NORMALIZING FLOW MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalable approaches to learning neural dynamical systems have placed diffusion, flow matching, and few-step generators in flow maps at the forefront of generative modeling. Such state-of-the-art models have limited access to explicit model densities despite evolving from classical generative modeling approaches that leveraged tractable likelihoods, e.g., Normalizing Flows and Autoregressive models. In this paper, we combine these threads of few-step generation and tractable sample likelihoods and introduce NORMALIZING FLOW MAPS (NFM), a class of *exactly invertible pushforward maps* that generalizes normalizing flows and flow maps. More precisely, NFMs learn the integrator of an underlying dynamical system, but crucially, they provide efficient exact likelihood evaluation by construction. To enable scalable training of NFMs, we derive two complementary axes of consistency inherent to NFMs that yield (self) distillation objectives: (1) pointwise matching, which unlocks ℓ_2-based flow-map style regression losses, and (2) distributional matching, which matches pushforward distributions to the marginals of a prescribed forward process. On conditional image generation on CIFAR-10 and ImageNet-64, NFMs achieve state-of-the-art sample quality among parameter-matched normalizing flows, while supporting likelihood-based RLHF. For equilibrium sampling in Boltzmann Generators, NFMs outperform prior state-of-the-art normalizing flows on both single-system and transferable peptide benchmarks.

1 INTRODUCTION

Learning a transport map between a tractable reference distribution and the empirical distribution of complex data has become the de facto paradigm in generative modeling, with varied application domains spanning audiovisual media (Ho et al., 2022; Liu et al., 2023a), biomolecular modeling (Abramson et al., 2024; Watson et al., 2023), and scientific simulation (Shu et al., 2023; Du et al., 2024). Driving this widespread adoption is the ability to leverage standard alignment techniques such as best-of-N selection (Ramesh et al., 2021), inference-time steering (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Bansal et al., 2023; Ma et al., 2025), and post-training algorithms (Black et al., 2024; Liu et al., 2025a) which effectively sharpen the model distribution—by increasing sample rewards assessed by an oracle. However, each generated sample necessitates computationally expensive numerical integration of an ODE or SDE—producing a substantial inference tax that computationally challenges the steering and post-training pipeline (Wohlwend et al., 2025; Scarpellini et al., 2026).

Indeed, while typical Autoregressive LLMs have a mature post-training stack—e.g., PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), etc.—corresponding algorithmic frameworks for diffusion and flow model post-training are both vast and varied (Wallace et al., 2024; Prabhudesai et al., 2023; Clark et al., 2024; Xu et al., 2023). The crucial distinction is that LLMs natively provide tractable sample likelihoods, whereas evaluating exact densities for diffusion and flow-matching models is computationally challenging. Specifically, evaluating sample likelihoods of flow models requires integrating the instantaneous change of variables along the underlying ODE, computing a high-dimensional network Jacobian, and necessitates stochastic trace estimation (Chen et al., 2018; Meyer et al., 2021). Instead, it is computationally more attractive to target the reward-tilted distribution, by modifying the drift of the underlying dynamics through path-space objectives. These objectives, however, crucially rely on high-variance value estimates of the terminal reward. More precisely, value function estimation necessitates propagating terminal rewards through noisy intermediate states, which requires backpropagating through time (Ben-Hamu et al., 2024)–or a noisy single-step denoising approximation (Blasingame & Liu, 2025; Lee et al., 2026b). These algorithmic features combine and give rise to tilting objectives that are inherently high variance, often leading to unstable post-training and mode collapse.

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

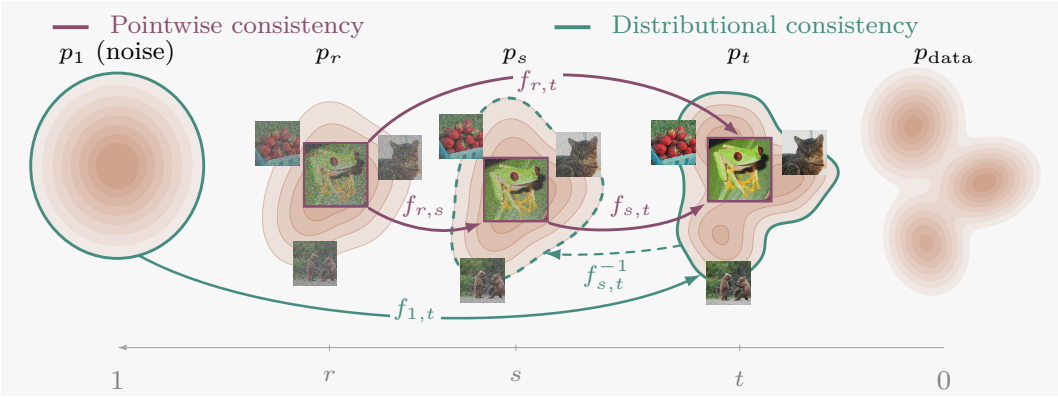


Figure 1: Overview of NFM, which performs generative transport along a stochastic interpolant. NFM-R exploits pointwise matching at the particle level, while NFM-D leverages distributional matching.

As generative models are deployed at scale, the cumulative cost of repeated sampling for generation, steering, and post-training makes inference efficiency a central design consideration. This practical motivation has led to the emergence of flow maps (Boffi et al., 2025a; Geng et al., 2025; Boffi et al., 2025b), that directly learn *the flow integrator*, effectively compressing numerical simulation into few network evaluations. Although the exact ODE flow is a diffeomorphism, standard parameterizations are not invertible, nor do they provide tractable Jacobian determinants. While flow maps reduce the inference tax, they do not resolve the crucial likelihood-access gap—limiting their usage in scientific applications like Boltzmann Generators (Noé et al., 2019). As a result, classical Normalizing Flows (NF) (Tabak & Vanden-Eijnden, 2010; Tabak & Turner, 2013; Dinh et al., 2017; Rezende & Mohamed, 2015) trained with MLE remain an attractive exact likelihood-based model class. However, their practical deployment critically relies on noise injection into the clean data distribution for stable training, which unfortunately eliminates the promised access to cheap sample likelihood (Chen et al., 2026). Taken together, these considerations motivate a new generative model that combines few-step sampling of flow maps with exact sample likelihoods for scalable, likelihood-based post-training.

Current work. In this paper, we introduce NORMALIZING FLOW MAPS (NFM), a new generative model family, depicted in fig. 1, that is purpose-built to deliver two objectives of interest: 1.) *Exact sample likelihoods* and 2.) *Few-step inference*; thereby generalizing classical invertible normalizing flows and endowing flow maps with tractable likelihoods. An NFM defines a time-conditioned generative transport operator that is tied to the law of a known stochastic interpolant. However, unlike existing model classes, NFM by construction defines a family of tractable invertible maps connecting arbitrary points along a prescribed probability path. Consequently, NFMs provide exact sample likelihoods *without* expensive numerical integration. Importantly, restricting this family to the prior and data endpoints recovers a normalizing flow, and querying arbitrary time pairs yields the few-step transport characteristics of a flow map. As a result, NFMs are naturally suited for applications that require sample likelihoods, such as Boltzmann generators (Noé et al., 2019) and RLHF post-training.

To train NFMs, we operationalize two complementary matching principles inherent to NFMs, which in turn lead to self-distillation training objectives. In particular, our key insight is that NFMs must satisfy a known stochastic interpolant at a *pointwise* level and/or match its *marginal distribution*. The former condition unlocks regression-based training common in flow maps, but adapted to NFM, while the latter is an altogether new algorithmic framework that trains few-step generators by exploiting NFMs family of (end-point) invertible maps. We summarize our main contributions:

1. **Model class**. We introduce NFM, a new generative model class, that learns an invertible pushforward map, while natively supporting few-step sampling and exact sample likelihoods.
2. **Training algorithms**. Defined by matching principles (§3.2), we design two instantiations in NFM-R and NFM-D, which leverages pointwise and distributional matching respectively. In particular, NFM-R optimizes ℓ_2-regression objectives. Meanwhile, NFM-D matches the induced pushforward distributions along the marginal law of the known forward process.
3. **Empirical analysis**. We evaluate NFM across image generation, post-training, and equilibrium sampling. On CIFAR-10 and ImageNet-64, NFM improves over MLE-trained NFs in FID and FLD while for BGs NFM improves over prior continuous flow approaches. We also demonstrate that exact tractable likelihoods of NFM enable direct post-training unlike diffusion models.

