)}{\sum_{l \in \mathcal{C}(i)} \lambda_l(i)}, \tag{54}$$

so that conditioning states closer to x_t^i contribute more strongly to its own proposal.

Rao-Blackwellized mixture sampling. We now explain how expectations under $m^\theta_{0|t}(\cdot|x_t^i)$ are computed. Just as in NF^{2}M-IS, we draw one *base batch* of M i.i.d. candidates $\xi_{j,1,1}, \dots, \xi_{j,1,M} \sim q^\theta_{0|t}(\cdot|x_t^j)$ for every trajectory $j = 1, \dots, N$ (this is the batches drawn under each source j in Fig. 3). To assemble trajectory i's CM candidates, each slot $c \in \mathcal{C}(i)$ draws on the batch of its conditioning trajectory: the first time a trajectory is used as a component it reuses the shared base batch, as source $j = 2$ does for both receiving trajectories in the figure. Because $\mathcal{C}(i)$ is a multiset, however, the same trajectory can occupy more than one slot of the *same* mixture; reusing one batch twice within a single mixture would break the independence the estimator below relies on, so each repeated occurrence instead draws a fresh, independent batch. Writing $r(i, c)$ for the occurrence number of slot c within $\mathcal{C}(i)$ (its first, second, … appearance), we denote the resulting candidates

$$x_0^{i,c,m} = \xi_{c,\, r(i,c),\, m}, \qquad \xi_{j,r,m} \sim q^\theta_{0|t}(\cdot \mid x_t^j), \quad m = 1, \dots, M,$$

where batches are shared across every trajectory whose component set calls for the same pair (j, r), and drawn independently otherwise. This lets us reuse our expensive flow samples as much as possible while still having independent candidates within each trajectory's own mixture. Expectations under $m^\theta_{0|t}(\cdot|x_t^i)$ can then be computed in a Rao-Blackwellized fashion, by conditioning on the mixture component rather than sampling it:

$$\mathbb{E}_{m^\theta_{0|t}(\cdot|x_t^i)}\left[\phi(X_0^i)\right] = \sum_{c \in \mathcal{C}(i)} \omega_{i,c}\, \mathbb{E}_{q^\theta_{0|t}(\cdot|x_t^c)}\left[\phi(X_0^{i,c})\right] \approx \frac{1}{M} \sum_{c \in \mathcal{C}(i)} \sum_{m=1}^{M} \omega_{i,c}\, \phi\big(x_0^{i,c,m}\big). \tag{55}$$

Self-normalized importance sampling for the reverse kernel. We now return to the denoising kernel approximation. Writing the exact transition as an expectation under the mixture proposal,

$$\begin{aligned}
p_{s|t}(x_s^i \mid x_t^i) &= \int p_{s|t,0}(x_s^i \mid x_t^i, x_0^i)\, p_{0|t}(x_0^i \mid x_t^i)\, \mathrm{d}x_0^i \\
&= \int p_{s|t,0}(x_s^i \mid x_t^i, x_0^i)\, \frac{p_{0|t}(x_0^i \mid x_t^i)}{m^\theta_{0|t}(x_0^i \mid x_t^i)}\, m^\theta_{0|t}(x_0^i \mid x_t^i)\, \mathrm{d}x_0^i \\
&= \mathbb{E}_{m^\theta_{0|t}(\cdot|x_t^i)}\left[\frac{p_{0|t}(X_0^i \mid x_t^i)}{m^\theta_{0|t}(X_0^i \mid x_t^i)}\, p_{s|t,0}(x_s^i \mid x_t^i, X_0^i)\right] \\
&\approx \sum_{c \in \mathcal{C}(i)} \sum_{m=1}^{M} \omega_{i,c}\, \Lambda_{i,c,m}\, p_{s|t,0}(x_s^i \mid x_t^i, x_0^{i,c,m}),
\end{aligned}$$

where the last line applies the Rao-Blackwellized estimator (55), replacing the intractable ratio $p_{0|t}/m^\theta_{0|t}$ by a self-normalized importance weight $\Lambda_{i,c,m}$. Concretely, the unnormalized weights are

$$\tilde{\Lambda}_{i,c,m} = \frac{p_{0|t}(x_0^{i,c,m} \mid x_t^i)}{m^\theta_{0|t}(x_0^{i,c,m} \mid x_t^i)} \propto \frac{p_{t|0}(x_t^i \mid x_0^{i,c,m})\, \exp\{-\mathcal{E}(x_0^{i,c,m})\}}{\sum_{c' \in \mathcal{C}(i)} \omega_{i,c'}\, q^\theta_{0|t}\left(x_0^{i,c,m} \mid x_t^{c'}\right)}.$$

Because these weights only enter the final mixture in combination with $\omega_{i,c}$, we normalize the two together in one step,

$$\Omega_{i,c,m} = \frac{\omega_{i,c}\, \tilde{\Lambda}_{i,c,m}}{\sum_{c' \in \mathcal{C}(i)} \sum_{m'=1}^{M} \omega_{i,c'}\, \tilde{\Lambda}_{i,c',m'}}. \tag{56}$$

This gives a new mixture approximation of $p_{s|t}$,

$$\hat{p}^{\mathcal{C},M,\theta}_{s|t}(x_s^i \mid x_t^i) = \sum_{c \in \mathcal{C}(i)} \sum_{m=1}^{M} \Omega_{i,c,m}\, p_{s|t,0}(x_s^i \mid x_t^i, x_0^{i,c,m}), \tag{57}$$

composed at the cost of only M flow draws per component, yet at least as expressive than the M-candidate proposal of NF^{2}M-IS.

Algorithm 2 NF^{2}M-MIS sampling.

Input: Grid $0 = t_0 < \cdots < t_K = T$; posteriors $q^\theta_{0|t_k}$; target $p(x_0) \propto \exp\{-\mathcal{E}(x_0)\}$; trajectories N; batch size M; component slots C.

Output: Ensemble $\{x^i_{t_0}\}_{i=1}^N$ approximately distributed as p_0.

Sample shared $x_{t_K} \sim \mathcal{N}(0, \sigma_T^2 I)$ and draw NM candidates from $q^\theta_{0|t_K}(\cdot \mid x_{t_K})$

Form joint-pool SNIS weights as in (11) with NM candidates; select N candidates with replacement according to these weights and propagate them through their DDIM kernels to obtain $x^i_{t_{K-1}}$

for $k = K-1, \ldots, 1$ **do**

  For every trajectory i, compute the proximity scores $\lambda_j(i) = \exp\{-\beta_{t_k}\|x^i_{t_k} - x^j_{t_k}\|^2\}$, draw the component multiset $\mathcal{C}(i)$ by proximity-weighted sampling with slot i fixed, and set $\omega_{i,\cdot}$ via (54)

  Let $r(i,c)$ count the occurrences of c among the slots of $\mathcal{C}(i)$ drawn so far. For each distinct pair (j,r) arising across all trajectories, draw one independent batch of M candidates $\xi_{j,r,m} \sim q^\theta_{0|t_k}(\cdot \mid x^j_{t_k})$

  Assemble trajectory i's CM candidates as $x_0^{i,c,m} = \xi_{c,\, r(i,c),\, m}$, evaluate the required cross-likelihoods $q^\theta_{0|t_k}(x_0^{i,c,m} \mid x^{c'}_{t_k})$ for $c, c' \in \mathcal{C}(i)$, and form the selection weights

$$\Omega_{i,c,m} \propto \omega_{i,c}\, \tilde{\Lambda}_{i,c,m}, \qquad \tilde{\Lambda}_{i,c,m} = \frac{p_{t_k|0}(x^i_{t_k} \mid x_0^{i,c,m})\, \exp\{-\mathcal{E}(x_0^{i,c,m})\}}{\sum_{c' \in \mathcal{C}(i)} \omega_{i,c'}\, q^\theta_{0|t_k}(x_0^{i,c,m} \mid x^{c'}_{t_k})}$$

  For every trajectory i, sample $(c^\star, m^\star) \sim \mathrm{Categorical}(\Omega_i)$ and set

$$x^i_{t_{k-1}} \sim p_{t_{k-1}|t_k,0}(\cdot \mid x^i_{t_k},\, x_0^{i,c^\star,m^\star})$$

end

return $\{x^i_{t_0}\}_{i=1}^N$

The computational catch and the autoregressive blessing. This added expressiveness is not free: computing $\Omega_{i,c,m}$ requires the cross-likelihoods

$$q^\theta_{0|t}\left(x_0^{i,c,m} \mid x_t^{c'}\right), \qquad \text{for all } c, c' \in \mathcal{C}(i).$$

The C "diagonal" terms $q^\theta_{0|t}(x_0^{i,\bar{c},m}|x_t^{\bar{c}})$ are returned for free with each flow sample and, like the samples themselves, are reused whenever a slot shares its source (Fig. 3). Only the remaining $C(C-1)$ cross terms per trajectory need a new evaluation, giving at most $NMC(C-1)$ additional evaluations per level (zero when $C = 1$, where the mixture collapses to NF^{2}M-IS) and less in practice when component sets overlap across trajectories. This overhead stays comparatively harmless for an autoregressive flow, where sampling is sequential and linear in token count but density evaluation is a single parallel pass: since sampling cost dominates, the batch-sharing scheme of Fig. 3 is what drives wall-clock cost, while the extra cross-likelihoods remain comparatively negligible.

The full sampler is summarized in Alg. 2.

Amortized cost. Let U denote the number of distinct pairs (j,r) drawn across the ensemble at a given level. Candidate generation then costs UM flow samples, with $N \le U \le NC$, against NCM without sharing.

The lower bound $U = N$ holds when no source repeats *within* any single trajectory's $\mathcal{C}(i)$: every external pick then falls at its first occurrence, already drawn by its source for itself, so cross-trajectory sharing is automatic. Such repeats become rarer as the proximity weights $\lambda_j(i)$ spread across more indices, the high-noise regime of (17) ($\beta_t \to 0$). The upper bound $U = NC$ is reached e.g. if every trajectory fills all C slots with itself: no batch is ever shared, and the mixture collapses to a single density, paying $C\times$ the sampling cost of NF^{2}M-IS for none of its added expressiveness.

Empirical ablation We compare the performance of NF^{2}M-MIS against NF^{2}M-IS along two axes: candidate-pool quality and downstream performance. Fig. 4 reports the average effective sample size at each denoising transition, and shows that NF^{2}M-MIS with C components and M candidates per

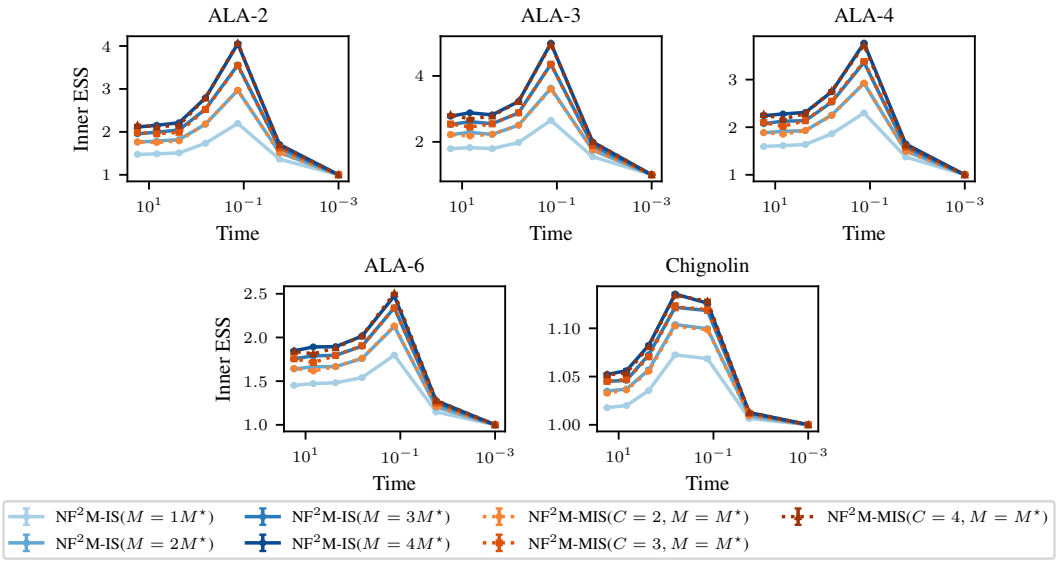


Figure 4: Average effective sample size of $q^{\theta}_{0|t}(\cdot|x_t)$ against $p_{0|t}(\cdot|x_t)$ as a function of t, for NF^{2}M-IS and NF^{2}M-MIS across different numbers of importance candidates or mixture components. $M^\star$ denotes the budget reported in Tab. 18.

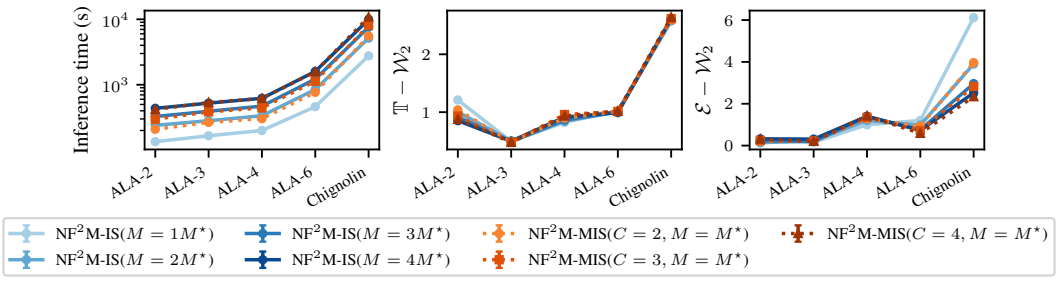


Figure 5: Inference time and distributional distances for all systems, comparing NF^{2}M-IS and NF^{2}M-MIS across different numbers of importance candidates or mixture components. $M^\star$ denotes the budget reported in Tab. 18.

component matches the pool quality of NF^{2}M-IS run with CM independent candidates, confirming that sharing candidates across trajectories does not degrade the proposal. However, both methods exhibit low inner ESS, particularly for chignolin and at the lowest noise levels, indicating substantial importance-weight concentration. This is surprising, as noise levels near $t = 0$ should make the target easiest to approximate. We hypothesize instead that the near-deterministic posterior forces the flow's affine-coupling scale network into a high-variance regime that is difficult to learn. Preconditioning the flow with techniques akin to Karras et al. (2022) often degraded performance rather than fixing this. Fortunately, this is precisely the regime where the MAExL steps (App. A.3) are most accurate, compensating for the weak proposal.

Fig. 5 shows that this parity carries over to downstream distributional distances, with the two methods performing on par at matched budgets. The left panel of Fig. 5 shows that inference time is likewise comparable between the two methods. This can be explained at our current scale, since for molecules this small the autoregressive flow's sampling and log-likelihood costs are already comparable on a modern GPU, leaving little headroom for the sharing scheme in NF^{2}M-MIS to reduce wall-clock time. We expect this gap to widen in NF^{2}M-MIS's favor as system size grows and sampling cost increasingly dominates evaluation cost. Finally, Fig. 5 confirms the expected trend that both methods' metrics improve monotonically as the candidate budget increases.

A.3 Metropolis-Adjusted Exponential Langevin

A recurring subroutine in our pipeline is sampling from a posterior whose likelihood is an isotropic Gaussian in the latent variable. This subsection describes the kernel we use for that subroutine: a Metropolis-adjusted exponential integrator of the corresponding Langevin diffusion, designed to remain stable across the full range of likelihood scales encountered during inference.

Target distribution. We need to sample from the conditional distribution of the clean sample x_0 given a noisy observation x_t produced by the forward process at time t. With schedule (α_t, σ_t), this is

$$p_{0|t}(x_0|x_t) \propto \mathcal{N}\left(x_t; \alpha_t x_0, \sigma_t^2 I\right) p(x_0), \tag{58}$$

where p is the data distribution, accessed through its log-density and score $\nabla \log p$. Throughout this subsection (α_t, σ_t) are fixed scalars.

Associated Langevin SDE. The score of (58) decomposes as

$$\nabla_{x_0} \log p_{0|t}(x_0|x_t) = \nabla \log p(x_0) - Ax_0 + b, \qquad A := \frac{\alpha_t^2}{\sigma_t^2}, \quad b := \frac{\alpha_t}{\sigma_t^2} x_t, \tag{59}$$

and the overdamped Langevin diffusion targeting (58) reads

$$dX_s = \left(-AX_s + b + \nabla \log p(X_s)\right) ds + \sqrt{2} dW_s. \tag{60}$$

The drift has an *affine* part $(-AX_s + b)$ coming from the Gaussian likelihood and a nonlinear part coming from the prior score.