2 Background and Preliminaries

Given a dataset of samples, $\mathcal{D} = \{x^k\}_{k=1}^n$, from a data distribution $p_{\text{data}} \in \mathbb{P}(\mathbb{R}^d)$, and a tractable base distribution $p_{\text{base}} \in \mathbb{P}(\mathbb{R}^d)$, generative models based on dynamical systems parametrize a map $f^\theta : \mathbb{R}^d \mapsto \mathbb{R}^d$—with learnable parameters θ. The induced pushforward distribution is thus $p^\theta := [f^\theta]_\#(p_{\text{base}})$, and the goal of learning is then to approximate $p^\theta(x) \approx p_{\text{data}}(x)$ in a distributional sense. We use superscripts to designate indices, e.g., $k \in [n]$, and reserve subscripts to indicate a notion of time $t \in [0, 1]$ associated with the mass transport problem. Finally, we set $p_1 := p_{\text{base}} = \mathcal{N}(0, I)$ to be the base distribution, $p_0 := p_{\text{data}}$ the data distribution and $p^\theta = [f^\theta]_\#(p_1)$.

2.1 Stochastic Interpolants and Flow Maps

Given $\mathbb{P}(\mathbb{R}^d)$ the space of probability distributions on $\mathbb{R}^d$, a *stochastic interpolant* (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025) $I : [0, 1] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a stochastic process that mixes samples from the base and target measures at the particle level $(t, x_0, x_1) \mapsto I_t(x_0, x_1) := x_t = \alpha_t x_0 + \beta_t x_1$, for a choice of smooth functions α and β, with boundary constraints $\alpha_0 = 1$, $\alpha_1 = 0$, $\beta_0 = 0$ and $\beta_1 = 1$. Given a coupling, π, of base and target measures $(x_0, x_1) \sim \pi(x_0, x_1)$, the stochastic interpolant defines a probability path that connects the measures p_0 and p_1 in time

$$p_t = \text{Law}(I_t), \quad \text{such that } \int \pi(x_0, x_1) dx_0 = p_1(x_1) \int \pi(x_0, x_1) dx_1 = p_0(x_0). \tag{1}$$

The marginal law in eq. (1) defines a *probability-flow* in time $t \in [0, 1]$ that pushes samples from p_0 to p_1 via the velocity field $v_t(x) = \mathbb{E}_{x_t \sim p_t(x_t)}[\partial I_t / \partial t \mid I_t = x_t]$, which averages over all interpolants. Unfortunately, this marginal velocity is computationally inaccessible. Instead, the conditional probability flow $p_t(\cdot|x_0, x_1)$ indexed for all $t \in (0, 1)$ can be leveraged to train a neural ODE (Chen et al., 2018). Specifically, setting $\alpha_t \equiv 1 - t$ and $\beta_t \equiv t$ recovers well-known methods such as flow-matching (Lipman et al., 2023; Liu et al., 2023b; Albergo & Vanden-Eijnden, 2023), under the linear interpolant $x_t = (1 - t)x_0 + tx_1$, yielding a simple simulation-free ℓ_2-regression objective,

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(x_0,x_1)\sim\pi(x_0,x_1),\, t\sim\nu(t)} \left[\|v_t^\theta(x_t) - v_t^*(x_t|x_0, x_1)\|^2 \right], \quad v_t^*(x_t|x_0, x_1) = x_1 - x_0, \tag{2}$$

where $\nu(t)$ is a distribution in the interval $(0, 1)$. Despite scalable training of eq. (2), likelihood evaluation is expensive due to computing the divergence (Grathwohl et al., 2018; Meyer et al., 2021).

Flow Maps. To reduce the number of sampling steps at inference, it is possible to learn a *flow map* $X_{s,t} : [0, 1]^2 \times \mathbb{R}^d \mapsto \mathbb{R}^d$ (Boffi et al., 2025a;b). The flow map is the operator that uniquely satisfies the jump condition $X_{s,t}(x_s) = x_t$, for all $(s, t) \in [0, 1]^2$ when $(x_t)_{t\in[0,1]}$ is any solution to the probability flow ODE. Flow maps are often parametrized by the *average velocity* $v_{s,t}^\theta(x_s)$:

$$X_{s,t}^\theta(x_s) = x_s + (t - s)v_{s,t}^\theta(x_s), \quad v_{s,t}^\theta(x_s) = \frac{1}{t - s} \int_s^t v_\tau(x_\tau) d\tau. \tag{3}$$

Furthermore, the true flow map simultaneously satisfies the following consistency conditions,

$$\underbrace{\partial_t X_{s,t}(x_s) = v_{t,t}(X_{s,t}(x_s))}_{\text{Lagrangian}}, \underbrace{\partial_s X_{s,t}(x_s) = -v_{s,s}(x_s)\nabla X_{s,t}(x_s)}_{\text{Eulerian}}, \underbrace{X_{s,t}(X_{r,s}(x_r)) = X_{r,t}(x_r)}_{\text{Semi-group}}. \tag{4}$$

Typical flow map objectives convert the above conditions into distillation losses with stop-gradients (Frans et al., 2025; Geng et al., 2025; Song et al., 2023; Sabour et al., 2025).

2.2 Normalizing Flows

Normalizing Flows (NFs) (Tabak & Vanden-Eijnden, 2010; Tabak & Turner, 2013; Rezende & Mohamed, 2015) parametrize a diffeomorphism between distributions using exactly invertible architectures. The key benefit is exact likelihood evaluation via the classical change of variables formula,

$$\log p_0^\theta(x_0) = \log p_1(x_1) - \log \left| \det \frac{\partial f^\theta(x_1)}{\partial x_1} \right|, \quad \text{where } p_1(x_1) := \mathcal{N}(0, I). \tag{5}$$

Certain classes of NFs are provably universal density approximators (Teshima et al., 2020; Huang et al., 2018) and can be constructed through the composition of m elementary diffeomorphisms, $f^\theta = f^{\theta,m} \circ f^{\theta,m-1} \circ \cdots \circ f^{\theta,1}$, and trained using Maximum Likelihood Estimation (MLE) according to eq. (5). To scale modern NFs, current approaches leverage a causal Transformer backbone with Autoregressive flows (TARFlow) (Zhai et al., 2025), interchanged with permutation layers (c.f. §D.1).

3 METHOD

We introduce NORMALIZING FLOW MAPS (NFM), a new generative model class built from two-time *pushforward maps* that learn generative transport along the probability path of a stochastic interpolant. More precisely, NFMs generalize classical normalizing flows across time, combining the large transport-step capabilities of jump operators while endowing previous flow maps with tractable inverses that enable exact sample likelihoods, i.e., $p_0(x_0)$. We first define the model and its parameterizations (§3.1), then expose the inherent computational properties at the particle and distributional level. These principles yield (self)-distillation objectives (§3.3) resulting in NFM-R that is constructed from pointwise matching of the probability flow and NFM-D that leverages distributional matching of its marginals. Finally, we demonstrate RLHF-based post-training in §3.4.