Exponential integration. We integrate (60) by treating the affine part exactly and freezing the prior score over each step. Variation of constants on $Y_s = \exp(A_s)X_s$ gives, on a step of size h,

$$X_{s+h} = e^{-Ah} X_s + \int_0^h e^{-A(h-u)} \left(b + \nabla \log p(X_{s+u})\right) du + \sqrt{2} \int_0^h e^{-A(h-u)} dW_u. \tag{61}$$

Freezing the score at X_s produces the (exponential Euler) one-step kernel

$$X_{s+h} \approx e^{-Ah} X_s + \frac{1 - e^{-Ah}}{A} \left(b + \nabla \log p(X_s)\right) + \sqrt{\tfrac{1-e^{-2Ah}}{A}} \xi, \qquad \xi \sim \mathcal{N}(0, I). \tag{62}$$

Compared to plain Euler–Maruyama, this scheme (i) preserves the Ornstein–Uhlenbeck invariant law of the linear sub-problem at *every* step size, so its noise variance is correct rather than $\mathcal{O}(h)$-biased, and (ii) is unconditionally stable in the affine part: there is no $hA \lesssim 1$ restriction. This is critical when the likelihood is informative (small σ_t), where Euler–Maruyama is forced to take vanishingly small steps.

Metropolis correction. Treating (62) as a Markov kernel proposal and accepting with the standard Metropolis–Hastings probability gives an exact sampler of (58). Because the proposal covariance is state-independent (scalar isotropic), the log-determinant terms cancel and the log Metropolis–Hastings ratio reduces to

$$\log \alpha(x, x') = \log p_{0|t}(x'|x_t) - \log p_{0|t}(x|x_t) + \frac{1}{4\tau}\left(\|x' - \mu(x)\|^2 - \|x - \mu(x')\|^2\right), \qquad (63)$$

where $\mu(x) = e^{-Ah}x + A^{-1}(1 - e^{-Ah})(b + \nabla \log p(x))$ and τ is defined below. The proposed point is accepted with probability $\min(1, e^{\log \alpha})$. We refer to this kernel as *Metropolis-Adjusted Exponential Langevin* (MAExL).

Step-size reparametrization. Rather than tuning h directly, we reparametrize via the noise-to-signal ratio $\tau := (1 - e^{-2Ah})/(2A) \in (0, \tau_{\max})$, with $\tau_{\max} := 1/(2A) = \sigma_t^2/(2\alpha_t^2)$, so that the proposal is expressed entirely through

$$c(\tau) := \sqrt{1 - 2A\tau}, \qquad \varphi(\tau) := \frac{1 - c(\tau)}{A} = \frac{2\tau}{1 + c(\tau)}, \qquad (64)$$

giving $\mu(x) = c(\tau)x + \varphi(\tau)(b + \nabla \log p(x))$ and noise variance 2τ. The right-hand expression for $\varphi(\tau)$ is algebraically equivalent but numerically well-conditioned for small τ, avoiding cancellation in $1 - \sqrt{1 - 2A\tau}$. This reparametrization has three advantages. First, it maps the unbounded, saturating $h \in (0, \infty)$ onto the bounded interval $(0, \tau_{\max})$. Second, 2τ is exactly the proposal variance, and as $\tau \to 0$ the proposal reduces to MALA with step size τ. Third, as $\tau \to \tau_{\max}$, $c(\tau) \to 0$ and the proposal forgets the current state except through the prior score, jumping at the likelihood scale σ_t/α_t. A single bounded scalar thus moves the kernel from local MALA moves to global, likelihood-scale moves.

$\tau_{\max}$ itself still shrinks by several orders of magnitude as $t \to 0$, so τ is not the quantity we carry across noise levels. We instead track the induced proposal noise standard deviation $\omega := \sqrt{2\tau} \in (0, \omega_{\max})$, with $\omega_{\max} := \sqrt{2\tau_{\max}} = \sigma_t/\alpha_t$: this is the intended proposal noise scale *in clean x_0-coordinates*, and it is ω that we adapt per chain and report per system, as a practical parameterization for transferring step sizes across noise levels; the next paragraph clarifies the limits of this interpretation. Because $\omega_{\max}$ depends on t, an adapted ω can legitimately exceed it. Rather than clip ω itself, we cap only the noise scale actually used by the proposal,

$$\epsilon := \min\left(\omega,\ \rho_{\max}\,\omega_{\max}\right), \qquad \tau = \frac{\epsilon^2}{2}, \qquad \rho_{\max} = 0.99, \qquad (65)$$

substituted for τ in (64), leaving ω itself unclamped above the cap so that the previously adapted value is reused unchanged once $\omega_{\max}$ grows again at a later (larger) t.

Why ω transfers across noise levels. The exponential integrator treats the Gaussian-likelihood drift exactly, so the only approximation is freezing the prior score. Concretely, suppose the prior score is constant , $\nabla \log p \equiv g_0$ for some constant g_0 (so $\log p(x) = g_0 \cdot x + \text{const}$). Then $\mu(x) = cx + \varphi(b + g_0)$, and the identities $1 - c^2 = A\epsilon^2$ and $\varphi(1 + c) = \epsilon^2$ give

$$\frac{1}{2\epsilon^2}\left(\|x' - \mu(x)\|^2 - \|x - \mu(x')\|^2\right) = \frac{A}{2}\left(\|x'\|^2 - \|x\|^2\right) - (b + g_0) \cdot (x' - x), \qquad (66)$$

which cancels exactly against $\log p_{0|t}(x') - \log p_{0|t}(x) = g_0 \cdot (x' - x) - \frac{A}{2}(\|x'\|^2 - \|x\|^2) + b \cdot (x' - x)$. Hence $\log \alpha(x, x') \equiv 0$ for *every* $A > 0$, every $\tau \in (0, \tau_{\max})$, and every pair (x, x'): whenever the prior score is constant, every proposal is accepted regardless of the likelihood precision.

Rejection therefore stems from the variation of $\nabla \log p$ over a step, i.e. from the curvature of the prior at the scale ω of the move, rather than from A_t directly. Acceptance is not exactly t-independent, since A_t still enters through c, φ and the visited states, but this makes ω the natural quantity to carry across noise levels.

Connection to score-matching trade-offs. The two ends of the τ-range, global likelihood-scale moves and local MALA moves on the posterior score, mirror the tension between using the likelihood (Tweedie's identity, denoising-score-matching style) and the posterior score (target-score-matching style) as the primary signal; a growing line of work argues that neither extreme is universally best Bortoli et al. (2024); Kahouli et al. (2025); Young et al. (2026); Blessing et al. (2026). Our step-size adaptation lets the sampler choose its own balance, per chain and per noise level.

Table 3: Per-system calibrated step sizes (Robbins–Monro, target acceptance 0.574).

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
MAExL ω (in-loop)	1.02×10^{-2}	7.48×10^{-3}	6.55×10^{-3}	4.68×10^{-3}	3.46×10^{-3}
MALA τ (post-hoc)	5.27×10^{-5}	2.75×10^{-5}	2.14×10^{-5}	1.10×10^{-5}	6.01×10^{-6}

Per-chain step-size adaptation. As above, adaptation runs on $\log\omega$ rather than $\log\tau$: step sizes are adapted independently for each chain in a batch by an asymmetric geometric update, so as to favor the high-acceptance regime:

$$\omega_{m+1} = \max\left(\omega_m \exp\left(\eta_m(\hat{\alpha}_m - \alpha^\star)\right),\, 10^{-30}\right), \tag{67}$$

$$\eta_m = \begin{cases} 0 & \text{if } \epsilon_m = \rho_{\max}\omega_{\max} \text{ and } \hat{\alpha}_m > \alpha^\star \\ \eta_\uparrow & \text{if } \hat{\alpha}_m > \alpha^\star \\ \eta_\downarrow & \text{if } \hat{\alpha}_m \le \alpha^\star, \end{cases} \tag{68}$$

with target acceptance $\alpha^\star = 0.574$ (Roberts & Rosenthal, 1998) and gains $\eta_\uparrow = 0.1, \eta_\downarrow = 0.5$. Here $\hat{\alpha}_m$ is the per-chain Metropolis acceptance probability $\min(1, e^{\log r})$ at iteration m, and $\epsilon_m = \min(\omega_m, \rho_{\max}\omega_{\max})$ is the noise scale actually used by the m-th proposal. The asymmetry ($\eta_\downarrow > \eta_\uparrow$) is deliberate: a stalled chain in the low-acceptance regime is far more harmful for short runs than a slightly conservative one, so we react aggressively to low acceptance and gently to high acceptance. Working in $\log\omega$ keeps ω positive automatically; rather than clip ω against the t-dependent cap $\rho_{\max}\omega_{\max}$, we freeze upward moves while that cap is already active, since acceptance is then insensitive to ω and the frozen value is picked back up once $\omega_{\max}$ grows again at a later (larger) t. A separate, diminishing-gain Robbins–Monro schedule on $\log\omega$ ($\eta_m = \eta_0/(m+t_0)^\kappa$ with $\eta_0 = 0.05$, $t_0 = 10$, $\kappa = 0.75$) is used only offline, to calibrate the fixed per-system presets of Table 3; the geometric schedule above is what runs in the loop. The analysis of App. A.1 applies to this kernel with the step sizes frozen.

Calibrated step sizes. Table 3 reports, per system, the in-loop MAExL step size ω, together with the step size τ used by a separate post-hoc MALA pass that polishes a finished sample set against the clean prior $p(x_0)$ directly, with no likelihood term. Both step sizes were obtained from very long calibration runs (128 chains, a short geometric burn-in followed by several thousand steps of the Robbins–Monro schedule above, at target acceptance $\alpha^\star = 0.574$) and are then held fixed as the shared starting point across all subsequent runs on that system.

Connection with EI-MALA (Durmus & Moulines, 2015). The MAExL kernel is closely related to the exponential-integrator version of MALA analyzed in Durmus & Moulines (2015). In that work, the authors study targets that are Gaussian reference measures perturbed by a nonlinear potential, and construct a Metropolis-adjusted proposal by integrating the Gaussian linear drift exactly while treating the nonlinear part explicitly. This is precisely the structure exploited here: the Gaussian likelihood contribution to $p_{0|t}(\cdot|x_t)$ induces the affine drift $-Ax + b$, which is integrated exactly, while the prior score $\nabla\log p$ is frozen over one step and corrected by a Metropolis–Hastings accept-reject step. In the notation of Durmus & Moulines (2015), our scalar precision AI plays the role of the Gaussian precision matrix Q, and the reparametrization $\tau = (1 - e^{-2Ah})/(2A)$ yields the same proposal covariance as their exponential-integrator MALA. Their Wasserstein convergence analysis is therefore directly aligned with the role of MAExL in our method: it provides quantitative control of the contraction and mixing of the $p_{0|t}$-invariant Markov kernel, which can be combined with the bridge sensitivity factor $\gamma_{t,s}$ to sharpen the propagated SNIS error bounds.

Corollary 19 (Quantitative Wasserstein control of MAExL) *Fix $t \in (0, T]$ and $x_t \in \mathbb{R}^d$, and write*

$$\pi_t(\mathrm{d}x_0) = p_{0|t}(x_0|x_t)\mathrm{d}x_0.$$

Let R_t denote the MAExL Markov kernel targeting π_t. Set

$$A_t = \frac{\alpha_t^2}{\sigma_t^2}, \qquad \bar{m}_t(x_t) = A_t^{-1}b_t = \frac{x_t}{\alpha_t}, \qquad Q_t = A_t I_d.$$

In the centered variable $z = x_0 - \bar{m}_t(x_t)$, assume that the target can be written as

$$\pi_t(\mathrm{d}z) \propto \exp\left(-\frac{1}{2} z^\top Q_t z - \Gamma_t(z)\right) \mathrm{d}z,$$

where $\Gamma_t(z) = -\log p(z + \bar{m}_t(x_t))$, or more generally where the prior contribution is split as in the EI-MALA framework of Durmus & Moulines (2015). Assume that the corresponding MAExL/EI-MALA proposal satisfies assumptions M1–M2 of Durmus & Moulines (2015) for the step size used by R_t. Then there exist constants

$$\eta_t > 0, \qquad \varepsilon_t > 0, \qquad C_t < \infty, \qquad r_t \in (0,1),$$

and a bounded distance $d_t = d_{\eta_t,\varepsilon_t}$, topologically equivalent to the Q_t-norm, such that, for all $L \geq 0$ and all $x_0, y_0 \in \mathbb{R}^d$,

$$W_{d_t}\left(R_t^L(x_0,\cdot), R_t^L(y_0,\cdot)\right) \leq C_t r_t^L \left\{V_t(x_0) + V_t(y_0)\right\}, \tag{69}$$

$$W_{d_t}\left(R_t^L(x_0,\cdot), \pi_t\right) \leq C_t r_t^L V_t(x_0), \tag{70}$$

where

$$V_t(x_0) = 1 \vee \|x_0 - \bar{m}_t(x_t)\|_{Q_t} = 1 \vee \sqrt{A_t}\|x_0 - \bar{m}_t(x_t)\|.$$

Consequently, for any measurable set $B \subset \mathbb{R}^d$, the local Wasserstein diameter of the L-step MAExL kernel satisfies

$$\Delta_{t,L}^{(d_t)}(B) := \sup_{x_0,y_0 \in B} W_{d_t}\left(R_t^L(x_0,\cdot), R_t^L(y_0,\cdot)\right) \leq 2C_t r_t^L \sup_{x_0 \in B} V_t(x_0).$$

In particular, for

$$B_u = \mathcal{B}\left(m_t(x_t), \kappa_t(x_t)(\sqrt{d} + u)\right),$$

one has

$$\Delta_{t,L}^{(d_t)}(B_u) \leq 2C_t r_t^L \left[1 \vee \sqrt{A_t}\left(\|m_t(x_t) - \bar{m}_t(x_t)\| + \kappa_t(x_t)(\sqrt{d} + u)\right)\right].$$

Proof. In the centered variable $z = x_0 - \bar{m}_t(x_t)$, the MAExL proposal coincides with the exponential-integrator MALA proposal applied to a Gaussian reference measure with precision $Q_t = A_t I_d$, perturbed by the prior potential. Under assumptions M1–M2 of Durmus & Moulines (2015), their Theorem 2 gives geometric convergence of the EI-MALA kernel in a Wasserstein distance W_{d_t} associated with a bounded distance $d_t = d_{\eta_t,\varepsilon_t}$. This yields (70). Their Proposition 3 gives the corresponding pairwise contraction estimate, which yields (69). Taking the supremum over $x_0, y_0 \in B$ gives the bound on $\Delta_{t,L}^{(d_t)}(B)$. The final display follows by bounding V_t on B_u. □

Impact of MAExL. Tab. 4 isolates the contribution of the in-loop MAExL steps by rerunning the selected NF^{2}M configurations with 1024, 512 and 0 steps per noise level. The effect is concentrated almost entirely in the energy metric. Disabling MAExL degrades $\mathcal{E}$-$\mathcal{W}_2$ on every system and for both weighting schemes, in most cases by a factor of two or more, and on ALA-4 it is what separates a usable estimator from an unreliable one: without MAExL the energy discrepancy grows by an order of magnitude and its across-seed standard deviation exceeds its mean, i.e. individual seeds fail outright, while the same configuration with MAExL steps is both accurate and tightly concentrated. MAExL therefore buys run-to-run reproducibility as much as average accuracy. By contrast, $\mathbb{T}$-$\mathcal{W}_2$ is essentially flat across the three settings on all five systems, moving by at most a few percent and generally within one seed standard deviation. This is what one expects from a local, $p_{0|t}$-invariant kernel: MAExL relaxes energetic strain in the proposed x_0 configurations, the clashes and distorted bonds that dominate the energy distribution, but its moves are too local to transfer probability mass between torsional metastable states, so the dihedral marginals remain governed by the coverage of the learned model. The price is modest and shrinks in relative terms with system size, since on the larger systems the flow evaluations and the post-hoc MALA ladder dominate the inference budget. Returns also saturate well before the largest setting: 512 and 1024 steps are statistically indistinguishable for NF^{2}M-IS and nominally favour the shorter chains on the three largest systems, with only NF^{2}M-MIS showing a consistent preference for 1024 steps.

Table 4: MAExL ablation on ALA-N systems, with $N = 2, 3, 4, 6$, and on chignolin. Each column is the NF^{2}M configuration of Tab. 10, rerun with the number of in-loop MAExL steps per level given in the header; the 512-step columns are those of Tab. 1. Results are obtained from three random seeds, reported as `mean` $\pm$ `std`. Energy and torus Wasserstein distances are computed on the test split against 10^4 reference samples, from a proposal budget of 10^4 samples. Inference time covers sampling and the post-hoc MALA ladder.

		1024 MAExL steps		512 MAExL steps		0 MAExL steps	
		NF^{2}M-IS	NF^{2}M-MIS	NF^{2}M-IS	NF^{2}M-MIS	NF^{2}M-IS	NF^{2}M-MIS
ALA-2 ($d = 66$)	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.127_{\pm 0.019}$	$0.186_{\pm 0.042}$	$0.132_{\pm 0.013}$	$0.242_{\pm 0.080}$	$0.256_{\pm 0.054}$	$0.367_{\pm 0.035}$
	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$1.044_{\pm 0.019}$	$1.241_{\pm 0.010}$	$1.050_{\pm 0.011}$	$1.239_{\pm 0.006}$	$1.055_{\pm 0.009}$	$1.257_{\pm 0.023}$
	Inference (s) $\downarrow$	$129.2_{\pm 5.8}$	$118.7_{\pm 0.2}$	$137.2_{\pm 14.5}$	$125.8_{\pm 14.4}$	$122.8_{\pm 14.4}$	$116.6_{\pm 14.6}$
ALA-3 ($d = 99$)	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.167_{\pm 0.016}$	$0.151_{\pm 0.030}$	$0.164_{\pm 0.011}$	$0.157_{\pm 0.009}$	$0.215_{\pm 0.037}$	$0.212_{\pm 0.033}$
	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$0.497_{\pm 0.011}$	$0.471_{\pm 0.020}$	$0.490_{\pm 0.016}$	$0.465_{\pm 0.007}$	$0.505_{\pm 0.022}$	$0.462_{\pm 0.020}$
	Inference (s) $\downarrow$	$162.8_{\pm 9.8}$	$163.2_{\pm 10.7}$	$164.1_{\pm 0.4}$	$147.2_{\pm 13.1}$	$137.2_{\pm 12.0}$	$149.7_{\pm 0.8}$
ALA-4 ($d = 126$)	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$1.022_{\pm 0.073}$	$0.970_{\pm 0.100}$	$0.988_{\pm 0.027}$	$1.107_{\pm 0.013}$	$7.125_{\pm 11.571}$	$1.745_{\pm 2.142}$
	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$0.827_{\pm 0.011}$	$0.850_{\pm 0.063}$	$0.824_{\pm 0.041}$	$0.882_{\pm 0.051}$	$0.840_{\pm 0.055}$	$0.882_{\pm 0.024}$
	Inference (s) $\downarrow$	$193.7_{\pm 12.3}$	$174.8_{\pm 12.6}$	$184.3_{\pm 13.5}$	$162.3_{\pm 13.6}$	$159.4_{\pm 14.0}$	$148.0_{\pm 12.5}$
ALA-6 ($d = 189$)	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$1.070_{\pm 0.137}$	$1.084_{\pm 0.123}$	$1.018_{\pm 0.063}$	$1.301_{\pm 0.092}$	$2.077_{\pm 0.196}$	$2.649_{\pm 0.109}$
	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$1.013_{\pm 0.023}$	$1.019_{\pm 0.016}$	$0.990_{\pm 0.026}$	$1.027_{\pm 0.021}$	$0.994_{\pm 0.018}$	$1.016_{\pm 0.004}$
	Inference (s) $\downarrow$	$455.3_{\pm 1.5}$	$440.3_{\pm 13.4}$	$452.8_{\pm 13.9}$	$405.6_{\pm 14.2}$	$417.3_{\pm 5.2}$	$372.8_{\pm 9.3}$
Chignolin ($d = 414$)	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$4.966_{\pm 0.369}$	$5.734_{\pm 0.091}$	$4.331_{\pm 0.262}$	$6.112_{\pm 0.129}$	$10.155_{\pm 0.502}$	$9.129_{\pm 0.077}$
	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$2.588_{\pm 0.018}$	$2.579_{\pm 0.023}$	$2.597_{\pm 0.054}$	$2.581_{\pm 0.018}$	$2.594_{\pm 0.045}$	$2.573_{\pm 0.017}$
	Inference (s) $\downarrow$	$2945.3_{\pm 13.1}$	$2356.3_{\pm 16.1}$	$2830.6_{\pm 17.4}$	$2264.5_{\pm 1.1}$	$2727.2_{\pm 9.7}$	$2181.8_{\pm 14.4}$

A.4 Centre-of-mass adjustment for the conditional posterior

We follow the Gaussian centre-of-mass (CoM) augmentation of SBG (Tan et al., 2025) and extend it to our diffusion posterior. Unlike SBG, we do not use a radial correction: every weight below is built from the Cartesian density of the centroid. We first derive valid importance weights and then, separately, characterise when matching the centroid proposal improves the effective sample size (ESS).

A.4.1 Setup and notation

For $x = (x_1, \dots, x_N) \in \mathbb{R}^{Nd}$, let $c = N^{-1} \sum_i x_i$ be the unweighted centroid, which is what we mean by the CoM throughout, and let $\hat{x} = x - \mathbf{1}_N \otimes c$ be the centred configuration. Then

$$x = \hat{x} + \mathbf{1}_N \otimes c, \qquad \hat{x} \in \mathcal{H} := \Big\{ s \in \mathbb{R}^{Nd} : \textstyle\sum_i s_i = 0 \Big\}.$$

The two components are orthogonal, and $\dim \mathcal{H} = (N-1)d$. All densities below are written in the coordinates $(\hat{x}, c)$, with respect to Lebesgue measure on $\mathcal{H} \times \mathbb{R}^d$. The change of variables to Cartesian coordinates has a constant Jacobian, which we include whenever the flow density is expressed in these coordinates.

Let p be the normalised Boltzmann density on $\mathcal{H}$. Restricting to $\mathcal{H}$ removes the translational degree of freedom, which would otherwise make the density unnormalisable. As in SBG, we lift p to the ambient space by attaching an independent Gaussian centroid:

$$\tilde{p}(\hat{x}, c) = p(\hat{x})\, g_\sigma(c), \qquad g_\sigma(c) = \mathcal{N}(c; 0, \sigma^2 I_d), \qquad \sigma^2 = 1/N. \tag{71}$$

This variance is that of the centroid of N independent standard Gaussian vectors. Since g_σ integrates to one, the centred marginal of $\tilde{p}$ is exactly p.

A.4.2 Relation to SBG's unconditional adjustment

Like SBG, we introduce an auxiliary Gaussian centroid so that the model can be trained in the full ambient space; we differ in the density used for reweighting. For $d = 3$, the radius $r = \|c\|$ has density

$$f_\sigma(r) = \frac{1}{\sigma}\chi_3(r/\sigma) = 4\pi r^2 g_\sigma(c), \qquad \|c\| = r,$$

where the factor $4\pi r^2$ arises from integrating the spherical volume element over the angles. Thus f_σ is a density with respect to dr, whereas g_σ is a density with respect to dc.

SBG's Eq. (5) subtracts $\log f_\sigma(\|c\|)$ from the flow log-density. For the Gaussian lift (71), however, the correct unconditional importance weight is

$$w(x) \propto \frac{p(\hat{x})\, g_\sigma(c)}{q^\theta(x)}.$$

Rewriting the denominator as q^θ / g_σ amounts to adding $\|c\|^2/(2\sigma^2)$ to the flow log-density, up to an additive constant. We omit SBG's additional $-2\log\|c\|$ term, since it stems from the radial Jacobian rather than from the Cartesian Gaussian density.

A.4.3 The conditional Gaussian lift

We consider variance-exploding noising with $\alpha_t = 1$ and $\sigma_t > 0$:

$$X_t \mid X_0 \sim \mathcal{N}(X_0, \sigma_t^2 I_{Nd}).$$

The conditional flow $q^\theta_{0|t}$ is trained on pairs (x_0, x_t) generated by first augmenting with a Gaussian centroid and then applying this noising. The posterior over centred configurations is

$$p^\perp_{0|t}(\hat{x}_0 \mid \hat{x}_t) \propto p(\hat{x}_0) \exp\!\left(-\frac{\|\hat{x}_t - \hat{x}_0\|^2}{2\sigma_t^2}\right), \tag{72}$$

whereas the lifted model has posterior

$$\tilde{p}_{0|t}(x_0 \mid x_t) \propto \mathcal{N}(x_t; x_0, \sigma_t^2 I_{Nd})\, p(\hat{x}_0)\, g_\sigma(c_0). \tag{73}$$

Here and below, evaluating a density at x_0 is shorthand for evaluating it at $(\hat{x}_0, c_0)$ in the coordinates fixed above.

The squared distance splits orthogonally as

$$\|x_t - x_0\|^2 = \|\hat{x}_t - \hat{x}_0\|^2 + N\|c_t - c_0\|^2,$$

so the centroid is observed through the Gaussian likelihood

$$c_t \mid c_0 \sim \mathcal{N}(c_0, \sigma_t^2 I_d / N).$$

Combining this likelihood with the Gaussian prior g_σ yields the factorisation

$$\tilde{p}_{0|t}(x_0 \mid x_t) = p^\perp_{0|t}(\hat{x}_0 \mid \hat{x}_t)\, g_t(c_0 \mid c_t), \qquad g_t(c_0 \mid c_t) = \mathcal{N}(c_0; \eta_t c_t, \beta_t^2 I_d), \tag{74}$$

where

$$\begin{aligned} \eta_t &= \frac{\sigma^2}{\sigma^2 + \sigma_t^2/N} = \frac{1}{1+\sigma_t^2}, \\ \beta_t^2 &= \frac{\sigma^2(\sigma_t^2/N)}{\sigma^2 + \sigma_t^2/N} = \frac{\sigma_t^2}{N(1+\sigma_t^2)}, \end{aligned} \tag{75}$$

and the second equalities use $\sigma^2 = 1/N$. In particular, lifting before conditioning leaves the centred posterior (72) unchanged.

Rotations. If p is rotationally invariant, the lifted posterior is jointly rotation invariant: for every $R \in \mathrm{SO}(d)$, acting identically on each particle,

$$\tilde{p}_{0|t}(Rx_0 \mid Rx_t) = \tilde{p}_{0|t}(x_0 \mid x_t). \tag{76}$$

This is invariance under rotating x_0 and x_t together, not under rotating x_0 alone at fixed x_t. Neither the factorisation above nor the weights below require rotational invariance; both follow solely from the independent Gaussian lift and the orthogonal split of the likelihood.

A.4.4 Importance weights and effective log-likelihood

Fix t and x_t, and draw independent samples from the learned flow $q^{\theta}_{0|t}(\cdot \mid x_t)$, whose support we assume covers that of the lifted posterior. The importance weights are

$$w_t(x_0) = \frac{p^{\perp}_{0|t}(\hat{x}_0 \mid \hat{x}_t)\, g_t(c_0 \mid c_t)}{q^{\theta}_{0|t}(x_0 \mid x_t)}, \tag{77}$$

where normalising constants independent of x_0 may be dropped in practice. These weights do not require the learned flow to factorise. Indeed, since $g_t(\cdot \mid c_t)$ integrates to one, every integrable centred observable F satisfies

$$\mathbb{E}_{q^{\theta}_{0|t}}\big[w_t(X_0)\, F(\hat{X}_0)\big] = \int_{\mathcal{H}} F(s)\, p^{\perp}_{0|t}(s \mid \hat{x}_t)\, ds.$$

Taking $F \equiv 1$ gives $\mathbb{E}_{q^{\theta}_{0|t}}[w_t] = 1$, so self-normalised importance sampling is consistent for such observables.

To compare with SBG's log-likelihood adjustment, we define the *effective* log-likelihood

$$\begin{aligned} \log q^{\theta,\mathrm{eff}}_{0|t}(x_0 \mid x_t) &:= \log q^{\theta}_{0|t}(x_0 \mid x_t) - \log g_t(c_0 \mid c_t) \\ &= \log q^{\theta}_{0|t}(x_0 \mid x_t) + \frac{\|\delta\|^2}{2\beta_t^2} + \frac{d}{2}\log\big(2\pi\beta_t^2\big), \qquad \delta = c_0 - \eta_t c_t, \end{aligned} \tag{78}$$

so that $w_t = p^{\perp}_{0|t}/q^{\theta,\mathrm{eff}}_{0|t}$. This merely rewrites (77); it is not a new sampling proposal. In general, $q^{\theta,\mathrm{eff}}_{0|t}$ is neither normalised on the ambient space nor equal to the centred marginal of the flow.

Note that the first factor in the numerator of (77) is the *centred* posterior (72), not the lifted posterior (73). Equivalently, the entire numerator may be replaced by the lifted posterior, that is, the full likelihood times $p(\hat{x}_0)\, g_\sigma(c_0)$, which already contains the centroid factor. Either way, the Gaussian centroid factor must not be included twice.

A.4.5 When centroid matching improves ESS

For K independent proposal samples with weights $w_1, \ldots, w_K$, the effective sample size is

$$\mathrm{ESS}_K = \frac{\big(\sum_{i=1}^K w_i\big)^2}{\sum_{i=1}^K w_i^2}.$$

For a normalised target π and a proposal q whose support covers that of π, if $w = \pi/q$ has a finite second moment, then

$$\frac{\mathrm{ESS}_K}{K} \xrightarrow[K\to\infty]{\text{a.s.}} \rho(q,\pi) := \frac{(\mathbb{E}_q[w])^2}{\mathbb{E}_q[w^2]}.$$

Our comparison concerns this asymptotic fractional ESS, not the ESS of any particular finite sample.

Proposition 20 (Matching the centroid proposal) *Fix t and x_t, and write $P(s) = p^{\perp}_{0|t}(s \mid \hat{x}_t)$ and $g(c) = g_t(c \mid c_t)$. Consider the lifted target $\pi(s,c) = P(s)\, g(c)$ and the two proposals*

$$q(s,c) = Q(s)\, k(c), \qquad q_{\mathrm{match}}(s,c) = Q(s)\, g(c).$$

Assume that all densities are normalised, that $Q > 0$ wherever $P > 0$ and $k > 0$ wherever $g > 0$ (up to null sets), and that

$$\mathbb{E}_Q\big[(P/Q)^2\big] < \infty, \qquad \mathbb{E}_k\big[(g/k)^2\big] < \infty.$$

Then

$$\rho(q,\pi) \le \rho(q_{\mathrm{match}},\pi), \tag{79}$$

with equality if and only if $k = g$ almost everywhere.

Proof. Under q, the weight factorises as $w = A(S)\,B(C)$ with $A = P/Q$ and $B = g/k$. These factors are independent and each has expectation one, so

$$\rho(q,\pi) = \frac{1}{\mathbb{E}_Q[A^2]\,\mathbb{E}_k[B^2]} \le \frac{1}{\mathbb{E}_Q[A^2]} = \rho(q_{\mathrm{match}},\pi),$$

where the inequality follows from $\mathbb{E}_k[B^2] \ge (\mathbb{E}_k B)^2 = 1$. Equality holds exactly when $B = 1$ k-almost surely, that is, when $k = g$ almost everywhere. □

The proof relies on the same independent-weight factorisation as SBG (Tan et al., 2025, Appendix B.1). It differs from their Proposition 1 in that it compares two normalised proposals against a single fixed, normalised target, under explicit moment assumptions. It is therefore a statement about proposal matching and provides no additional ESS guarantee for the effective log-likelihood.

Several consequences follow. If the flow already factorises as $q^{\theta}_{0|t} = Q\,g_t$, its ordinary lifted-target weights reduce to P/Q, because the centroid factor cancels; if moreover $Q = P$, all weights are constant. For a mismatched factorised proposal, realising the improvement requires actually sampling from $Q\,g_t$ and evaluating its density; modifying the likelihood in the denominator alone is not enough. Finally, for a generic non-factorised flow, the weights (77) remain valid, but the ESS comparison no longer applies.