3.1 NORMALIZING FLOW MAP

Let $p_t = \mathrm{Law}(x_t)$ denote the marginals of a prescribed forward (corruption) process, induced by a stochastic interpolant introduced in §2.1. We seek to perform few-step transport along this probability path through a shared family of two-time maps with tractable inverses and exact change of variables at data time $t = 0$. This motivates the construction of the following generative model, defined as:

Definition 1 (Normalizing Flow Map)**.** Let $\mathcal{T} \subseteq \{(s,t) : 0 \le t < s \le 1\}$ be a set of admissible time pairs containing $(1,0)$. A *normalizing flow map* is a jointly parameterized family

$$\mathcal{F}_\theta = \left\{ f^\theta_{s,t} : \mathbb{R}^d \to \mathbb{R}^d \right\}_{(s,t)\in\mathcal{T}}, \tag{6}$$

where, for every admissible θ and $(s,t) \in \mathcal{T}$, $f^\theta_{s,t}$ is a $\mathcal{C}^\infty$ diffeomorphism satisfying

$$(f^\theta_{s,t})^{-1} \circ f^\theta_{s,t} = f^\theta_{s,t} \circ (f^\theta_{s,t})^{-1} = \mathrm{Id}. \tag{7}$$

Additionally, given a source density p_s, the map induces the density $p^\theta_{s\to t} := (f^\theta_{s,t})_\# p_s$ through

$$\log p^\theta_{s\to t}(x_t) = \log p_s(x_s) - \log \left| \det \frac{\partial f^\theta_{s,t}(x_s)}{\partial x_s} \right|, \qquad x_t = f^\theta_{s,t}(x_s),\ x_s \sim p_s. \tag{8}$$

Each supported function $f_{s,t}(x_s)$ that maps $(s,t,x_s) \mapsto x_t$ provides an invertible transformation between intermediate times $s \to t$. Consequently, density evaluation at time t is possible whenever the source density $p_s(x_s)$ is tractable. Moreover, starting from the prior p_1, compositions of admissible jumps support few-step sampling and exact density evaluation through repeated application of eq. (8). A map of particular interest is the one-step map which tractably admits the density $p^\theta_{1\to 0}(x_0)$,

$$x_1 \sim p_1(x_1), \qquad x_0 = f^\theta_{1,0}(x_1), \qquad p^\theta_{1\to 0} = (f^\theta_{1,0})_\# p_1. \tag{9}$$

We note that eq. (9) provides one-step sample likelihoods. We now make the following remark.

Remark 1. Restricting NFM to $\mathcal{T} = \{(1,0)\}$ recovers the structure of a normalizing flow.

Differences to modern NF MLE training. To scale state-of-the-art NF's via MLE training (Zhai et al., 2025; Chen et al., 2026) it is crucial to smooth out the data distribution p_0 via noise injection p_σ, i.e., $x_\sigma = x_0 + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, which prevents well-known training stability issues, e.g., poor performance, and *numerical* loss of invertibility (Behrmann et al., 2021). While the resulting flow remains invertible and provides exact likelihoods for x_σ, clean samples are recovered at inference time using Tweedie's posterior mean estimator at noise level σ, $\hat{x}_0 = \mathbb{E}[x_0 \mid x_\sigma] = x_\sigma + \sigma^2 \nabla_{x_\sigma} \log p^\theta_\sigma(x_\sigma)$ (Miyasawa, 1961). This additional denoising transformation is not covered by the change-of-variables guarantee and thus likelihoods over clean samples are not available.

Remark 2. NFs trained with noise-augmented MLE provide exact likelihoods at p_σ, whereas NFMs amortize transport across noise levels and provide clean sample likelihoods p_0.

We provide further details of practical training instability of MLE without noise injection in §B.

NFM parametrizations. We construct NFMs from a base NF architecture $\Phi^\theta_{s,t}$ and use it to instantiate three distinct NFM parametrizations (A) – (C) (see §D.3.4 for (C)). The simplest parametrization is the *dual-time* NFM, which directly conditions the flow on the source and target times, (A) : $f^\theta_{s,t}(x_s) = \Phi^\theta_{s,t}(x_s, s, t)$. This construction is invertible for every admissible pair (s,t), but does not enforce the flow map identity $f^\theta_{s,s}(x_s) = x_s$. When the limit $t \to s$ is needed in pointwise matching, the identity can be incorporated into the parametrization. To this end, we use $\Phi^\theta_{s,t}$ to parametrize a two-time denoiser $\delta^\theta_{s,t}$, corresponding to a generalized form of x-prediction (Li & He, 2026; Lee et al., 2026a; Lu et al., 2026a),

$$\text{(B)} : \delta^\theta_{s,t}(x) \coloneqq \Phi^\theta_{s,t}(x, s, t), \quad v^\theta_{s,t}(x) \coloneqq \frac{x - \delta^\theta_{s,t}(x)}{s}, \quad f^\theta_{s,t}(x_s) \coloneqq \frac{t}{s}x + \frac{s-t}{s}\delta^\theta_{s,t}(x). \tag{10}$$

This *denoiser NFM* satisfies $f^\theta_{s,s}(x_s) = x_s$ by construction, while its diagonal velocity $v^\theta_{s,s}$ can be trained to match the instantaneous velocity of the probability flow. Moreover, at $t = 0$ we have $f^\theta_{t,0} = \delta^\theta_{t,0}$, ensuring that the endpoint map, i.e., from any time $t \to 0$, remains solely an invertible NF.

3.2 Pointwise and Distributional Matching

An NFM describes generative transport with respect to a target stochastic interpolant, through both its action on particles and its induced pushforward density. At optimality, the particle and distributional perspectives must be *consistent* in describing the same stochastic interpolant, yielding two complementary matching laws that describe transport along an NFM. We now develop these pointwise and distributional matching principles which, in turn, result in self-distillation losses in §3.3.

Pointwise matching laws. At the particle level, an NFM resembles a conventional flow map operator and jumps along the stochastic interpolant I_t, with a target conditional velocity field, $v^*_t(x_t|x_0, x_1)$, as defined in Equation (2). In particular, for the time intervals $\mathcal{T}$, an NFM $f^\theta_{s,t}$ parameterizes a conventional flow map restricted to admissible invertible maps. In pointwise matching, we seek to recover the particle correspondence specified by the probability-flow ODE. This is achieved via losses based on the flow map consistency conditions (Equation (4)), which often involve the average and instantaneous velocities of the flow map, both of which can be recovered from the NFM parametrization.

Distributional matching laws. At the distributional level, we aim to transport the pushforward distribution to match the *marginal law* of the forward process $p_t(x_t)$. Specifically, given a time pair $(s,t) \in \mathcal{T}$, the NFM induced pushforward distribution is required to match $p^\theta_{s\to t} := (f^\theta_{s,t})_\# q_s$. Exploiting the invertibility of NFM, we can write equivalent characterizations as the pushforward or pullback; in other words, we have that $\left(f^\theta_{s,t}\right)_\# p_s = p_t \iff \left(\left(f^\theta_{s,t}\right)^{-1}\right)_\# p_t = p_s$. This law can be quantified under the KL divergence, $\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^\theta_{s\to t}\right)$. Thus, matching the generated distribution to the destination marginal is equivalent to matching the inverse-transformed destination distribution to the source marginal. When $s = 1$, the tractable reference density p_1 enables exact evaluation of $p^\theta_{1\to t}$ through the change of variables afforded by an NFM at intermediate times instead of optimizing for a weighted Evidence Lower Bound (ELBO) as in diffusion models (Kingma et al., 2021).

3.3 Distillation Losses

The matching laws developed in §3.2 yield two complementary perspectives on an NFM. We next convert these into training frameworks, resulting in two model variants NFM-Regression (NFM-R) and NFM-Distribution (NFM-D) that exploit pointwise and distributional laws, respectively.

NFM Regression. To design NFM-R, we restrict our focus to the denoiser parametrization of an NFM eq. (10). Furthermore, we instantiate a variant of the Eulerian consistency with conditional velocity $v^*_t(x_t|x_0, x_1)$ and the stopgradient operator (sg) as introduced in improved Mean Flow (iMF) and Pixel MeanFlow (pMF) (Geng et al., 2026; Lu et al., 2026a),

$$\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{(s,t)\sim\nu(s,t),(x_0,x_1)\sim\pi}\left[\left\| v^*_t + \mathrm{sg}((t-s)(\partial_s v^\theta_{s,t}(x_s) + \nabla v^\theta_{s,t}(x_s) v^\theta_{s,s}(x_s))) - v^\theta_{s,t}(x_s)\right\|^2\right].$$

We highlight that the loss $\mathcal{L}_{\mathrm{R}}$ implicitly trains the one-step NFM map $f^\theta_{s,0}$ through intermediate time regression. Although the density of multi-step samples is inaccessible, NFM-R natively

supports multi-step sampling. Crucially, inference can be performed using the one-step map $f^{\theta}_{1\to 0}$, which enjoys exact sample likelihoods as it corresponds to inverting the NF backbone $\delta^{\theta}_{1\to 0}(x_1) = \Phi^{\theta}_{1,0}(x_1)$. Additionally, as proven in §A.1, $\mathcal{L}_{\mathrm{R}}$ directly controls the 2-Wasserstein distance between p_0 and $p^{\theta}_{1\to 0}$, providing endpoint control for pointwise distillation.

NFM Distribution. NFM-D exploits distributional matching across interpolation times to learn the marginal probability path. We first construct a single source time objective, and then provide its extension to intermediate source times in § A. Specifically, we fix $s = 1$ and consider the restricted time pair set $\mathcal{T}_{\mathrm{S}} := \{(s,t) \in \mathcal{T} \mid s = 1\} \subset \mathcal{T}$. We seek to minimize the time-averaged marginal KL discrepancy $\mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right]$. We exploit conditioning to circumvent the generally intractable marginal p_t, and introduce the conditional single-time objective, akin to SoftFlow (Kim et al., 2020),

$$\mathcal{L}_{\mathrm{D}} := \mathbb{E}_{t,x_0}\mathbb{D}_{\mathrm{KL}}\left(p_t(\cdot \mid x_0)\middle\|p^{\theta}_{1\to t}\right) \overset{\mathrm{c}}{=} \mathbb{E}_{t,x_t}\left[\frac{1}{2}\left\|(f^{\theta}_{1,t})^{-1}(x_t)\right\|^2 - \log\left|\det\frac{\partial(f^{\theta}_{1,t})^{-1}(x_t)}{\partial x_t}\right|\right], \quad (11)$$

where $\overset{\mathrm{c}}{=}$ denotes equality up to an additive constant independent of θ. As stated in the following proposition, conditional supervision upper-bounds the marginal objective while preserving its minimizers, allowing us to learn the marginal pushforward without evaluating p_t.

Proposition 1 (Conditional NFM-D controls marginal NFM-D)**.** *Assume that* $\mathbb{E}_t[\mathcal{I}(x_0; x_t)] < \infty$*, where* $\mathcal{I}$ *denotes mutual information. Then, for every* θ*,*

$$\mathcal{L}_{\mathrm{D}} = \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right] + \mathbb{E}_t\left[\mathcal{I}(x_0; x_t)\right] \geq \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right].$$

Consequently, the conditional and marginal objectives share the same minimizers,

$$\arg\min_{\theta} \mathcal{L}_{\mathrm{D}}(\theta) = \arg\min_{\theta} \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right].$$

The gap between the conditional and marginal objectives is the mutual information, which depends only on the prescribed corruption process. Separately, empirical conditional NFM-D is bounded below (§ A), avoiding the unbounded likelihood degeneracy that can arise in empirical MLE. Finally, we also relate accuracy along the corruption path to the quality of the generated distribution.

Proposition 2 (Single-time NFM-D endpoint control)**.** *Assume* $t \mapsto \mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)$ *is locally Lipschitz at* $t = 0$*, with constant* C *for* $t \in [0, t_0]$*. Suppose the time-sampling density satisfies* $\nu(t) \geq \nu_{\min} > 0$ *on* $[0, t_0]$*. If* $\mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right] \leq C\nu_{\min}t_0^2/2$*, then*

$$\mathbb{D}_{\mathrm{KL}}\left(p_0 \,\middle\|\, p^{\theta}_{1\to 0}\right) \leq \sqrt{\frac{2C}{\nu_{\min}}\mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^{\theta}_{1\to t}\right)\right]}.$$

Thus, minimizing the single-time NFM-D objective provides direct control of the generated endpoint distribution, provided the marginal discrepancy is locally regular. We provide the proof in §A.

3.4 Post-training Normalizing Flow Maps

A key advantage of NFM is tractable access to the endpoint density $p^{\theta}_{1\to 0}(x_0)$. This makes distribution-level post-training objectives directly available:

$$\mathcal{J}(\theta) = \mathbb{E}_{x_0\sim p^{\theta}_{1\to 0}}[R(x_0)] + \beta\ \mathrm{Reg}(p^{\theta}_{1\to 0}),\ \beta > 0. \quad (12)$$

Here $R : \mathbb{R}^d \to \mathbb{R}$ is a scalar reward function , and $\mathrm{Reg}(p^{\theta}_{1\to 0})$ is either the entropy $\mathcal{H}(p^{\theta}_{1\to 0})$, as in our maximum-entropy experiments, or $-\mathbb{D}_{\mathrm{KL}}(p^{\theta}_{1\to 0}\|p_{\mathrm{ref}})$ for a fixed reference density.

For diffusion and flow matching models, endpoint likelihoods are expensive to evaluate, making Equation (12) challenging to optimize directly. Post-training must therefore leverage alternative learning signals, e.g., transition-level policy updates (Black et al., 2024; Liu et al., 2025a), adjoint or value-gradient methods (Domingo i Enrich et al., 2025; Liu et al., 2025b), forward process matching (Zheng et al., 2026), and ELBO-based proxies (Wallace et al., 2024; McAllister et al., 2026). NFMs avoid the complications introduced by these methods as endpoint log densities give direct access to score-function gradients (Williams, 1992), which can be used with non-differentiable rewards, and they provide the density ratios employed by prominent large language model post-training methods like DPO (Rafailov et al., 2023) and GRPO (Shao et al., 2024).

4 EXPERIMENTS

We empirically validate NFM on image generation and RL post-training, as well as on a variety of molecular conformation sampling tasks. We use denoiser and dual-time parametrizations (§ 3) for NFM-R and NFM-D respectively. For both NFMs our base invertible architecture $\Phi^{\theta}_{s,t}$ is based off of the recently introduced iTARFlow model (Lu et al., 2026b) (see §D.2), which also acts as the principal baseline when trained using MLE. We also report additional ablations in §E.

4.1 IMAGE GENERATION AND POST-TRAINING



Figure 2: NFM post-training. (a) NFM-D, *golden retriever*; (b) NFM-D post-trained, *a single golden retriever on a sandy beach*; (c) NFM-R, *bighorn sheep*; (d) NFM-R post-trained, *bighorn sheep on a sandy beach at sea*.

Image generation. We validate NFM-R and NFM-D on conditional image generation on CIFAR-10 (Krizhevsky, 2009) and ImageNet-64 (Deng et al., 2009) both in terms of Fréchet Inception Distance (FID) (Heusel et al., 2017) and Feature Likelihood Divergence (FLD) (Jiralerspong et al., 2023) (Tables 1a & 1b). We find that NFM-D and NFM-R improve both metrics over MLE trained iTARFlow, and NFM-D outperforms SoftFlow on CIFAR-10. Moreover, NFM-R achieves the best overall performance. Maintaining invertibility with NFM-R requires noise annealing during training and `FP64` inference. Additionally, NFM-R models with an invertible one-step map may trade invertibility for improved quality using multistep sampling at inference time. Finally, training iTARFlow on clean data preserves invertibility but substantially worsens sample quality. We report additional experiments and ablations in § E.

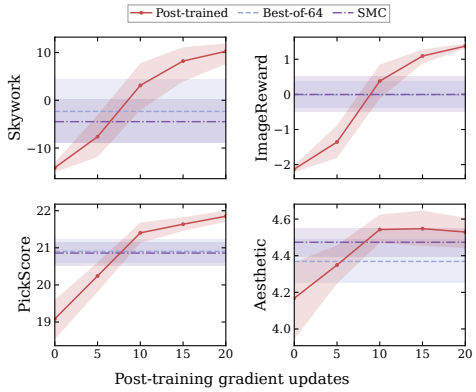


Figure 3: Post-trained NFM-D vs. best-of-64 and SMC baselines.

Table 1: Class-conditional generation results. Best results are in **bold**, and second-best are underlined.

(a) CIFAR-10 results.

Model	FID ↓	FLD ↓
Non-Invertible NFs		
iTARFlow	6.55	8.38
NFM-R (Ours)	**3.87**	7.00
NFM-R (Ours) *noise anneal, 2 steps*	4.63	**6.89**
Invertible NFs		
iTARFlow *no noise*	40.83	23.21
SoftFlow	6.76	7.84
NFM-D (Ours)	6.26	7.70
NFM-R (Ours) *noise anneal*	**5.17**	**7.62**

(b) ImageNet-64 results.