Limits. For fixed x_t, as $\sigma_t \to \infty$ we have $\eta_t \to 0$ and $\beta_t^2 \to \sigma^2$, which recovers the unconditional *Gaussian* adjustment above rather than SBG's radial formula. As $\sigma_t \to 0$, we have $\eta_t \to 1$ and $\beta_t^2 \to 0$, so the centroid posterior concentrates at c_t. At zero noise it becomes singular, which is why the density formulas are stated only for $\sigma_t > 0$. Under g_t, writing $\delta = \beta_t Z$ with $Z \sim \mathcal{N}(0, I_d)$ gives

$$-\log g_t(c_0 \mid c_t) = \frac{\|Z\|^2}{2} + \frac{d}{2}\log\bigl(2\pi\beta_t^2\bigr).$$

The second term is constant in x_0 and cancels under self-normalisation, but the sample-dependent first term does not vanish at low noise. It does, however, cancel exactly in the importance ratio at every positive noise level whenever the proposal centroid matches g_t.

A.5 Conditional TarFlow and StarFlow

Sec. 3.2 introduces our conditional molecular TarFlow and StarFlow backbones and the two mechanisms used to inject x_t. This appendix gives the detail that does not fit there: a brief recap of the two base architectures (Sec. A.5.1, pointing to their own papers for anything not needed below), exactly how time and space conditioning are added on top of them (Sec. A.5.2), the cache that makes conditioning nearly free at sampling time (Sec. A.5.3), and the resulting time and space complexity of every operation our sampler uses (Sec. A.5.4).

A.5.1 TarFlow and StarFlow

TarFlow. TarFlow (Zhai et al., 2025) is an autoregressive normalising flow built from causal-Transformer coupling layers. Given tokens $x = (x_1, \dots, x_N)$, $x_i \in \mathbb{R}^D$, one layer computes

$$z_i = \bigl(x_i - \mu_i(x_{<i})\bigr) \odot \exp\bigl(-\alpha_i(x_{<i})\bigr), \qquad i = 1, \dots, N, \tag{80}$$

with $z_1 \equiv x_1$ and (μ_i, α_i) produced by a single causal-Transformer pass over the prefix $x_{<i}$; the Jacobian is lower-triangular, with $\log|\det \partial z/\partial x| = -\sum_i \mathbf{1}^\top \alpha_i(x_{<i})$. Training computes every (μ_i, α_i) in one parallel pass under a causal mask; sampling reruns the same Transformer autoregressively, with a standard KV cache. A deep flow stacks L such layers, permuting the sequence between layers so every coordinate sits both early and late in the autoregressive order across the stack. We refer to Zhai et al. (2025) for the full architecture, including the image-patch tokenisation, which does not affect anything below.

StarFlow. StarFlow (Gu et al., 2026) scales the same recipe with two changes we adopt. First, *deep–shallow depth allocation*: rather than L equal layers, one *deep* block of ℓ layers generates a first sample from noise and $L - 1$ *shallow* blocks of two layers each refine it, written $\ell(L)$-d for width d.

Second, a *guarded affine head*: TarFlow's $\exp(\alpha_i)$ overflows once $|\alpha_i| \approx 88$, so StarFlow soft-clips the pre-activation and passes it through a softplus instead,

$$\tilde{\alpha}_i = s\tanh(\alpha_i/s), \qquad \sigma_i = \mathrm{softplus}\big(\tilde{\alpha}_i + \mathrm{softplus}^{-1}(1)\big), \qquad z_i = (x_i - \mu_i)/\sigma_i, \tag{81}$$

with $s = 4$, which we use throughout, including for conditional TarFlow. StarFlow's remaining two contributions (a pretrained-autoencoder latent space and a classifier-free guidance rule) are specific to text-to-image generation: an autoencoder would break the exactness of $\log p_\theta$ that our importance sampler needs, and there is no class or text label to guide on. We do not adopt either; see Gu et al. (2026) for both.

Molecular adaptation. Following Tan et al. (2025), we tokenise an N-atom configuration $x \in \mathbb{R}^{N\times 3}$ as one token per atom, so StarFlow's 2-D patch grid becomes a degenerate $1 \times N$ image, and handle the target's $\mathrm{SE}(3)$ symmetry outside the architecture through centering, a lifted centroid, and random rotation augmentation rather than through an equivariant backbone, Sec. 3.2 gives the details we share with the main text. Our molecular blocks use learned per-block positional embeddings rather than rotary ones, a choice Sec. A.5.3 returns to, full multi-head attention, GELU feed-forwards, and the guarded affine head of Eq. (81). Per-system depth and width are given in App. B.2.

A.5.2 Making the flow conditional on (x_t, t)

Both backbones are made *diffusion-ready* by turning the unconditional density $p_\theta(x_0)$ into the posterior $p^\theta_{0|t}(x_0|x_t)$ of Sec. 3.2: a small continuous t and an entire noisy configuration $x_t \in \mathbb{R}^{N\times 3}$ are injected into every block without disturbing the autoregressive structure or the KV-cached sampling cost. Fig. 6 gives the resulting picture before the two paragraphs below detail the conditioning pathways themselves, and Fig. 7 zooms into a single block.

Time, via AdaLN-Zero. We follow DiT (Peebles & Xie, 2023): t is embedded by a sinusoidal map and a two-layer MLP into a single vector $c \in \mathbb{R}^C$, which drives an AdaLN-Zero modulation in every attention layer of every block. A $\mathrm{SiLU} \to \mathrm{Linear}(C, 9C)$ produces a (shift, scale, gate) triple for each of the self-attention, cross-attention (when present) and MLP sub-layers, and every one of these modulation linears is zero-initialised. The unconditional flow already starts at the identity because its affine head is zero-initialised, while attention and the MLP still run live. The conditional flow starts at a strictly stronger identity, since every gate is zero too, so attention, cross-attention and the MLP contribute nothing at all, on top of the zero affine head.

Space, via cross-attention or a causal prefix. x_t enters through one of two mechanisms, both shown in Fig. 7. *Cross-attention* projects x_t once per block through a shared $\mathbb{R}^3 \to \mathbb{R}^C$ map, adds the same positional code used for the running x_0 representation and lets each layer, every layer or a strided subset every m-th one, project its own keys and values from that shared embedding and attend to it from a query built out of x_0. *Prefix conditioning* instead concatenates the projected, positionally-coded x_t ahead of x_0 along the sequence axis and lets the block's existing causal self-attention do the work: writing π for the block's permutation and $W_{\mathrm{in}}, W_{\mathrm{cond}}$ for the two input projections,

$$z = \big[\, W_{\mathrm{cond}} x_t^\pi + p^\pi + c\mathbf{1}^\top \,;\, W_{\mathrm{in}} x_0^\pi + p^\pi \,\big] \in \mathbb{R}^{2N\times C}, \tag{82}$$

reading $(\mu_i, \tilde{\alpha}_i)$ off the last N positions only. Because attention is causal and the prefix leads, the first N positions never depend on x_0, so $x_0 \mapsto z$ is still autoregressive with the same lower-triangular Jacobian. Every conditioning atom is simply visible to every query atom through the same attention operator that carries the autoregressive chain, at the cost of attending over $2N$ rather than N tokens and no parameters beyond W_{cond}.

Both mechanisms are available to either backbone, but their scope differs. Conditional TarFlow conditions every block while conditional StarFlow restricts conditioning to the last (deep) block only, while every block still receives t through its own AdaLN-Zero. In practice, cross-attention is what nearly every reported model uses: every TarFlow model, and StarFlow on every system but chignolin. Chignolin,the largest system with the deepest deep block, is the sole case where the prefix variant is selected instead, over its cross-attention twin at matched depth (Tabs. 9 and 10).

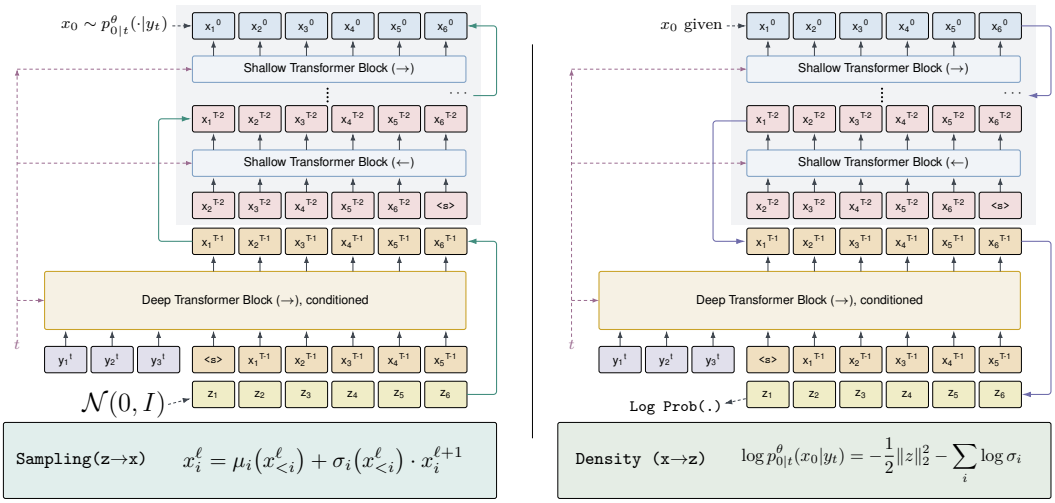


Figure 6: **Sampling and density evaluation share one stack.** The same L blocks (a deep block conditioned on y_t, shallow blocks refining) run in two directions. *(a)* Autoregressively, from $z \sim \mathcal{N}(0, \mathrm{I})$ up to a sample $x_0 \sim p^{\theta}_{0|t}(\cdot|y_t)$: N sequential per-atom steps per block, under a KV cache. *(b)* In parallel, from a given x_0 down to $\log p^{\theta}_{0|t}(x_0|y_t)$: one causal pass per block. Here y_t denotes the noisy configuration x_t of Sec. 3.2, relabelled so the conditioning is never mistaken for another stage of the flow's own $x^0, \ldots, x^{T-1}$ sequence. The green/purple arrows are the inverse/forward instance of the same per-atom update between every pair of adjacent levels, $x_i^{\ell} = \mu_i(x^{\ell}_{<i}) + \sigma_i(x^{\ell}_{<i}) \cdot x_i^{\ell+1}$ (bottom left), $x_i^{\ell+1}$ is literally z_i only at the very first level. y_t is shown entering as a causal prefix at the deep block only. Fig. 7 shows the cross-attention alternative and the every-block scope TarFlow uses instead (Sec. A.5.2), plus the AdaLN-Zero time conditioning we add on top of both shown here only as the dashed t arrows reaching every block, deep and shallow alike (the modulation itself is not drawn). Both directions read the same conditioning cache, built once from (y_t, t) and shared across every particle drawn at that step (Sec. A.5.3). Layout follows Gu et al. (2026)'s own figure.

A.5.3 The conditioning cache

x_t is fixed across every particle drawn from the same posterior: one denoising step of NF^{2}M-IS/MIS scores and draws M (or CM) particles against a single (x_t, t). Recomputing the conditioning pathway from scratch for each of them (the time embedding, the projected x_t tokens, and every conditioned layer's cross-attention or prefix keys and values) would repeat identical work M times over.

Instead, a dedicated preparation step builds this state once per distinct (x_t, t), and every subsequent call to the flow takes it as an argument together with an index selecting each particle's row, so only the row lookup runs per particle. We have verified this reduction bit-for-bit against recomputing the conditioning independently for every particle. For StarFlow under prefix conditioning this goes one step further: the object the preparation step builds is not a separate cache that later gets copied in, but literally the same key/value buffer the autoregressive sampling loop appends to, so pre-filling and decoding share one piece of memory rather than two.

A.5.4 Computational complexity

We summarise the per-evaluation cost of the molecular TarFlow and StarFlow variants. Let N be the number of atoms, C the Transformer channel width and L the number of blocks, with block ℓ carrying K_ℓ attention layers, so that the full flow contains $K_{\mathrm{tot}} = \sum_{\ell=1}^{L} K_\ell$ layers. The TarFlow models set $K_\ell = K$ for every block, giving $K_{\mathrm{tot}} = LK$ and the StarFlow models set $K_\ell = 2$ for the shallow blocks and $K_\ell = K_{\mathrm{deep}}$ for the deep ones. We write K_{cond} for the number of layers that carry the x_t conditioning and m for the cross-attention period, so that $m = 1$ places cross-attention on every layer. Tab. 5 gives the cost of a single call into the flow, Tab. 6 the cost once the conditioning cache of Sec. A.5.3 is in place.

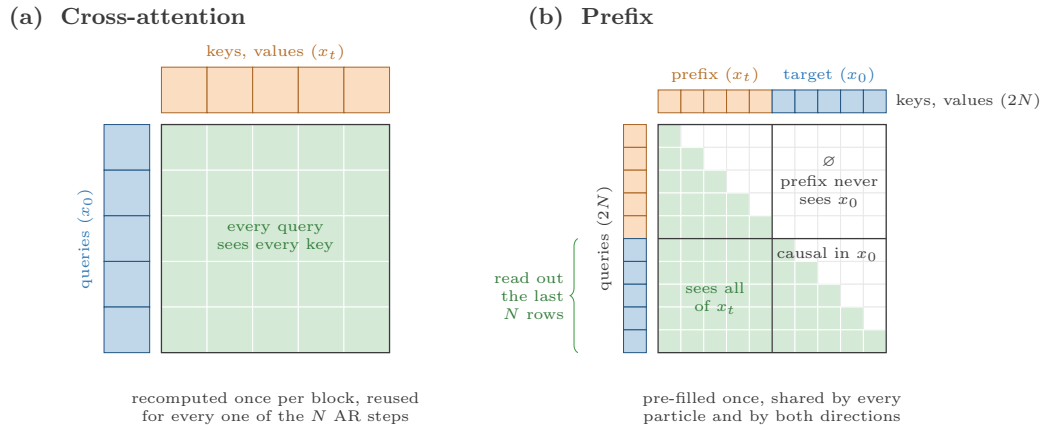


Figure 7: **The two conditioning mechanisms, inside one block.** *(a) Cross-attention:* x_t is embedded once per block, and each of the K layers still projects its own keys and values from that shared embedding and attends to it from a query built out of x_0; every sub-layer also carries an AdaLN-Zero modulation driven by the time embedding c, with every modulation linear zero-initialised, so the block is the identity map at init. *(b) Prefix:* x_t instead joins the sequence as a causal prefix (Eq. (82)). A plain lower-triangular mask over the concatenated $2N$ tokens already gives every target atom access to every conditioning atom while keeping the prefix invisible to x_0, so the added cost is a longer attention span and W_{cond}, not a new sub-layer. In both variants the conditioning state is fixed across every particle sharing x_t, hence cacheable (Sec. A.5.3).

Table 5: Cost of one call into the flow, dropping constants, with nothing precomputed: every call rebuilds the conditioning state from x_t. Notation as in the text. The *flow backbone* column is what every variant pays for the flow itself; the other two are what conditioning adds on top of it.

		Added by conditioning	
	Flow backbone	cross-attention	prefix
$\log p$	$K_{\text{tot}}(N^2C + NC^2)$	$\frac{1}{m}K_{\text{cond}}(N^2C + NC^2)$	$K_{\text{cond}}(3N^2C + NC^2)$
$x \sim p$	$K_{\text{tot}}(\frac{1}{2}N^2C + NC^2)$	$\frac{1}{m}K_{\text{cond}}(N^2C + NC^2)$	$K_{\text{cond}}(2N^2C + NC^2)$
additional params.	—	$\frac{1}{m}K_{\text{cond}} \cdot 4C^2$	0

One caveat comes first, because it changes which term matters. Every Transformer layer costs $\mathcal{O}(N^2C + NC^2)$: the attention matrix, plus the Q/K/V and feed-forward linears. At the sizes we run, N between 22 (alanine dipeptide) and 138 (chignolin), C between 256 and 512, the ratio of the two terms is C/N, so the linear term dominates by an order of magnitude on the smallest system and still by roughly $4\times$ on the largest. The quadratic-in-N attention cost is therefore *not* the binding constraint at this scale, and a forward pass is close to linear in the number of atoms.

Forward pass (density evaluation). All variants compute the entire collection $\{(\mu_i, \alpha_i)\}_{i=1}^N$ in a single parallel call with a causal attention mask, at a total cost of $\mathcal{O}(K_{\text{tot}}(N^2C + NC^2))$. Cross-attention conditioning adds one further attention against x_t, also $\mathcal{O}(N^2C + NC^2)$, on every m-th layer of the conditioned blocks, placing it on every layer ($m = 1$) roughly doubles the per-layer constant, and larger m scales that addition by $1/m$. Prefix conditioning instead runs the conditioned block over a sequence of length $2N$, costing $\mathcal{O}(K_{\text{cond}}(4N^2C + 2NC^2))$; once the prefix keys and values have been pre-filled, only the N target positions need to be projected and attended, which reduces this to $\mathcal{O}(K_{\text{cond}}(2N^2C + NC^2))$. Neither form of conditioning changes the asymptotics, and prefix conditioning adds no parameters beyond the shared $\mathbb{R}^3 \to \mathbb{R}^C$ projection.

Reverse pass (sampling). Sampling is inherently sequential along the autoregressive axis: producing token $i + 1$ requires the predictions $(\mu_{i+1}, \alpha_{i+1})$, which depend on the prefix $z_{1:i+1}$. With

Table 6: Amortising the conditioning. x_t is fixed across every x_0 drawn from the same posterior, so the conditioning state is built once per distinct x_t and read by every particle that conditions on it, in both directions of the flow. Entries are the cost *added* to the flow backbone; notation as in Tab. 5. Amortising deletes the projection terms from the per-particle cost, leaving attention against the cached state; what remains is the state itself, which scales with the number of conditioning rows held live rather than with the number of particles.

	cross-attention		prefix	
	uncached	amortised	uncached	amortised
Per particle				
density evaluation	$\frac{1}{m}K_{\text{cond}}(N^2C + NC^2)$	$\frac{1}{m}K_{\text{cond}}N^2C$	$K_{\text{cond}}(3N^2C + NC^2)$	$K_{\text{cond}}N^2C$
sampling	$\frac{1}{m}K_{\text{cond}}(N^2C + NC^2)$	$\frac{1}{m}K_{\text{cond}}N^2C$	$K_{\text{cond}}(2N^2C + NC^2)$	$K_{\text{cond}}N^2C$
Once per distinct x_t				
build	—	$\frac{1}{m}K_{\text{cond}}NC^2$	—	$K_{\text{cond}}(N^2C + NC^2)$
state retained	—	$2K_{\text{cond}}NC$	—	$2K_{\text{cond}}NC$

KV caching, each step extends the cached keys and values by one token and computes a query of length one against a prefix of length $i + 1$, costing $\mathcal{O}((i{+}1)C + C^2)$ per attention layer. Summing over $i = 0, \ldots, N{-}1$ gives the same total $\mathcal{O}(K_{\text{tot}}(N^2C + NC^2))$ as the forward pass, but with a substantially larger constant due to the N sequential kernel launches.

Conditioning does not change that order, and neither mode lengthens the autoregressive chain: the conditioning is either attended in parallel with the decode step or pre-filled in a single parallel pass, so a draw is N sequential steps whether or not the flow is conditioned. For cross-attention, the projection of x_t and its keys and values are computed once per block, outside the autoregressive loop, and reused for all N steps; each step then costs $\mathcal{O}(NC + C^2)$, contributing N^2C per conditioned layer against the $N^2C/2$ of the causal loop. For prefix conditioning, a single pre-fill of $\mathcal{O}(K_{\text{cond}}(N^2C + NC^2))$ populates the cache, after which step i attends over $N + i$ keys rather than i; the sum is $3N^2/2$ against $N^2/2$, i.e. three times the attention term of the bare causal loop, with the NC^2 term unchanged. The real price is memory: the KV cache doubles in length, and its prefix half must be retained for as long as its conditioning row is alive. For the deep block of our chignolin model (20 layers, 138 conditioning tokens, 8 key/value heads of dimension 64, in bfloat16) that is roughly 5.5 MiB per conditioning row, which is why the forward and reverse consumers read one shared store rather than each building its own (Sec. A.5.3).

The two conditioning modes therefore cost the same per decode step, $K_{\text{cond}}N^2C$ of extra attention, and differ only in what surrounds it: cross-attention pays for it in parameters, prefix conditioning pays for it in a pre-fill and a doubled cache, and the pre-fill is amortised over every particle that shares the conditioning row.

Finally, the deep–shallow allocation reshapes where the sequential cost sits. Each of the $L - 1$ shallow blocks contributes only two layers, so the reverse pass is dominated by K_{deep}; at a fixed K_{tot} the StarFlow allocation concentrates the sequential work in the same block that carries the conditioning, which is what makes the "condition the deep block only" choice of Sec. A.5.2 nearly free.

B Experimental details

B.1 Datasets and Metrics

Datasets. We reuse the molecular dynamics datasets of Tan et al. (2025); we briefly summarise them here for completeness and refer the reader to their Appendix D for full details. All datasets consist of all-atom Cartesian configurations sampled from MD trajectories generated with classical force fields in implicit solvent using OpenMM (Eastman et al., 2017). For each system, the first $1\mu s$ of trajectory is used for training, the next $0.2\mu s$ for validation, and the remainder for test, except for alanine dipeptide which follows the split of Klein & Noé (2024). Some metastable states may be missing from the training set, which is the realistic regime in which we evaluate our models.

The systems considered, in increasing order of size, are:

- **Alanine dipeptide** (Amber ff99SBildn, 300K): a single alanine residue capped by an acetyl group and an N-methyl group. We use the dataset and split of Klein & Noé (2024), in which the lower-probability φ state is intentionally over-represented in training to provide a tractable but biased setting.
- **Trialanine** (Amber 14, 310K): three alanine residues without capping groups, generated as in Klein et al. (2023a).
- **Alanine tetrapeptide** (Amber ff99SBildn, 300K): three alanines with acetyl and N-methyl caps. We use the system setup of Dibak et al. (2022) but treat all bonds as flexible (the original setup constrained hydrogen bonds for use with internal-coordinate flows).
- **Hexa-alanine** (Amber 14, 310K): six alanine residues without capping groups.
- **Chignolin** (Amber 14, 310K): the 10-residue protein `GYDPETGTWG`, 138 atoms, simulated for $4.5\mu s$. Notable for forming a β-hairpin in solvent (Honda et al., 2004).

All MD simulations use a 1fs integration time step. We use the same train/val/test trajectory splits, force fields, and evaluation subsets (10^4 uniformly strided samples from the held-out portion) as Tan et al. (2025) to ensure direct comparability with their reported numbers.

Target energy evaluation. While the datasets themselves are generated by OpenMM MD, we do not call OpenMM to evaluate the target energy and its gradient at inference time: an OpenMM `Context` evaluates one configuration at a time with no batched interface, making it the dominant cost of importance-weighted and MCMC-refined evaluation. We instead reimplement the same potential (bonded, Lennard-Jones/Coulomb, and Generalized-Born terms, OBC1 or OBC2 depending on the system) as a batched PyTorch module whose parameters are read directly from the `openmm.System`, so OpenMM remains the sole authority on the force field, atom typing, and topology; only the summation, evaluated as vectorized tensor operations with exact autograd gradients in float32, is reimplemented. Validated against OpenMM's double-precision Reference platform on 3.4×10^6 configurations spanning all splits plus perturbed and clashing geometries, the worst-case deviation below $10^2\, k_BT$ is $1.2 \times 10^{-5}\, k_BT$ in float64 and $7.7 \times 10^{-4}\, k_BT$ in float32 (more accurate than OpenMM's own CPU platform), whose internal single-precision nonbonded and GB evaluation deviates from the Reference platform by up to $3.3 \times 10^{-4}\, k_BT$ on typical frames and $126\, k_BT$ near atomic clashes. This batched evaluation is 30–400$\times$ faster per configuration than OpenMM's serial loop and, unlike OpenMM's hard failure on non-finite energies, returns a large but finite, differentiable energy at clashing configurations, avoiding sampler crashes during exploration.

Metrics. For computational efficiency we subsample 10^4 reference configurations from the held-out trajectory to serve as ground truth, and likewise restrict generated samples to a random subset of size 10^4 when more are produced. We quantify distributional similarity using empirical Wasserstein-2 distances between generated and reference samples. Given two empirical distributions $p = \frac{1}{n}\sum_i \delta_{x_i}$ and $q = \frac{1}{m}\sum_j \delta_{y_j}$,

$$W_2(p, q) = \min_{\pi \in \Pi(p,q)} \sqrt{\sum_{i,j} \pi_{ij} c(x_i, y_j)^2},$$

where the optimal coupling is computed with the POT library (Flamary et al., 2021). We use two cost functions:

- **Energy distance ($\mathcal{E}$-$\mathcal{W}_2$):** the squared cost is $c_{\mathcal{E}}(x, y)^2 = |E(x) - E(y)|^2$ where E is the force-field potential energy. This metric is highly sensitive to fine-grained structural details (bond lengths, angles, clashes).
- **Dihedral torus distance ($\mathbb{T}$-$\mathcal{W}_2$):** for a peptide with L residues we extract the backbone dihedral angles $\mathrm{Dihedrals}(x) = (\varphi_1, \psi_1, \ldots, \varphi_{L-1}, \psi_{L-1})$, and use the wrapped squared cost on the torus

$$c_{\mathbb{T}}(x, y)^2 = \sum_{i=1}^{2(L-1)} \left[(\mathrm{Dihedrals}(x)_i - \mathrm{Dihedrals}(y)_i + \pi) \bmod 2\pi - \pi\right]^2.$$

  This captures macrostructural information such as metastable-state occupancy and is robust to fine-grained noise.

Table 7: Unconditional TarFlow architecture.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
Channels	256	256	256	384	512
Blocks	4	6	6	6	8
Layers / block	4	6	6	6	8

Table 8: Unconditional StarFlow architecture. Every variant (here and in Tab. 9) follows the pattern layers/block $= [2, 2, 2, 2, n, n]$: the first four blocks are fixed at 2 layers, and only the last two ("top-block depth" n) vary.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
Channels	256	256	256	384	512
Blocks	6	6	6	6	6
Head dim	64	64	64	64	64
Top-block depth n	4	14	14	14	28

B.2 Hyperparameters

B.2.1 Training

The unconditional flows (TarFlow/StarFlow) and NF^{2}M share one training recipe: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$) under bf16 mixed precision, gradient-norm clipping at 1.0, and an EMA of the weights with decay 0.999. Each model trains for 1000 epochs (195k steps) at batch size 512 over the 10^5 training frames of the corresponding single-peptide system, on a single GPU, or with standard data-parallel DDP when more than one is used.

Unconditional TarFlow/StarFlow. Following Tan et al. (2025), the flows are trained by maximum likelihood with learning rate 10^{-4} under a one-cycle schedule (5% warmup, division factor 500, final division factor 1) and weight decay 4×10^{-4}. Tab. 7 gives the per-system TarFlow architecture, and Tab. 8 the alternative StarFlow architecture.

NF^{2}M. NF^{2}M trains with the loss of Sec. 3.2, under an EDM noise schedule ($\sigma_{\min} = 10^{-3}$, $\sigma_{\max} = 40$) (Karras et al., 2022) and uniform weighting of the loss across noise levels. Every run uses the same optimizer recipe as the unconditional flows: learning rate 10^{-4}, one-cycle schedule, weight decay 4×10^{-4}, 1000 epochs / 195k steps at batch 512.

The conditional backbone is one of six variants, each defined relative to the corresponding unconditional preset (Tables 7 and 8):

- **T1** – a shallow TarFlow backbone, conditioned by cross-attention injected every 4th block;
- **K1** – the same shallow backbone as T1, but with cross-attention injected at every block;
- **W1** – the same shallow backbone as T1, widened rather than left narrow (channels $+64$);
- **S1** – a shallow-top-block StarFlow, conditioned through a single prefix token rather than cross-attention;
- **S1k** – the same shallow-top-block StarFlow as S1, but conditioned by cross-attention rather than a prefix token;
- **S2** – StarFlow with no depth cut at all, conditioned by cross-attention.

Cross-attention stride is $\min(4, l)$, where l is the variant's (top-block) depth, so it falls back from every 4th block to every block on the shallowest variants. The three TarFlow variants (T1, K1, W1) fail to train on chignolin and are not reported there; only S1, S1k, and S2 are available for that system. Tab. 9 gives the resulting shape of every variant on every system.

For each system, we choose the architecture variant and number of steps ($K = 5$ or $K = 8$) according to the validation performance of NF^{2}M-IS. For NF^{2}M-MIS, selection instead uses NF^{2}M-IS performance, restricted to $K = 8$ since the mixture is only efficient at that setting. Tab. 10

Table 9: NF^{2}M architecture sweep. TarFlow variants report channels / blocks $\times$ layers-per-block / conditioning stride; StarFlow variants report channels / top-block depth n (layers/block $= [2, 2, 2, 2, n, n]$) / conditioning stride. "–" marks variants that failed to train.

Variant	Conditioning	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
T1	cross-attn (stride 3–4)	256/4×3/3	256/6×4/4	256/6×4/4	384/6×4/4	–
K1	cross-attn (stride 1)	256/4×3/1	256/6×4/1	256/6×4/1	384/6×4/1	–
W1	cross-attn (stride 3–4)	320/4×3/3	320/6×4/4	320/6×4/4	448/6×4/4	–
S1	prefix	256/3	256/10	256/10	384/10	512/20
S1k	cross-attn (stride 3–4)	256/3/3	256/10/4	256/10/4	384/10/4	512/20/4
S2	cross-attn (stride 4)	256/4/4	256/14/4	256/14/4	384/14/4	512/28/4

Table 10: Selected NFM checkpoint per system, with its architecture (same notation as Tab. 9: TarFlow variants as channels / blocks $\times$ layers-per-block / conditioning stride, StarFlow variants as channels / top-block depth / conditioning stride or "prefix"). K is the number of inference-time levels; training time is the wall-clock of the 1000-epoch run of the checkpoint actually loaded.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
NF^{2}M-IS					
Variant	W1	W1	T1	S2	S1
Architecture	320/4×3/3	320/6×4/4	256/6×4/4	384/14/4	512/20 (prefix)
K	5	8	8	5	5
Training (h)	4.3	6.0	5.3	12.8	21.7
NF^{2}M-MIS					
Variant	W1	W1	T1	S1k	S1
Architecture	320/4×3/3	320/6×4/4	256/6×4/4	384/10/4	512/20 (prefix)
K	8	8	8	8	8
Training (h)	4.3	6.0	5.3	10.2	21.7

reports both selected configurations, together with the wall-clock training time of the corresponding checkpoint.

Velocity teacher. Before turning to the flow-map baselines (FALCON, FALCON-A, F2D2, SCALLOP), we describe the velocity field they all warm-start from and which we also evaluate as a standalone flow-matching baseline in the style of ECNF++ (Tan et al., 2025). A velocity field v_ϕ is trained by flow matching on the linear interpolant $x_t = (1 - t)\, x_0 + t\, \varepsilon$, with $t \in [10^{-4}, 1 - 10^{-4}]$: $t = 0$ is data and $t = 1$ is the prior. The network is the DiT backbone (Ying et al., 2021; Peebles & Xie, 2023) of Tab. 11. We use the antithetic variance reduction trick for the loss (Albergo et al., 2025). One teacher is trained per system, shared by all four flow-map methods and both of their pretraining objectives (below), and is always read at its EMA weights. Unlike every phase that reads it, the teacher trains against an *ambient* $\mathcal{N}(0, I)$ prior with centre-of-mass augmentation on the data. Its own network removes the mean from input and output, so what it supplies downstream is the mean-free component of the velocity. Its optimizer settings, step count, and per-step cost are given alongside the flow-map phases that follow it, in Tab. 13 ("Velocity, phase 1"), Tab. 16, and Tab. 17 respectively, since its own recorded wall-clock is what the flow-map training budget below is computed from.

FALCON, FALCON-A, F2D2 and SCALLOP. We follow the original papers for the architecture and optimizer hyperparameters of these four flow-map baselines (FALCON/FALCON-A: Rehman et al. (2026a); F2D2: Ai et al. (2026); SCALLOP: OuYang et al. (2026)). All four continue from the velocity teacher described above, through a two-phase pipeline : flow-map pretraining (Phase 1.5) and the distillation objective itself (Phase 2). Each phase frozen before the next one reads it. The original FALCON paper does not use a pretrain-then-distill procedure; we adopt it uniformly across all four baselines for consistency with F2D2 and SCALLOP, and since it starts from an independently pretrained velocity field, it can only match or improve on FALCON's original recipe.