Model	Param.	FID ↓	FLD ↓
Diffusion & Few Step Models			
iDDPM (Nichol & Dhariwal, 2021)	270M	2.92	–
EDM (Karras et al., 2022)	296M	1.36	–
CD (Song et al., 2023)	296M	4.70	–
Iterative TARFlow (Chen et al., 2026)	350M	2.05	–
Non-Invertible NFs			
TARFlow (Zhai et al., 2025)	466M	2.90	–
iTARFlow	122M	4.89	6.98
NFM-R (Ours)	127M	**3.60**	**3.15**
Invertible NFs			
NFM-D (Ours)	123M	**4.59**	**6.78**

RL Post-training. NFM models provide access to exact sample likelihoods, enabling RL-based post-training. Here we post-train *purely class-conditional* NFM models to align their samples with a text-caption supplied to a Visual Language Model (VLM) via on-policy RL. Specifically, we optimize eq. (12); NFM-R uses maximum entropy regularization (Jaynes, 1957; Haarnoja et al., 2018) and pathwise gradients, while we train NFM-D with KL-regularized REINFORCE (Williams,

1992). We use a suite of rewards for training and evaluation: ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), HPSV2.1 (Wu et al., 2023), and Skywork-VL (Wang et al., 2025) (See § D.5 for more details). Figure 2 clearly demonstrates the effect of RL fine-tuning on the models' output, and in Figure 3 we observe clear gains on multiple rewards against best-of-N and sequential Monte Carlo (SMC) baselines over the course of training. We report additional experiments, including on-policy RL post-training on the CIFAR-10 NFM models with generated samples in §E.

4.2 Boltzmann Generators

We now apply NFMs to equilibrium sampling with Boltzmann Generators (Noé et al., 2019) (BGs), which are designed to sample molecular states at thermodynamic equilibrium. BGs aim to sample from the *Boltzmann distribution*: $\mu_{\text{target}}(x) \propto \exp\left(-\mathcal{E}(x)/k_{\text{B}}T\right)$, where $\mathcal{E}(x)$ is the potential energy of a conformation $x \in \mathbb{R}^{n\times 3}$ of a molecular system with n atoms, and $k_{\text{B}}T$ denotes the thermal energy at temperature T. Observables $\phi(x)$ under μ_{target} are commonly estimated using self-normalized importance sampling (IS) (Owen, 2013) and sequential Monte Carlo (SMC) (Tan et al., 2025a). We refer to §C for a detailed discussion on BGs, including the baselines we consider.

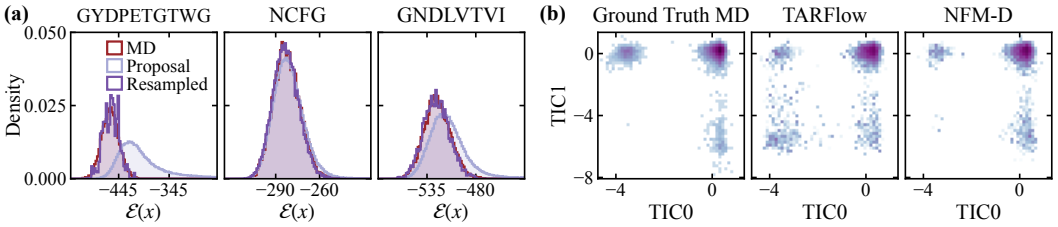


Figure 4: **a)** Energy histograms of MD data, the NFM-D proposal, and the resampled distribution, for a single-system model (left) and a transferable model evaluated on a 4- and 8-residue peptides (middle, right). **b)** TICA projections for the ground truth MD, TARFlow, and NFM-D on NHQYGSDP.

Metrics. We evaluate BGs on single-peptide systems using the 2-Wasserstein distance between energy distributions ($\mathcal{E}$-$\mathcal{W}_2$) and between dihedral-angle distributions ($\mathbb{T}$-$\mathcal{W}_2$). Transferable models are additionally evaluated with TICA-$\mathcal{W}_2$ on the first two TICA components. Following Rehman et al. (2026c), we omit effective sample size. The MLE counterpart of NFM is SBG (Tan et al., 2025a). We report NFM-D throughout, as it outperforms NFM-R. Further details in §C.2 and §E.2.

Single-peptide Systems We evaluate NFM on equilibrium conformation sampling for Chignolin (GYDPETGTWG, 10 residues), with results reported in Table 2. Figure 4 shows the corresponding energy distributions. Chignolin is the most challenging single-peptide benchmark due to the presence of the β-hairpin secondary structure. NFM-D substantially outperforms SBG, and achieves performance competitive with the autoregressive ArBG baseline. To our knowledge, no other continuous flow baseline has reported comparable performance on Chignolin.

Table 2: Performance comparison on Chignolin. SBG-iTARFlow refers to our reimplementation of SBG, used to control for architectural, optimizer, and training-recipe changes. Best results are in **bold**; second-best results are underlined.

Algorithm	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$
ArBG (Rehman et al., 2026c)	7.94 ± 1.05	2.78 ± 0.11
SBG SMC (Tan et al., 2025a)	10.82 ± 7.21	3.78 ± 0.44
SBG-iTARFlow SMC	10.87 ± 10.18	3.56 ± 0.92
NFM-D IS (Ours)	5.12 ± 0.26	$\mathbf{2.71 \pm 0.05}$
NFM-D SMC (Ours)	$\mathbf{3.42 \pm 0.27}$	2.81 ± 0.12

Transferable generation. We evaluate zero-shot generalization to unseen peptides (Klein & Noé, 2024) using a transferable NFM-D trained on ManyPeptidesMD (Tan et al., 2025b), with results reported in Table 4. NFM-D consistently improves over its MLE-trained TARFlow counterpart, demonstrating the benefit of our training objective in the transferable setting. NFM-D also outperforms Prose and Robin on five of six metrics. Finally NFM-D uses substantially fewer parameters than Prose (118M vs. 288M). Figure 4 shows a representative TICA projection,

Table 3: Performance of transferable models with temperature tuning. Best results are in **bold**; second-best results are underlined.

	4AA (30 systems)			8AA (30 systems)		
Model	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$
Robin	1.168	0.886	0.471	4.251	2.325	0.943
Robin SMC	1.079	0.874	0.463	4.263	2.315	0.977
NFM-D (Ours)	**0.865**	**0.745**	**0.398**	**4.148**	**2.296**	**0.892**

illustrating the improved agreement of NFM-D with the ground truth distribution. Figure 4 shows additional energy distributions.

Table 4: Transferable models comparison on unseen 4- and 8-residue peptides, using a budget of 10^4 energy evaluations and $T = 1$. Best results are in **bold**; second-best results are underlined.

	4AA (30 systems)			8AA (30 systems)		
Model	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$
TimeWarp (Klein et al., 2023a)	7.237	2.204	0.993	—	—	—
BioEmu (Lewis et al., 2025)	90.079	2.037	1.479	193.873	4.638	1.601
UniSim (Yu et al., 2025)	$> 10^4$	2.766	1.733	$> 10^3$	6.156	1.495
ECNF++ (Tan et al., 2025b)	10.032	1.121	0.572	—	—	—
Prose (Tan et al., 2025b)	0.932	0.752	**0.367**	10.038	2.456	0.988
Robin (Rehman et al., 2026c)	1.077	0.900	0.447	**6.373**	2.440	0.986
TARFlow (Tan et al., 2025b)	1.260	0.924	0.492	11.298	2.733	1.087
NFM-D (Ours)	**0.907**	**0.750**	0.387	7.396	**2.314**	**0.890**

Temperature tuning. Table 3 shows that temperature tuning can further improve transferable performance. For NFM-D, we tune the temperature of the base distribution, $p_{1,T} \coloneqq \mathcal{N}(0, TI)$, while Robin tunes its attention temperature. With their respective tuned temperatures, NFM-D achieves the best performance, including a substantial improvement in $\mathcal{E}$-$\mathcal{W}_2$ on 8-residue peptides.

5 Related Work

Normalizing flows. NFs (Tabak & Vanden-Eijnden, 2010; Tabak & Turner, 2013; Rezende & Mohamed, 2015; Dinh et al., 2017) are invertible generative models whose *exact* likelihood can be evaluated via the change-of-variables formula. Despite this appealing property, scaling to tasks like image generation remains challenging due to numerical instabilities (Xu & Campbell, 2023; Andrade, 2024) and architectural constraints (Chen et al., 2019). Recently, TARFlows (Zhai et al., 2025) have demonstrated impressive capabilities in image (Gu et al., 2026a; Shen et al., 2026) and video generation (Gu et al., 2026c), as well as in sampling from Boltzmann distributions (Midgley et al., 2023; Tan et al., 2025a;b). While TARFlow is trained via noise augmented MLE (Loaiza-Ganem et al., 2023; Zhai et al., 2025), alternative training strategies include training over a probability path (Kim et al., 2020; Chen et al., 2026; Gu et al., 2026b), and direct regression-based training (Rehman et al., 2026b).