Table 11: DiT backbone for the phase-1 velocity teacher.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
d_{model}	192	192	192	192	384
Layers	6	6	6	6	12
Heads	6	6	6	6	12
MLP ratio	4.0	4.0	4.0	4.0	4.0
Time-embedding dim	64	64	64	64	128

The flow map itself extends the teacher's time convention: $X(x_t, t, s) = x_t + (s - t)\, u_\theta(x_t, t, s)$ with $s < t$. All three papers run the opposite direction; each identity below is the reversal of the printed one, and each is invariant under it.

Phase 1.5 — flow-map pretraining. One self-consistency identity for the average-velocity map $u_\theta(x_t, t, s)$, trained against the frozen teacher on the same single-head backbone Phase 2 uses. Two objectives, neither of them new mathematics:

- **ESD** (Eulerian), the MeanFlow identity $u_\theta = v + (s - t)\,(v \cdot \nabla_x u_\theta + \partial_t u_\theta)$, which is FALCON's average-velocity term standing alone;
- **LSD** (Lagrangian), $u_\theta(x_t, t, s) + (s - t)\,\partial_s u_\theta(x_t, t, s) = \mathrm{sg}[u_\theta](x_s, s, s)$ at the mapped point x_s, which is F2D2's and SCALLOP's shared self-distillation term with no divergence head attached.

We encourage to look at Boffi et al. (2025) for more details. ESD warm-starts FALCON and FALCON-A, LSD warm-starts F2D2 and SCALLOP, so there are two pretraining runs per system rather than four. The student loads the pretraining checkpoint's EMA weights only: fresh optimizer, fresh schedule, step counter back to zero. For F2D2/SCALLOP the single-head parameters are remapped onto the dual-head layout (blocks → trunk, head → velocity head) and the divergence head is left at its zero initialization.

Phase 2 — distillation. The four methods proper, on the single-head backbone (FALCON, FALCON-A) or the dual-head backbone with its separate divergence-readout head (F2D2, SCALLOP); Tab. 12 gives the sizes. Tab. 13 gives the optimizer settings of all three phases and Tables 14 and 15 the per-objective loss hyperparameters. The teacher enters phase 2 in three places: it is the regression target of the instantaneous/diagonal term, the tangent of the average-velocity JVP (FALCON, ESD), and the field whose divergence F2D2 estimates.

Common to every phase. The whole problem is posed in the mean-free subspace so the map transports the centroid by the identity. Both times reach the network through a Fourier embedding of *raw* t. F2D2, SCALLOP and LSD condition on the anchor and the step $s - t$; FALCON, FALCON-A and ESD on the two times independently, as MeanFlow (Geng et al., 2025) does. The choice has to agree across a warm start, which is why the pretraining rows match the students they initialize. Training runs under bf16 mixed precision but the log-density accumulator is in fp32. Each phase keeps an EMA of decay 0.999 and skips the optimizer step on a non-finite loss or gradient norm. Chignolin trains 4-way DDP at the same global batch of 1024, every other system on a single GPU.

Training budget. OuYang et al. (2026, Appendix C.1) reports a budget of 4×10^6 update steps at batch 1000, but this figure is inconsistent with the paper's own wall-clock numbers: at our measured per-step time on our GPU class, the reported step count would take an order of magnitude longer than the ∼7–8 h the paper reports for its two phases combined. We therefore size the pipeline by wall-clock instead: the whole of it (all three phases) is given the recorded training time of the NF^{2}M actually selected for that system (Tab. 10). Tab. 17 gives all three quantities and Tab. 16 the resulting step counts. Matching time is not matching capacity or data: these backbones are 4.1M (single-head) and 6.1M (dual-head) parameters on the ALA systems against 18–87M for the NF^{2}M flows, so a flow-map step is much cheaper and a time-matched pipeline takes many more of them. Summed over the three phases, the baselines see 2.7–9.0× the number of passes over the training set that NF^{2}M's own 1000 epochs give.

Departures from a literal reading of the papers. Of the three, only Ai et al. (2026) released code, and the molecular code the F2D2/SCALLOP comparison is reproduced from isn't available. Where code and paper disagree, we follow the code; where a value is left unstated, we flag our choice as unswept. The departures below fall into three kinds: the released code overriding a paper's written equation, a stability or correctness fix we made and verified against a diagnostic (synthetic or molecular), and a value neither paper states at all.

Code overrides the papers' prose.

- **Self-distillation stop-gradient (F2D2, SCALLOP).** Both papers write the correction term with the whole derivative detached. The released code detaches only the divergence-head correction and lets the velocity term's derivative backpropagate. The literal, fully-detached version is unstable in our hands: on our synthetic diagnostic, the weight norm drifts unboundedly until a batch produces $|x_s| \sim 10^{10}$ and training diverges.
- **Time-pair sampling.** SCALLOP's prose restricts (s, t) to the upper triangle, but the F2D2 code samples both independently and uniformly over the full square and its own inverse sampler at evaluation time relies on having seen the reversed ordering during training. We sample the full square for all six objectives, so every pretraining checkpoint covers the same measure as the student that reads it.
- **Divergence head and target scale (F2D2).** SCALLOP's stated contribution is a vectorized divergence head (F2D2's released code has a scalar one). Vectorizing F2D2 looks at first like a free upgrade so we did it. Targets are then scaled by the reference's own head-shape rule ($\boldsymbol{\nabla} \cdot 1$ vectorized, $\boldsymbol{\nabla} \cdot d$ scalar) rather than the factor of 10 transcribed from the papers' 2D toy, which has no molecular support: their other published anchor is $\sim 10^4$ at $d = 12288$, four orders away, with the peptides ($d = 66$–414) in between.

Stability and correctness fixes. They were all obtained by doing ablations on synthetic or molecular targets.

- **Unsigned average-velocity head (FALCON).** The paper's $u_\theta = \mathrm{sign}(t - s)\, h_\theta$ forces h_θ to jump by $2v$ at the diagonal, so only the branch the tie-forcing sits on ever learns the correct side; we leave the head unsigned.
- **Stopped average-velocity derivative.** Left live, as in the reference, the residual becomes second-order in the network for both FALCON's L_{avg} and the ESD identity, since prediction and target come out of the same JVP call. We stop it.
- **Floored, projected SCALLOP divergence target.** $1/\sigma_t$ with $\sigma_t = t$ reaches 10^4 near the data end, so a single sample would dominate the batch; we floor it at 10^{-3}, at the cost of a small bias over the innermost decade of times, and state the target in the mean-free subspace rather than the ambient one.
- **FALCON-A's sub-jump weights.** The paper prints the two shares swapped relative to the identity's own derivative; they only agree at the paper's fixed $\lambda = 0.5$, which is why the error never surfaces in its results.
- **SCALLOP's loss-uncertainty weighting.** Implemented but left off: as specified, $\exp(-\psi)L + \psi$ is unbounded below wherever the (self-distillation) loss it weights can reach zero, stationary at $\psi = \log L$ with value $1 + \log L$. Clamping ψ bounds the weight but not the outcome; we found no stated fix and record this as an open parity gap rather than paper over it.

Left unstated by both papers.

- **Diagonal batch share.** 0.75 for FALCON/ESD (MeanFlow's (Geng et al., 2025) own ratio, the only published value for an average-velocity objective) and 0.5 for the Lagrangian family (the complement of F2D2's stated self-distillation split). Never swept.
- **Chignolin's flow-map backbone and NFE.** Neither paper reports chignolin; both are extrapolated with the same channel/block-doubling rule already used for the DiT presets (Tab. 11), giving the backbone in Tab. 12 and the step count in Tab. 20.

Table 12: Flow-map backbone architecture. The single-head network serves the ESD pretraining objective, FALCON and FALCON-A; the dual-head network serves LSD's students, F2D2 and SCALLOP, and counts its divergence-readout blocks only (the velocity reads straight off the shared trunk). Both carry a learned atom-type embedding, a bond-distance attention bias and centre-of-mass removal.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
Single-head (ESD, FALCON, FALCON-A)					
d_{model}	192	192	192	192	384
Blocks	6	6	6	6	12
Heads	6	6	6	6	12
Parameters			4.1M		32M
Dual-head (LSD, F2D2, SCALLOP)					
d_{model}	192	192	192	192	384
Shared blocks	6	6	6	6	12
Divergence-head blocks	3	3	3	3	6
Heads	6	6	6	6	12
Parameters			6.1M		48M
MLP ratio	4.0	4.0	4.0	4.0	4.0
Time-embedding dim	64	64	64	64	128

Table 13: Flow-map optimizer hyperparameters, per phase. Each pretraining objective takes the optimizer settings of the phase it initializes, so a warm start lands its student in a basin that student's own optimizer is already tuned for. All three columns use $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, an EMA of decay 0.999 and batch size 1024. "Flat-then-sqrt" holds the rate at its maximum for an absolute number of steps and then decays as $1/\sqrt{\text{step}}$; the onset is 10,000 steps on ALA-2 and ALA-4 and 20,000 elsewhere, read off the reference's molecular configurations. RAdam's variance rectification is why that schedule carries no warmup.

	FM	ESD / FALCON / FALCON-A	LSD / F2D2 / SCALLOP
Optimizer	AdamW	AdamW	RAdam
Learning rate	10^{-4}	5×10^{-4}	10^{-4}
Weight decay	10^{-4}	10^{-4}	0
Schedule	sqrt-anneal	cosine	flat-then-sqrt
Warmup	2.5%	5%	none
Grad-norm clip	10.0	10.0	10.0

B.2.2 Inference

NF^{2}M. Both NF^{2}M-IS and NF^{2}M-MIS sample a checkpoint from Tab. 10, using the time grid of (Karras et al., 2022) with $\rho = 7$, run in bf16 precision with $N = 10{,}000$ particles.

Shared initialization of multiple trajectories. At terminal noise, X_T and X_0 are nearly independent, so $p_{0|T} \approx p_0$: the first denoising kernel, from t_K to t_{K-1}, is essentially independent of the value of x_T. We exploit this when drawing N trajectories in parallel: rather than sampling N independent terminal states $x_T^n \sim \mathcal{N}(0, \sigma_T^2 I)$ and running N separate SNIS corrections (12) with M particles each, we draw a single x_T and build one SNIS approximation with NM candidates, which we then reuse by sampling from it N times. Since $p_{0|T} \approx p_0$ means the denoising posterior at this step is as hard to approximate as the target itself, pooling all NM candidates concentrates statistical power precisely where the conditional flow $q^\theta_{0|t}$ faces its hardest task. This sharing introduces a slight correlation between the N trajectories, which were otherwise independent; the correlation stays weak, however, precisely because the denoising kernel depends only weakly on x_T at this step.

Candidate budget. For NF^{2}M-IS, the candidate count M is solved per (system, variant, K) for the largest value whose measured wall-clock matches the corresponding unconditional baseline's own wall-clock; the selected checkpoints use the values in Tab. 18. For NF^{2}M-MIS, the architecture is fixed to the single variant with the best validation NF^{2}M-IS score at $K = 8$ per system (Tab. 10, MIS row); the values of M and the mixture size are given in Tab. 19.

Table 14: ESD / FALCON / FALCON-A loss hyperparameters, the single-head family. "Diagonal share" is the fraction of each batch drawn at $s = t$ and given to the instantaneous term, its complement carrying the self-consistency term; it is an exact-count split of every batch rather than a Bernoulli mask. FALCON-A is the exception at 1.0: forcing $s = t$ degenerates its additivity target to $0 = 0$ rather than to flow matching, so it keeps an explicit instantaneous term and evaluates both on every sample. Values marked † are not stated numerically by any of the three papers and were never swept. "–" marks a term the objective does not have.

	ESD	FALCON	FALCON-A
Self-consistency weight λ_{avg}	1.0†	1.0†	1.0†
Cycle-consistency weight λ_r	–	10.0	10.0
Split point λ	–	–	0.5
Diagonal share	0.75†	0.75†	1.0
Time pair		unordered square	
Time conditioning	pair	pair	pair
Average-velocity derivative	stopped	stopped	–

Table 15: LSD / F2D2 / SCALLOP loss hyperparameters, the dual-head family, which alone carries a divergence head. Same conventions as Tab. 14: † marks values neither paper states numerically, "–" marks a term the objective does not have.

	LSD	F2D2	SCALLOP
Self-consistency weight λ_{avg}	1.0†	1.0	1.0
Diagonal share	0.5	0.5	0.5
Time pair		unordered square	
Time conditioning	anchor + step	anchor + step	anchor + step
Average-velocity derivative	live	live	live
Divergence derivative	–	stopped	stopped
Divergence head	–	vectorized	vectorized
Divergence target scale	–	1.0	1.0
Divergence estimator	–	Hutchinson, 1 probe†	closed-form
CDM-v time weighting	–	–	linear
CDM-v denominator floor	–	–	10^{-3}

Refinement. At every level, each of the N particles is refined in-loop by 512 steps of MAExL at a per-system calibrated initial step size. After the last level, the returned samples are polished by a post-hoc MALA chain targeting the clean Boltzmann density $p(x_0) \propto \exp(-\mathcal{E}(x_0))$ (geometric step-size schedule, per-system calibrated initial step size τ) for 128 steps. Tab. 3 gives the initial step sizes used for MAExL and MALA.

Velocity teacher. The phase-1 velocity field is also evaluated on its own, as a flow-matching baseline in the style of ECNF++ (Tan et al., 2025), on the same protocol as everything else so its row sits beside theirs. Samples are the probability-flow ODE's push-forward of an ambient $\mathcal{N}(0, I)$ prior, integrated from the prior end down to the data end by adaptive Dormand–Prince at $\texttt{rtol} = \texttt{atol} = 10^{-4}$ with the data end inset at $t_{\min} = 10^{-5}$, where the linear schedule is singular. That comes to roughly 130–165 function evaluations per solve. The log-density is the instantaneous change of variables accumulated along the same trajectory, with the *exact* divergence: one JVP per input coordinate, $d = 66$–414. There is no probe count to tune. This run is fp32 rather than bf16: the log-determinant is an integral accumulated over $\sim$160 solver steps and is the whole point of the evaluation. We use $N = 10^4$ proposals per seed over 3 seeds then post-hoc MALA chain as every other method to match the energy evaluation budget (see Tab. 25). 10^4 rather than the 10^6 the flow-map and TarFlow cells use is what the exact likelihood affords .

FALCON, FALCON-A, F2D2 and SCALLOP. All four sample the phase-2 checkpoint at its EMA weights, at a fixed per-system number of flow-map steps (Tab. 20), in bf16 with the prior, the log-determinant, and the log-density accumulator held in fp32. FALCON and FALCON-A step on FALCON's own EDM-shaped grid ($\sigma_{\min} = 10^{-3}$, $\sigma_{\max} = 1$, $\rho = 7$, mapped affinely onto

Table 16: Training steps per system and phase, at batch size 1024. Phase 1 is one velocity teacher per system, shared by all four methods; phase 1.5 is one pretraining run per objective, shared by the two students that read it (ESD → FALCON, FALCON-A; LSD → F2D2, SCALLOP). Each entry is what the reported checkpoint was actually trained for.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
Phase 1 — velocity teacher					
Velocity	177,510	162,378	139,389	206,125	117,661
Phase 1.5 — flow-map pretraining					
ESD	130,768	226,158	208,892	362,830	293,385
LSD	68,937	70,578	55,023	77,465	270,018
Phase 2 — distillation					
FALCON	111,321	179,349	163,935	304,518	258,576
FALCON-A	166,981	171,378	147,863	215,861	161,439
F2D2	76,066	77,869	66,004	97,224	119,655
SCALLOP	109,927	104,984	91,525	137,399	189,516

Table 17: Measured per-step cost of every flow-map objective, and the two wall-clock quantities the step budget is solved against. Costs are seconds per optimizer step at batch size 1024 on one H100, each measured at the compilation mode its production run uses; chignolin is measured on its own 4-rank configuration at 256 rows per rank, so its column is directly comparable to the others per step.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
Seconds per step at batch 1024 (one H100)					
FM	0.0383	0.0380	0.0416	0.0581	0.1453
ESD	0.0332	0.0341	0.0361	0.0470	0.1040
LSD	0.0284	0.0320	0.0339	0.0472	0.1130
FALCON	0.0390	0.0430	0.0460	0.0560	0.1180
FALCON-A	0.0260	0.0450	0.0510	0.0790	0.1890
F2D2	0.0970	0.1020	0.1060	0.1230	0.2550
SCALLOP	0.0410	0.0470	0.0500	0.0690	0.1610

$[10^{-4}, 1-10^{-4}]$, which puts the fine steps at the data end); F2D2 and SCALLOP on the uniform grid in t that their paper states.