Stochastic interpolants and flow maps. Diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) and Flow Matching (Lipman et al., 2023; Liu et al., 2023b; Albergo & Vanden-Eijnden, 2023; Tong et al., 2024) models have achieved state-of-the-art sample fidelity across numerous modalities (Rombach et al., 2022; Abramson et al., 2024; Zeni et al., 2025). Generating samples and evaluating likelihoods with these models typically require costly numerical integration (Salimans & Ho, 2022; Chen et al., 2018; Grathwohl et al., 2018), motivating the development of few-step models (Song et al., 2023; Song & Dhariwal, 2024; Frans et al., 2025; Geng et al., 2025; 2026), and the flow map framework (Boffi et al., 2025a;b; Sabour et al., 2025). While flow maps accelerate sampling, likelihood evaluation remains costly (Rehman et al., 2026a). Alternatively, proxy likelihood evaluation can be amortized using an auxiliary divergence predictor (Ai et al., 2026; OuYang et al., 2026).

6 Conclusion

In this paper, we introduce Normalizing Flow Maps, a family of invertible transport maps that combine few-step sampling with exact change-of-variables densities for one-step endpoint samples. We demonstrate the application of these cheap likelihoods in Boltzmann generators through importance sampling, and straightforward RL-based post-training of image generation models. Furthermore, we demonstrate superior sample quality to existing NF methods and avoid the use of Tweedie denoising which requires spatial gradients and breaks invertibility guarantees. While NFM-R obtains the highest sample quality, it is prone to becoming numerically non-invertible, which impedes RL methods relying on the score-function or KL-regularization. This points to the importance of designing training schemes that enjoy the benefits of both pointwise and distributional consistency.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity of the author-written text, and improve the correctness and presentation of mathematical claims; identify potentially relevant literature; assist with implementation of the method; and provide critical feedback on our research ideas and methodology. We did not use generative AI tools to draft sections of the paper or generate synthetic datasets.

The authors reviewed all AI-assisted work. We read the literature suggested by AI tools and used separate LLM reviewers to suggest code checks and tests. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our method is a general purpose generative modeling approach. In the image experiments, we post-train class-conditional generators using text–to–image reward models. These rewards are imperfect proxies for human preferences: optimizing them may amplify biases in the reward models or favor images that score well without matching a user's intent. In the molecular experiments, we evaluate equilibrium conformation sampling on existing peptide benchmarks using molecular dynamics data. The molecules studied in this work are considered safe and non-toxic. Applying the method to other molecular systems would require evaluation specific to those uses.

REPRODUCIBILITY STATEMENT

We provide the following information to support reproduction of our results:

- **Method and theory:** Section 3 defines the model, training objectives, and theoretical results.
- **Main experiments:** Section 4 describes the image generation, post-training, and Boltzmann generator experiments and reports the main results.
- **Implementation and additional experiments:** Appendix D and E carefully explains all the implementation details and additional ablations to properly implement NFM.
- **Code:** We will release the code in a public GitHub repository.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Xinyue Ai, Yutong Kelly He, Albert Gu, Russ Salakhutdinov, Zico Kolter, Nicholas Boffi, and Max Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. In *International Conference on Learning Representations*, volume 2026, pp. 44940–44977, 2026.

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

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Berni J Alder and Thomas Everett Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

Daniel Andrade. Stable training of normalizing flows for high-dimensional variational inference. *arXiv preprint arXiv:2402.16408*, 2024.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 843–852, June 2023.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1792–1800. PMLR, 2021.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. 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. 3462–3483. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ben-hamu24a.html.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Zander W. Blasingame and Chen Liu. Greed is good: A unifying perspective on guided generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=s14pdQgoLb.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856. URL https://openreview.net/forum?id=cqDH0e6ak2.

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*, 2025b. URL https://openreview.net/forum?id=Di5apl8HSH.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in neural information processing systems*, 32, 2019.

Tianrong Chen, Jiatao Gu, David Berthelot, Joshua M. Susskind, and Shuangfei Zhai. Normalizing flows with iterative denoising. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oUVzFuemEn.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

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.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15 (1):10416, 2024.

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. String method for the study of rare events. *Physical Review B*, 66(5):052301, 2002.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, Inc., USA, 1st edition, 1996. ISBN 0122673700.

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.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

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, 2026a.

Jiatao Gu, Tianrong Chen, Ying Shen, David Berthelot, Shuangfei Zhai, and Josh Susskind. Normalizing trajectory models. *arXiv preprint arXiv:2605.08078*, 2026b.

Jiatao Gu, Ying Shen, Tianrong Chen, Laurent Dinh, Yuyang Wang, Miguel Angel Bautista, David Berthelot, Josh Susskind, and Shuangfei Zhai. Starflow-v: End-to-end video generative modeling with autoregressive normalizing flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9084–9094, June 2026c.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. 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. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International conference on machine learning*, pp. 2078–2087. PMLR, 2018.

E. T. Jaynes. Information Theory and Statistical Mechanics. *Phys. Rev.*, 106:620–630, 1957. doi: 10.1103/PhysRev.106.620.

Marco Jiralerspong, Joey Bose, Ian Gemp, Chongli Qin, Yoram Bachrach, and Gauthier Gidel. Feature likelihood divergence: evaluating the generalization of generative models using samples. *Advances in Neural Information Processing Systems*, 36:33095–33119, 2023.

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.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184. IEEE, 2024.

Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33:16388–16397, 2020.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663. Curran Associates, Inc., 2023. doi: 10.52202/075280-1594. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/73aacd8b3b05b4b503d58310b523553c-Paper-Conference.pdf.

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

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

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026a. URL https://arxiv.org/abs/2602.16813.

Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. *arXiv preprint arXiv:2604.17415*, 2026b.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew YK Foong, Victor García Satorras, Osama Abdin, Bastiaan S Veeling, Iryna Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761): eadv9817, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026. URL https://arxiv.org/abs/2511.13720.

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

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21450–21474. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/liu23f.html.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025a.

Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zhen Liu, Tim Z. Xiao, Carles Domingo-Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for flow matching alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=6MmOy2Ji8V.

Gabriel Loaiza-Ganem, Brendan Leigh Ross, Luhuan Wu, John Patrick Cunningham, Jesse C Cresswell, and Anthony L Caterini. Denoising deep generative models. In *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, pp. 41–50. PMLR, 2023.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=WUK8JIeetF.

Yiyang Lu, Qiao Sun, Xianbang Wang, Zhicheng Jiang, Hanhong Zhao, and Kaiming He. Bidirectional normalizing flow: From data to noise and back. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2069–2078, 2026b.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, volume 2026, pp. 36352–36372, 2026.

Raphael A Meyer, Cameron Musco, Christopher Musco, and David P Woodruff. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, pp. 142–155. SIAM, 2021.

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

Koichi Miyasawa. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38:181–188, 1961.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

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 M. Boffi, Pradeep Ravikumar, Jose Miguel Hernandez-Lobato, Max Simchowitz, Benjamin Kurt Miller, and Omar Chehab. Few-Step Boltzmann Generators via Scalable Likelihood Flow Maps, June 2026. URL http://arxiv.org/abs/2606.29110. arXiv:2606.29110 [cs.LG].

Art B Owen. Monte carlo theory, methods and examples, 2013.

Raj Kumar Pathria. *Statistical Mechanics: International Series of Monographs in Natural Philosophy*, volume 45. Elsevier, 2017.

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.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

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, Oscar Davis, Jiarui Lu, Jian Tang, Michael M. Bronstein, Yoshua Bengio, Alexander Tong, and Joey Bose. Efficient regression-based training of normalizing flows for boltzmann generators. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=ctdnzPxDI3.