Density evaluation. The two families read their density differently, and the difference is what the evaluation costs. FALCON and FALCON-A have no divergence head, so $\log q$ is a sum of per-step Jacobian log-determinants, a dense $d \times d$ Jacobian per step, evaluated in fp32 whatever precision the network itself ran in, since a determinant amplifies rounding error with matrix size. F2D2 and SCALLOP read their divergence head and accumulate $\sum_i (s_i - t_i) \sum_j D_\theta(x_{t_i}, t_i, s_i)_j$.

Evaluation protocol. Evaluation is self-normalized importance sampling against the Boltzmann target, with 10^6 proposals per seed over 3 seeds, scored on the test split. The one exception is FALCON and FALCON-A on chignolin, where the exact per-step Jacobian costs $\sim$500 GPU-hours per seed at 10^6 proposals, and the pool is 10^5 instead. Importance weights are computed on the original proposals using their proposal densities and Boltzmann energies. We then resample 10^4 configurations according to the normalized weights and apply post-hoc MALA steps to each selected configuration (see Tab. 25 for the number of steps). The refined configurations retain equal weights, with no post-MALA importance-weight computation, and are compared against 10^4 reference configurations. A MALA kernel that leaves the Boltzmann target invariant can be applied before resampling while retaining the pre-move importance weights, or after resampling while retaining equal weights. Both constructions preserve the target in the importance-sampling limit, although their finite-sample dependence and variance can differ. The unconditional SNIS baseline follows the same order. No centre-of-mass correction is applied, unlike the TarFlow/StarFlow baselines: these models are mean-free end to end and no centre-of-mass augmentation enters their training, so their proposal density carries no centroid factor to correct.

Table 18: NF^{2}M-IS candidate budget M for the selected (variant, K) of Tab. 10.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
M	14	12	14	20	20

Table 19: NF^{2}M-MIS selected mixture configuration. C and M are the validation-selected mixture size and per-particle candidate count in every system.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
C	4	4	2	3	2
M	2	3	7	5	6

Batched evaluation and chunk sizes. Every sampler above draws and scores its pool in chunks rather than at once, and the chunk size is a throughput and memory setting that no reported number depends on. Tab. 21 collects all of them.

Comparing the distilled to their teacher. Tab. 22 compares SNIS performed directly against the velocity teacher with SNIS performed against each of its four distilled students, on the four ALA systems. The teacher itself is a strong sampler in its own right (competitive with the global comparison of Tab. 1) and on ALA-2 it beats all four flow maps outright. On the larger systems the picture is mixed rather than uniformly closing: it is specifically the single-head family, FALCON and FALCON-A, that occasionally overtakes the teacher, while the dual-head family essentially never does. Since the teacher is already a competitive sampler, these occasional wins are not evidence that it was undertrained.

The inference-time column looks paradoxical: the flow maps draw $100\times$ more particles than the teacher (10^6 against 10^4 as 10^6 was unfeasible with the velocity teacher) while taking roughly $10\times$ fewer steps (4–16 NFE against $\sim$130–165, Tab. 20). For FALCON and FALCON-A these two factors largely cancel: their dense $d \times d$ Jacobian log-determinant, paid at every step, erases the step-count saving, leaving wall-clock inference never faster than the teacher and up to $6.3\times$ slower on ALA-6. F2D2 and SCALLOP avoid that cost by reading a divergence head instead, so the step-count saving survives the extra particles: both are consistently 19–22$\times$ faster than the teacher across all four systems.

Cost of the Langevin refinements. Every sampler reported here spends part of its inference budget on Langevin dynamics, and at a matched target-energy budget the two kernels enter at the same scale. NF^{2}M pays both: 512 MAExL steps at every level, over the full particle set, plus 128 steps of post-hoc MALA on its (comparatively small) returned sample. Every other method pays only the post-hoc chain, but at the step count that equalises the budget (Tab. 25). Tab. 23 gives the measured wall-clock and its share of each sampler's inference time.

Skipping post-hoc MALA steps. Tab. 24 reports every sampler at matched budget, both with its post-hoc MALA ladder (128 steps for NF^{2}M, Tab. 25 for the others) and on its raw output. With refinement, NF^{2}M is best in energy on every system except ALA-6, and best in torsion on ALA-6 and chignolin. Without it, NF^{2}M is best in energy on every system and in torsion on the three largest. Its own numbers barely depend on the ladder: torsion is unchanged to within 0.001 and energy moves by small amounts in both directions. The baselines behave in the opposite way. Most of their energy accuracy comes from the ladder, their raw distances are up to an order of magnitude worse on the larger systems, and for several of them the raw seed standard deviation is comparable to the mean. FALCON's refined energy win on ALA-6 in particular does not survive: its raw distance is more than twice that of NF^{2}M-IS, even though FALCON runs over a hundred times longer. In torsion space, the ladder improves the baselines by at most about 15% and leaves their order unchanged, so the shape of that comparison is a property of the proposals themselves.

Matching the target-energy budget. Since MAExL and the post-hoc ladder both query the target, wall-clock alone does not settle whether NF^{2}M simply spends more energy evaluations than the

Table 20: Flow-map inference steps (NFEs), shared by all four methods. The four ALA values are the ones both papers report; chignolin appears in neither and reuses the largest.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
NFE	4	8	8	16	16

Table 21: Evaluation chunk sizes, in rows, on an A100-80GB. The SNIS and NF^{2}M architectures are selected per system based on validation metrics. NF^{2}M variant labels (T1, W1, S1, S1k, S2) are defined in Tab. 9.

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
SNIS: Unconditional TarFlow/StarFlow					
Selected flow	TarFlow	TarFlow	StarFlow	StarFlow	TarFlow
Sample	131072	131072	65536	32768	16384
Log-prob	131072	131072	32768	32768	32768
NF^{2}M (measured knee, bf16)					
Selected variant (IS / MIS)	W1	W1	T1	S2 / S1k	S1
Sample	32768	65536	65536	16384	4096
Log-prob	4096	8192	8192	4096	256
NF^{2}M-IS chunk	32768	65536	65536	16384	4096
NF^{2}M-MIS chunk	32768	65536	65536	16384	1024
Flow maps (sample with log-prob, bf16)					
FALCON / FALCON-A	4096	1024	1024	256	16
F2D2 / SCALLOP	8192	8192	8192	8192	8192
Velocity teacher (exact divergence, fp32)					
Sample with log-prob	64	48	32	32	–

baselines. We therefore equalize the count exactly at three operating points, one per MAExL setting of Tab. 4, re-polishing every SNIS-based sampler from its unrefined samples with the ladder length that spends the same number of target-energy calls, counting a score as two. Tab. 25 gives those lengths, and Figs. 8 to 10 trace every method across the three budgets. The leftmost point handicaps NF^{2}M, since matching it means running with no MAExL at all, precisely the setting the ablation identifies as its weakest, while the baselines lose little from a shortened chain. From the middle budget onward NF^{2}M recovers its Tab. 1 numbers and with them the ranking of the main table, whereas the extra calls granted to the baselines buy energy accuracy only: their torsion curves are close to flat and stay well behind on the larger systems, since a local kernel cannot redistribute mass between torsional metastable states however long it runs. The one place the verdict shifts is chignolin energy at the largest budget, where several baselines become indistinguishable from NF^{2}M in the mean, though their across-seed spread remains far wider. Matching energy calls also does not equalize wall-clock, and the gap widens with the budget, since a longer ladder is pure serial cost for the baselines while NF^{2}M pays its share inside the sampling loop. The conclusion of the main table is therefore unchanged: the separation lies in the proposal rather than in the refinement budget.

Comparison with SBG. Tab. 26 provides a full comparison against SBG-SMC on both proposal architectures. The two methods trade places depending on the regime. On the smallest peptides, the geometric annealing path is clearly beneficial in torsion space, where SBG-SMC is the strongest baseline, while NF^{2}M already dominates in energy space. This ordering reverses as the dimension grows: on the largest systems NF^{2}M improves on the best SBG-SMC variant on both metrics, and the gap is widest on chignolin, where the SMC ladder degrades sharply. Two further properties separate the methods. First, SBG-SMC is markedly less stable, both across seeds, with several runs whose standard deviation is of the order of their mean, whereas NF^{2}M varies little across seeds and across its two importance-sampling variants. Second, at an equal proposal budget the annealing ladder roughly doubles inference time on the ALA systems and costs several times more on chignolin, for comparable training cost. Finally, since SNIS and SBG-SMC use the same unconditional checkpoints and differ only at inference, their comparison isolates the effect of the annealing path itself. Although the two use different proposal budgets for a comparable wall-clock, the picture matches the ablation

Table 22: Flow-map samplers against the velocity field they are distilled from, on ALA-N systems with $N = 2, 3, 4, 6$. Results are obtained from three random seeds, reported as `mean` $\pm$ `std`. Energy and torus Wasserstein distances are computed on the test split against 10^4 reference samples.

		Velocity	FALCON	FALCON-A	F2D2	SCALLOP
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.765_{\pm 0.097}$	$0.787_{\pm 0.091}$	$0.966_{\pm 0.285}$	$1.132_{\pm 0.018}$	$1.110_{\pm 0.014}$
ALA-2	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$0.274_{\pm 0.032}$	$0.524_{\pm 0.212}$	$0.892_{\pm 0.932}$	$0.774_{\pm 0.059}$	$0.391_{\pm 0.029}$
($d = 66$)	Inference (s) $\downarrow$	$1375.3_{\pm 10.4}$	$1477.6_{\pm 2.0}$	$1461.3_{\pm 11.9}$	$70.6_{\pm 0.1}$	$71.5_{\pm 0.6}$
	Training (h) $\downarrow$	1.9	4.8	5.0	6.3	5.6
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.410_{\pm 0.219}$	$0.250_{\pm 0.031}$	$0.445_{\pm 0.410}$	$0.648_{\pm 0.211}$	$0.596_{\pm 0.448}$
ALA-3	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$0.677_{\pm 0.168}$	$0.468_{\pm 0.144}$	$1.102_{\pm 0.540}$	$1.072_{\pm 0.370}$	$0.946_{\pm 0.117}$
($d = 99$)	Inference (s) $\downarrow$	$2956.8_{\pm 3.0}$	$7832.2_{\pm 7.4}$	$7813.1_{\pm 5.2}$	$158.1_{\pm 0.1}$	$158.9_{\pm 0.9}$
	Training (h) $\downarrow$	1.7	7.1	7.2	6.5	5.8
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$1.410_{\pm 0.380}$	$1.556_{\pm 0.164}$	$2.375_{\pm 1.573}$	$3.477_{\pm 2.242}$	$3.306_{\pm 1.521}$
ALA-4	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$2.133_{\pm 0.370}$	$0.948_{\pm 0.125}$	$1.532_{\pm 0.929}$	$2.564_{\pm 0.060}$	$2.490_{\pm 0.142}$
($d = 126$)	Inference (s) $\downarrow$	$4299.4_{\pm 7.3}$	$12172.3_{\pm 0.2}$	$12155.4_{\pm 11.9}$	$191.4_{\pm 0.1}$	$192.5_{\pm 0.7}$
	Training (h) $\downarrow$	1.6	7.1	7.1	5.6	5.1
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$2.339_{\pm 1.352}$	$0.652_{\pm 0.137}$	$0.980_{\pm 0.488}$	$2.749_{\pm 2.219}$	$2.216_{\pm 0.248}$
ALA-6	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$2.696_{\pm 0.557}$	$1.829_{\pm 0.291}$	$1.659_{\pm 0.008}$	$3.111_{\pm 0.261}$	$3.300_{\pm 0.410}$
($d = 189$)	Inference (s) $\downarrow$	$9489.6_{\pm 73.7}$	$59606.9_{\pm 40.4}$	$59610.4_{\pm 50.4}$	$492.3_{\pm 0.6}$	$484.4_{\pm 13.2}$
	Training (h) $\downarrow$	3.3	15.1	15.1	10.3	9.8

Table 23: Langevin refinement cost at a matched target-energy budget: 512 MAExL steps for NF^{2}M and the budget-matched post-hoc MALA ladder for all other methods. Each entry gives the wall-clock time spent in the Langevin kernels and its share of the sampler's total inference time (means over 3 seeds, A100-80GB).

	ALA-2	ALA-3	ALA-4	ALA-6	Chignolin
In-loop MAExL (512 steps) and post-hoc MALA (128 steps)					
NF^{2}M-IS	6 (4.1%)	14 (8.6%)	17 (9.4%)	23 (5.0%)	108 (3.8%)
NF^{2}M-MIS	9 (7.0%)	14 (9.5%)	17 (10.7%)	36 (8.8%)	112 (5.0%)
Budget-matched post-hoc MALA only (Tab. 25)					
TarFlow/StarFlow	8 (7.2%)	14 (10.6%)	17 (10.1%)	35 (9.3%)	166 (6.8%)
Velocity	435 (24.2%)	322 (9.8%)	470 (9.9%)	499 (5.0%)	–
FALCON	8 (0.6%)	14 (0.2%)	17 (0.1%)	35 (0.1%)	168 (0.1%)
FALCON-A	8 (0.6%)	13 (0.2%)	17 (0.1%)	35 (0.1%)	168 (0.1%)
F2D2	8 (18.5%)	13 (11.5%)	17 (12.0%)	35 (8.6%)	167 (3.8%)
SCALLOP	8 (18.3%)	13 (11.5%)	17 (11.9%)	35 (8.6%)	167 (3.8%)

of Tan et al. (2025), the ladder helping on some systems and hurting on others rather than yielding a consistent gain.

Table 24: Results on ALA-N systems, with $N = 2, 3, 4, 6$, and on chignolin, with and without post-hoc MALA refinement, at a matched target-energy budget Tab. 25. Results are obtained from three random seeds, reported as `mean` $\pm$ `std`. Energy and torus Wasserstein distances are computed on the test split against 10^4 reference samples. The proposal budget differs by method: 10^4 samples for NF^{2}M, 10^5 for FALCON and FALCON-A on chignolin, and 10^6 for every other column. Inference time covers sampling, plus the post-hoc ladder for the MALA rows. Best in **bold**, second best underlined, within each row.