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

Danilo Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1530–1538. PMLR, June 2015. URL https://proceedings.mlr.press/v37/rezende15.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pzHuesCvcO.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Gianluca Scarpellini, Ron Shprints, Peter Holderrieth, Juno Nam, Pranav Murugan, Rafael Gómez-Bombarelli, Tommi Jaakkola, Maruan Al-Shedivat, Nicholas Matthew Boffi, and Avishek Joey Bose. Few-step cofolding with all-atom flow maps. *arXiv preprint arXiv:2606.08375*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Ying Shen, Tianrong Chen, Yuan Gao, Yizhe Zhang, Yuyang Wang, Miguel Ángel Bautista, Shuangfei Zhai, Joshua M. Susskind, and Jiatao Gu. STARFlow2: Bridging Language Models and Normalizing Flows for Unified Multimodal Generation, May 2026. URL http://arxiv.org/abs/2605.08029. arXiv:2605.08029 [cs.CV].

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265. PMLR, June 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

E. G. Tabak and Cristina V. Turner. A Family of Nonparametric Density Estimation Algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013. ISSN 1097-0312. doi: 10.1002/cpa.21423. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21423.

Esteban G. Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010. ISSN 15396746, 19450796. doi: 10.4310/CMS.2010.v8.n1.a11. URL https://link.intlpress.com/JDetail/1806265554799382529.

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 2025a. URL https://proceedings.mlr.press/v267/tan25a.html.

Charlie B. Tan, Majdi Hassan, Leon Klein, Saifuddin Syed, Dominique Beaini, Michael M. Bronstein, Alexander Tong, and Kirill Neklyudov. Amortized sampling with transferable normalizing flows. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=JenfC3ovzU.

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Mark E Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Xiaokun Wang, Peiyu Wang, Jiangbo Pei, Wei Shen, Yi Peng, Yunzhuo Hao, Weijie Qiu, Ai Jian, Tianyidan Xie, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork-vl reward: An effective reward model for multimodal understanding and reasoning, 2025. URL https://arxiv.org/abs/2505.07263.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

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

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 democratizing biomolecular interaction modeling. *bioRxiv*, 2025. doi: 10.1101/2024.11.19.624167. URL https://www.biorxiv.org/content/early/2025/05/06/2024.11.19.624167.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zuheng Xu and Trevor Campbell. Embracing the chaos: analysis and diagnosis of numerical instability in variational flows. *Advances in Neural Information Processing Systems*, 36:32360–32386, 2023.

Ziyang Yu, Wenbing Huang, and Yang Liu. UniSim: A unified simulator for time-coarsened dynamics of biomolecules. 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. 73097–73114. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/yu25j.html.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

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.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

Daniel M Zuckerman. Equilibrium sampling in biomolecular simulations. *Annual review of biophysics*, 40:41–62, 2011.

Appendix

A Theoretical Results **20**
A.1 NFM-R Proofs 20
A.2 NFM-D Proofs 22

B Exact likelihood without Tweedie denoising **27**

C Extended background on Boltzmann Generators **28**
C.1 Baselines 28
C.2 Metrics 29

D Implementation details **30**
D.1 Transformer Autoregressive Flows 30
D.2 NFM Architecture 30
D.3 Algorithmic aspects 30
D.4 Pseudocode 34
D.5 Reinforcement Learning Finetuning 36

E Additional Results **37**
E.1 Image Generation 37
E.2 Boltzmann Generators 41
E.3 Transferable Samplers 43

A Theoretical Results

A.1 NFM-R Proofs

In the following, we adapt Boffi et al. (2025a;b) settings to improved Mean Flow.

For the linear interpolant in eq. (2),

$$x_s = I_s(x_0, x_1) = (1-s)x_0 + sx_1, \qquad (x_0, x_1) \sim \pi, \qquad x_s \sim p_s.$$

The conditional and marginal velocities are, respectively,

$$v_s^*(x_s \mid x_0, x_1) = x_1 - x_0, \qquad v_s(x) = \mathbb{E}[v_s^*(x_s \mid x_0, x_1) \mid x_s = x]\,. \tag{13}$$

From the denoiser parametrization in eq. (10),

$$f_{s,t}^\theta(x) = x + (t-s)v_{s,t}^\theta(x), \qquad f_{s,0}^\theta(x) = \delta_{s,0}^\theta(x) = x - sv_{s,0}^\theta(x), \tag{14}$$

and $p_{1\to 0}^\theta = (f_{1,0}^\theta)_\# p_1$.

Regression loss and Eulerian consistency. Let $r_{s,t}^\theta$ denote the regression residual in section 3.3, evaluated on the interpolant sample x_s:

$$\begin{aligned} r_{s,t}^\theta := v_s^*(x_s \mid x_0, x_1) - v_{s,t}^\theta(x_s) \\ + (t-s)\left(\partial_s v_{s,t}^\theta(x_s) + \nabla_x v_{s,t}^\theta(x_s) v_{s,s}^\theta(x_s)\right). \end{aligned} \tag{15}$$

The derivative ∂_s acts on the first time argument while holding the second fixed. Stop-gradient changes the training gradient but not the loss value, so it is omitted here. We introduce

$$\mathcal{L}_{\mathrm{R}}(s,t) = \mathbb{E}_{(x_0,x_1)\sim\pi}\left[\|r_{s,t}^\theta\|^2\right], \qquad \mathcal{L}_{\mathrm{R}}(\theta) = \mathbb{E}_{(s,t)\sim\mathcal{T}}\left[\mathcal{L}_{\mathrm{R}}(s,t)\right]. \tag{16}$$

At $t = s$, the residual reduces to $v_s^*(x_s \mid x_0, x_1) - v_{s,s}^\theta(x_s)$. Identifying the learned velocity v_s^θ in eq. (2) with $v_{s,s}^\theta$, define

$$\mathcal{L}_{\mathrm{FM}} := \mathbb{E}_{s\sim\nu(s)}\left[\mathcal{L}_{\mathrm{R}}(s,s)\right], \tag{17}$$

$$\mathcal{L}_{\mathrm{distill}} := \mathbb{E}_{(s,t)\sim\mathcal{T},\, s\neq t}\left[\mathcal{L}_{\mathrm{R}}(s,t)\right], \tag{18}$$

$$\mathcal{L}_{\mathrm{end}} := \int_0^1 \mathcal{L}_{\mathrm{R}}(s,0)\,ds. \tag{19}$$

Here $\nu(s)$ is the time-sampling density from eq. (2), and the expectation defining $\mathcal{L}_{\mathrm{distill}}$ is conditional on an off-diagonal pair. For the formal statement below, suppose the time-sampling law in section 3.3 assigns probability $\eta \in (0,1)$ to diagonal pairs (s,s), with $s \sim \nu(s)$, and probability $1-\eta$ to off-diagonal pairs. Consequently,

$$\mathcal{L}_{\mathrm{R}}(\theta) = \eta\mathcal{L}_{\mathrm{FM}} + (1-\eta)\mathcal{L}_{\mathrm{distill}}. \tag{20}$$

The diagonal is understood as the continuous extension of the admissible time pairs used in section 3.3.

Proposition 3 (Endpoint control for pointwise distillation)**.** *Consider the loss and time-sampling law in eqs. (16) and (20). Assume p_0, p_1 have finite second moments and the marginal velocity v_s admits a probability flow $X_{s,t}$: for $x_1 \sim p_1$, $s \mapsto X_{1,s}(x_1)$ is almost surely absolutely continuous,*

$$\frac{d}{ds}X_{1,s}(x_1) = v_s(X_{1,s}(x_1)) \quad \textit{a.e.}, \qquad (X_{1,s})_\# p_1 = p_s, \qquad X_{1,1} = \mathrm{Id}.$$

Assume the derivatives in eq. (15) exist, $f_{s,0}^\theta$ is continuously differentiable in (s,x) up to the time boundaries, $f_{0,0}^\theta = \mathrm{Id}$, and

$$\int_0^1 \mathbb{E}_{x_s\sim\nu_s}\left[\left\|\partial_s f_{s,0}^\theta(x_s) + \nabla_x f_{s,0}^\theta(x_s) v_s(x_s)\right\|^2\right] ds < \infty.$$

Suppose $\nu(s) \geq \nu_{\min} > 0$ almost everywhere on $(0,1)$ and, for finite constants $\widehat{L}, C_{\mathrm{tr}} \geq 0$ independent of θ,

$$\sup_{s\in[0,1],\, x\in\mathbb{R}^d} \left\|I - \nabla_x f_{s,0}^\theta(x)\right\|_{\mathrm{op}} \leq \widehat{L}, \qquad \mathcal{L}_{\mathrm{end}} \leq C_{\mathrm{tr}}\mathcal{L}_{\mathrm{distill}}.$$