		SNIS	NF^{2}M-IS	NF^{2}M-MIS	FALCON	FALCON-A	F2D2	SCALLOP
ALA-2 ($d = 66$)	$\mathcal{E}$-$\mathcal{W}_2$ MALA ↓	0.77±0.03	**0.13**±0.01	0.24±0.08	0.76±0.05	0.60±0.33	0.89±0.03	0.87±0.03
	$\mathcal{E}$-$\mathcal{W}_2$ raw ↓	1.07±0.15	**0.20**±0.05	0.24±0.07	0.64±0.07	1.50±1.05	2.84±0.42	2.77±0.36
	$\mathbb{T}$-$\mathcal{W}_2$ MALA ↓	**0.24**±0.01	1.05±0.01	1.24±0.01	0.52±0.25	0.86±1.09	0.75±0.07	0.37±0.04
	$\mathbb{T}$-$\mathcal{W}_2$ raw ↓	**0.25**±0.01	1.05±0.01	1.24±0.01	0.52±0.20	0.92±0.97	0.78±0.06	0.40±0.03
	Inference MALA (s) ↓	117±0	137±15	126±14	1453±2	1448±5	46±0	46±0
	Inference raw (s) ↓	109±0	137±15	126±14	1444±2	1440±4	37±0	38±0
ALA-3 ($d = 99$)	$\mathcal{E}$-$\mathcal{W}_2$ MALA ↓	0.27±0.07	0.16±0.01	**0.16**±0.01	0.18±0.05	0.24±0.14	0.55±0.13	0.70±0.19
	$\mathcal{E}$-$\mathcal{W}_2$ raw ↓	0.30±0.07	0.17±0.03	**0.14**±0.02	0.65±0.21	3.87±2.64	2.30±0.92	1.73±0.53
	$\mathbb{T}$-$\mathcal{W}_2$ MALA ↓	**0.36**±0.02	0.49±0.02	0.47±0.01	0.46±0.17	1.10±0.71	1.05±0.48	0.85±0.20
	$\mathbb{T}$-$\mathcal{W}_2$ raw ↓	**0.38**±0.01	0.49±0.02	0.47±0.01	0.48±0.15	1.13±0.55	1.12±0.38	1.00±0.10
	Inference MALA (s) ↓	128±0	164±0	147±13	7792±9	7789±6	117±0	117±0
	Inference raw (s) ↓	115±0	164±0	147±13	7778±7	7775±5	104±0	104±0
ALA-4 ($d = 126$)	$\mathcal{E}$-$\mathcal{W}_2$ MALA ↓	1.52±0.35	**0.99**±0.03	1.11±0.01	1.76±0.19	1.99±0.74	2.75±2.10	2.98±0.97
	$\mathcal{E}$-$\mathcal{W}_2$ raw ↓	2.25±1.26	**0.95**±0.14	1.00±0.06	2.09±0.33	4.40±4.38	6.37±1.22	5.36±1.95
	$\mathbb{T}$-$\mathcal{W}_2$ MALA ↓	**0.78**±0.13	0.82±0.04	0.88±0.05	0.89±0.14	1.46±1.09	2.43±0.12	2.37±0.19
	$\mathbb{T}$-$\mathcal{W}_2$ raw ↓	0.86±0.13	**0.83**±0.04	0.88±0.05	1.00±0.13	1.59±0.96	2.66±0.04	2.59±0.13
	Inference MALA (s) ↓	167±5	184±14	162±14	12124±1	12118±5	143±0	143±0
	Inference raw (s) ↓	150±5	184±14	162±14	12107±0	12101±4	126±0	126±0
ALA-6 ($d = 189$)	$\mathcal{E}$-$\mathcal{W}_2$ MALA ↓	0.92±1.02	1.02±0.06	1.30±0.09	**0.61**±0.41	0.68±0.49	2.16±1.72	3.77±1.72
	$\mathcal{E}$-$\mathcal{W}_2$ raw ↓	2.66±1.97	**1.05**±0.13	1.36±0.09	2.43±0.88	1.70±0.23	13.33±1.61	11.58±1.46
	$\mathbb{T}$-$\mathcal{W}_2$ MALA ↓	1.77±0.68	**0.99**±0.03	1.03±0.02	1.72±0.33	1.56±0.03	2.93±0.29	3.13±0.46
	$\mathbb{T}$-$\mathcal{W}_2$ raw ↓	1.95±0.74	**0.99**±0.03	1.03±0.02	1.91±0.29	1.74±0.01	3.24±0.26	3.44±0.43
	Inference MALA (s) ↓	373±0	453±14	406±14	59502±50	59528±58	405±1	402±5
	Inference raw (s) ↓	338±0	452±14	405±14	59467±40	59493±47	370±1	367±4
Chignolin ($d = 414$)	$\mathcal{E}$-$\mathcal{W}_2$ MALA ↓	7.75±5.29	**4.33**±0.26	6.11±0.13	4.81±1.07	5.03±3.62	4.79±2.34	4.77±1.08
	$\mathcal{E}$-$\mathcal{W}_2$ raw ↓	27.76±19.91	**4.05**±0.27	6.11±0.23	8.69±2.25	12.00±3.43	26.12±11.05	46.48±16.13
	$\mathbb{T}$-$\mathcal{W}_2$ MALA ↓	3.86±0.17	2.60±0.05	**2.58**±0.02	2.99±0.09	3.45±0.62	3.93±0.45	3.57±0.20
	$\mathbb{T}$-$\mathcal{W}_2$ raw ↓	4.22±0.14	2.60±0.05	**2.58**±0.02	3.33±0.08	3.76±0.53	4.29±0.37	3.94±0.22
	Inference MALA (s) ↓	2443±2	2831±17	2265±1	169239±171	169131±151	4380±4	4384±1
	Inference raw (s) ↓	2277±2	2826±17	2259±1	169071±140	168963±124	4214±3	4218±1

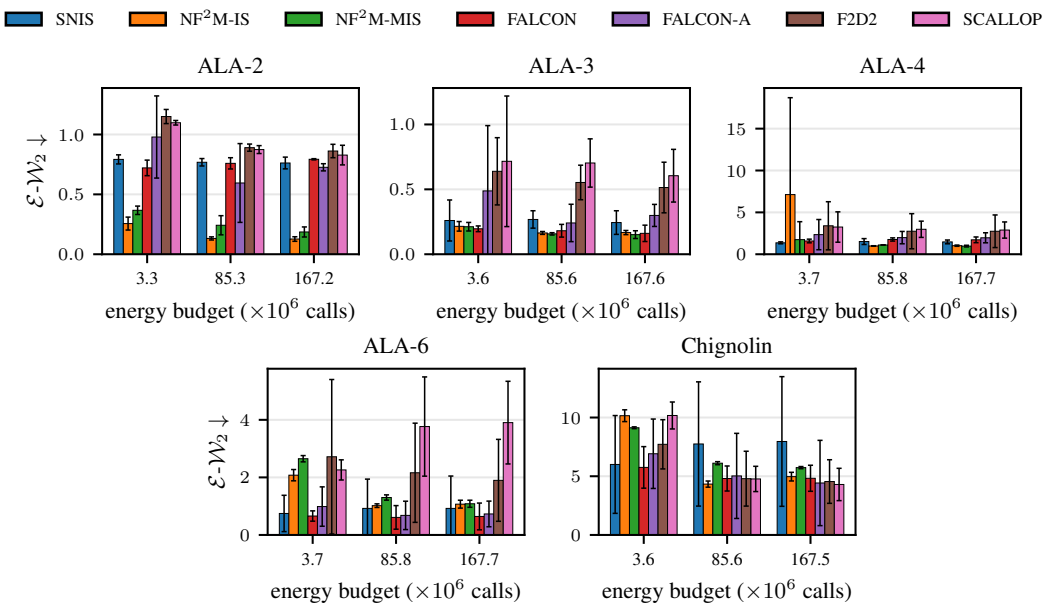


Figure 8: Energy Wasserstein distance against the matched target-energy budget, on the four ALA systems and chignolin. Every point on a panel spends the same number of target-energy calls. The three budgets are the 0-, 512- and 1024-MAExL-step settings of Tab. 4 for NF^{2}M, and the equalizing post-hoc MALA ladders of Tab. 25 for the SNIS-based samplers. Means over three seeds.

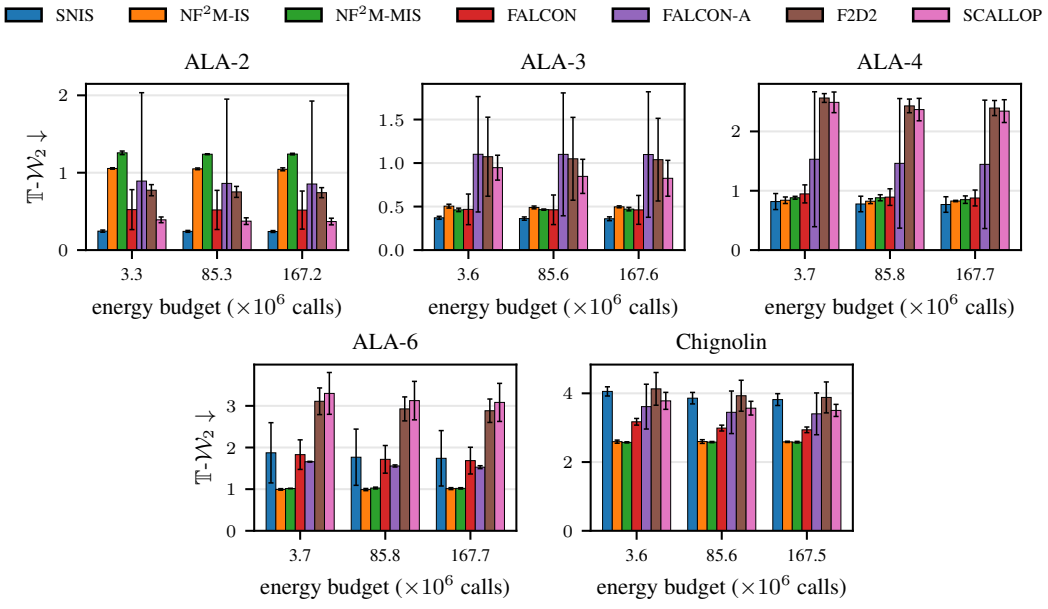


Figure 9: Torus Wasserstein distance against the matched target-energy budget, same protocol and seeds as Fig. 8.

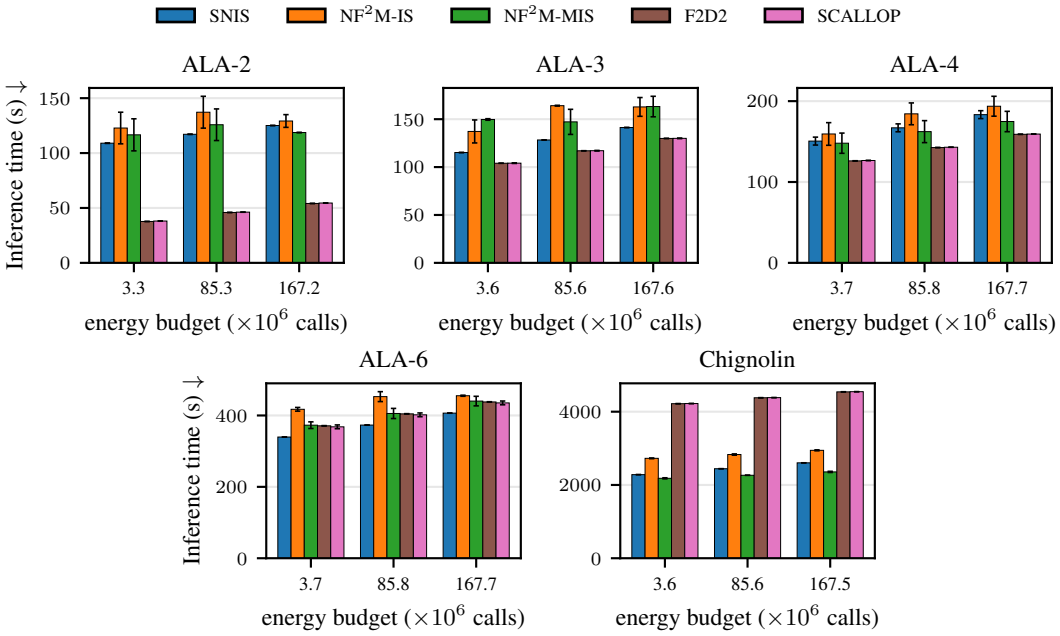


Figure 10: Inference wall-clock against the matched target-energy budget, same protocol and seeds as Fig. 8. FALCON and FALCON-A are omitted, their per-step Jacobian log-determinant putting them off the scale of every panel.

Table 25: Number of post-hoc MALA steps run for the SNIS-based samplers at each matched budget. Each step count is chosen so that the SNIS-based total, P energy calls for the importance weights plus $n\,(S+1)$ score calls on $n = 10^4$ chains with one score call counted as two energy calls, matches the budget of the corresponding NF^{2}M setting. All SNIS-based samplers use $P = 10^6$ proposals, except FALCON and FALCON-A on chignolin, which use $P = 10^5$ and therefore take longer ladders for the same total.

		MAExL steps		
System	**Methods**	0	512	1024
ALA-2 ($d = 66$)	SNIS, FALCON, FALCON-A, F2D2, SCALLOP	114	4212	8308
ALA-3 ($d = 99$)		127	4231	8327
ALA-4 ($d = 126$)		135	4239	8335
ALA-6 ($d = 189$)		134	4238	8334
Chignolin ($d = 414$)	SNIS, F2D2, SCALLOP	129	4228	8324
	FALCON, FALCON-A	174	4273	8369
ALA-2 ($d = 66$)	Velocity teacher	163	4261	8357
ALA-3 ($d = 99$)		176	4280	8376
ALA-4 ($d = 126$)		184	4288	8384
ALA-6 ($d = 189$)		183	4287	8383

Table 26: Results on ALA-N systems, with $N = 2, 3, 4, 6$, and on chignolin. Results are obtained from three random seeds, reported as `mean ± std`. Energy and torus Wasserstein distances are computed on the test split against 10^4 reference samples. The proposal budget differs by method: 10^4 samples for NF^{2}M and SBG-SMC, and 10^6 for SNIS. The SNIS column reports, per system, the better of the TarFlow and StarFlow checkpoints; the SBG-SMC columns are reported per architecture. Inference time covers sampling and the post-hoc MALA ladder; training time is the wall-clock of the corresponding model's training run. Best in **bold**, second best underlined.

		SNIS	NF^{2}M-IS	NF^{2}M-MIS	SBG-SMC StarFlow	SBG-SMC TarFlow
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.752_{\pm 0.027}$	$\mathbf{0.132}_{\pm 0.013}$	$\underline{0.242}_{\pm 0.080}$	$0.737_{\pm 0.142}$	$0.686_{\pm 0.042}$
ALA-2	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathbf{0.245}_{\pm 0.014}$	$1.050_{\pm 0.011}$	$1.239_{\pm 0.006}$	$0.387_{\pm 0.075}$	$\underline{0.385}_{\pm 0.021}$
($d = 66$)	Inference (s) $\downarrow$	$128.8_{\pm 0.4}$	$137.2_{\pm 14.5}$	$125.8_{\pm 14.4}$	$242.1_{\pm 9.7}$	$263.2_{\pm 11.4}$
	Training (h) $\downarrow$	3.4	4.3	4.3	3.6	3.4
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.225_{\pm 0.047}$	$\underline{0.164}_{\pm 0.011}$	$\mathbf{0.157}_{\pm 0.009}$	$2.715_{\pm 3.857}$	$0.524_{\pm 0.072}$
ALA-3	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathbf{0.371}_{\pm 0.016}$	$0.490_{\pm 0.016}$	$\underline{0.465}_{\pm 0.007}$	$1.256_{\pm 1.256}$	$0.470_{\pm 0.044}$
($d = 99$)	Inference (s) $\downarrow$	$163.1_{\pm 0.3}$	$164.1_{\pm 0.4}$	$147.2_{\pm 13.1}$	$282.5_{\pm 9.6}$	$288.4_{\pm 9.5}$
	Training (h) $\downarrow$	4.7	6.0	6.0	5.3	4.7
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$1.313_{\pm 0.159}$	$\mathbf{0.988}_{\pm 0.027}$	$\underline{1.107}_{\pm 0.013}$	$1.927_{\pm 0.096}$	$1.734_{\pm 0.138}$
ALA-4	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\underline{0.819}_{\pm 0.134}$	$0.824_{\pm 0.041}$	$0.882_{\pm 0.051}$	$0.896_{\pm 0.035}$	$\mathbf{0.789}_{\pm 0.093}$
($d = 126$)	Inference (s) $\downarrow$	$206.0_{\pm 7.3}$	$184.3_{\pm 13.5}$	$162.3_{\pm 13.6}$	$337.2_{\pm 0.3}$	$357.7_{\pm 10.5}$
	Training (h) $\downarrow$	5.8	5.3	5.3	5.8	5.2
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$0.714_{\pm 0.584}$	$1.018_{\pm 0.063}$	$1.301_{\pm 0.092}$	$\mathbf{0.249}_{\pm 0.116}$	$\underline{0.395}_{\pm 0.182}$
ALA-6	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$1.874_{\pm 0.724}$	$\mathbf{0.990}_{\pm 0.026}$	$\underline{1.027}_{\pm 0.021}$	$1.355_{\pm 0.061}$	$1.797_{\pm 0.068}$
($d = 189$)	Inference (s) $\downarrow$	$460.9_{\pm 0.3}$	$452.8_{\pm 13.9}$	$405.6_{\pm 14.2}$	$1014.0_{\pm 11.9}$	$975.2_{\pm 11.2}$
	Training (h) $\downarrow$	11.7	12.8	10.2	11.7	10.7
	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\underline{5.898}_{\pm 3.986}$	$\mathbf{4.331}_{\pm 0.262}$	$6.112_{\pm 0.129}$	$8.635_{\pm 4.105}$	$14.439_{\pm 6.417}$
Chignolin	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$4.056_{\pm 0.134}$	$\underline{2.597}_{\pm 0.054}$	$\mathbf{2.581}_{\pm 0.018}$	$3.933_{\pm 0.749}$	$4.302_{\pm 0.392}$
($d = 414$)	Inference (s) $\downarrow$	$2792.2_{\pm 3.5}$	$2830.6_{\pm 17.4}$	$2264.5_{\pm 1.1}$	$26337.6_{\pm 175.1}$	$11875.9_{\pm 178.6}$
	Training (h) $\downarrow$	13.3	21.7	21.7	17.4	13.3