Then

$$W_2^2\big(p_0, p^\theta_{1\to 0}\big) \le \left(\frac{\widehat{L}^2}{\eta\nu_{\min}} + \frac{C_{\mathrm{tr}}}{1-\eta}\right)\mathcal{L}_{\mathrm{R}}(\theta).$$

Proof. Define the endpoint Eulerian residual using the marginal velocity v_s from eq. (13):

$$R^\theta_s(x) := \partial_s f^\theta_{s,0}(x) + \nabla_x f^\theta_{s,0}(x) v_s(x). \tag{21}$$

Fix a reference sample $x_1 \sim p_1$ and consider its true probability-flow trajectory $s \mapsto X_{1,s}(x_1)$. By definition,

$$\frac{d}{ds}X_{1,s}(x_1) = v_s(X_{1,s}(x_1)), \qquad X_{1,1}(x_1) = x_1.$$

Applying the chain rule to the predicted endpoint along this trajectory gives, almost everywhere in s,

$$\begin{aligned}\frac{d}{ds} f^\theta_{s,0}(X_{1,s}(x_1)) &= \partial_s f^\theta_{s,0}(X_{1,s}(x_1)) + \nabla_x f^\theta_{s,0}(X_{1,s}(x_1))\,\frac{d}{ds}X_{1,s}(x_1)\\ &= \partial_s f^\theta_{s,0}(X_{1,s}(x_1)) + \nabla_x f^\theta_{s,0}(X_{1,s}(x_1))\, v_s(X_{1,s}(x_1))\\ &= R^\theta_s(X_{1,s}(x_1)).\end{aligned}$$

Integrating from $s = 0$ to $s = 1$ therefore yields

$$f^\theta_{1,0}(X_{1,1}(x_1)) - f^\theta_{0,0}(X_{1,0}(x_1)) = \int_0^1 R^\theta_s(X_{1,s}(x_1))\, ds.$$

Using $X_{1,1} = \mathrm{Id}$ and $f^\theta_{0,0} = \mathrm{Id}$, this becomes

$$f^\theta_{1,0}(x_1) - X_{1,0}(x_1) = \int_0^1 R^\theta_s(X_{1,s}(x_1))\, ds. \tag{22}$$

Now $(X_{1,0})_\# p_1 = p_0$ and $(f^\theta_{1,0})_\# p_1 = p^\theta_{1\to 0}$. Thus evaluating both maps on the same $x_1 \sim p_1$ gives a coupling of p_0 and $p^\theta_{1\to 0}$. Since the squared Wasserstein distance is the infimum of the expected squared distance over all such couplings,

$$\begin{aligned}W_2^2\big(p_0, p^\theta_{1\to 0}\big) &\le \mathbb{E}_{x_1\sim p_1}\left[\big\|f^\theta_{1,0}(x_1) - X_{1,0}(x_1)\big\|^2\right]\\ &= \mathbb{E}_{x_1\sim p_1}\left[\left\|\int_0^1 R^\theta_s(X_{1,s}(x_1))\, ds\right\|^2\right].\end{aligned}$$

For almost every x_1, the triangle inequality followed by Cauchy–Schwarz in the time variable gives

$$\begin{aligned}\left\|\int_0^1 R^\theta_s(X_{1,s}(x_1))\, ds\right\|^2 &\le \left(\int_0^1 \big\|R^\theta_s(X_{1,s}(x_1))\big\|\, ds\right)^2\\ &\le \left(\int_0^1 1\, ds\right)\left(\int_0^1 \big\|R^\theta_s(X_{1,s}(x_1))\big\|^2\, ds\right)\\ &= \int_0^1 \big\|R^\theta_s(X_{1,s}(x_1))\big\|^2\, ds.\end{aligned}$$

Taking expectations and interchanging the nonnegative integrals by Tonelli's theorem, we obtain

$$\begin{aligned}W_2^2\big(p_0, p^\theta_{1\to 0}\big) &\le \int_0^1 \mathbb{E}_{x_1\sim p_1}\left[\big\|R^\theta_s(X_{1,s}(x_1))\big\|^2\right] ds\\ &= \int_0^1 \mathbb{E}_{x_s\sim p_s}\left[\|R^\theta_s(x_s)\|^2\right] ds.\end{aligned} \tag{23}$$

The last equality uses $(X_{1,s})_\# p_1 = p_s$ for each fixed s. It identifies the one-time distributions of the probability flow and the stochastic interpolant, not their sample paths.

Differentiating $f^\theta_{s,0}(x) = x - sv^\theta_{s,0}(x)$ and using eq. (15) at $t = 0$ yields

$$R^\theta_s(x_s) = \mathbb{E}\big[r^\theta_{s,0} \mid x_s\big] + \big(I - \nabla_x f^\theta_{s,0}(x_s)\big)\big(v^\theta_{s,s}(x_s) - v_s(x_s)\big). \tag{24}$$

Indeed, conditioning the regression residual on x_s replaces $v^*_s(x_s \mid x_0, x_1)$ by $v_s(x_s)$, while $I - \nabla_x f^\theta_{s,0} = s\nabla_x v^\theta_{s,0}$. Conditional Jensen gives

$$\mathbb{E}\left[\left\|\mathbb{E}[r^\theta_{s,0} \mid x_s]\right\|^2\right] \le \mathcal{L}_{\mathrm{R}}(s,0),$$
$$\mathbb{E}\left[\left\|v^\theta_{s,s}(x_s) - v_s(x_s)\right\|^2\right] \le \mathcal{L}_{\mathrm{R}}(s,s).$$

Moreover, $\nu(s) \ge \nu_{\min}$ implies

$$\int_0^1 \mathcal{L}_{\mathrm{R}}(s,s)\, ds \le \frac{\mathcal{L}_{\mathrm{FM}}}{\nu_{\min}}.$$

Applying the triangle inequality in $L^2(ds\, p_s(dx))$ to eq. (24) therefore gives

$$\left(\int_0^1 \mathbb{E}\left[\|R^\theta_s(x_s)\|^2\right] ds\right)^{1/2} \le \sqrt{\mathcal{L}_{\mathrm{end}}} + \frac{\widehat{L}}{\sqrt{\nu_{\min}}}\sqrt{\mathcal{L}_{\mathrm{FM}}}. \tag{25}$$

Combining eqs. (23) and (25) with the endpoint-loss assumption and Cauchy–Schwarz yields

$$\begin{aligned} W_2^2\big(p_0, p^\theta_{1\to 0}\big) &\le \left(\sqrt{C_{\mathrm{tr}}\mathcal{L}_{\mathrm{distill}}} + \frac{\widehat{L}}{\sqrt{\nu_{\min}}}\sqrt{\mathcal{L}_{\mathrm{FM}}}\right)^2 \\ &\le \left(\frac{C_{\mathrm{tr}}}{1-\eta} + \frac{\widehat{L}^2}{\eta\nu_{\min}}\right)\left((1-\eta)\mathcal{L}_{\mathrm{distill}} + \eta\mathcal{L}_{\mathrm{FM}}\right) \\ &= \left(\frac{C_{\mathrm{tr}}}{1-\eta} + \frac{\widehat{L}^2}{\eta\nu_{\min}}\right)\mathcal{L}_{\mathrm{R}}(\theta). \end{aligned}$$

□

Remark 4 (Loss normalization). *If the objective is instead defined as the unweighted sum* $\mathcal{L}_{\mathrm{R}} = \mathcal{L}_{\mathrm{FM}} + \mathcal{L}_{\mathrm{distill}}$, *the same proof gives*

$$W_2^2\big(p_0, p^\theta_{1\to 0}\big) \le \left(C_{\mathrm{tr}} + \frac{\widehat{L}^2}{\nu_{\min}}\right)\mathcal{L}_{\mathrm{R}}.$$

For uniform source-time sampling, $\nu(s) = 1$, *this reduces to* $(C_{\mathrm{tr}} + \widehat{L}^2)\mathcal{L}_{\mathrm{R}}$.

A.2 NFM-D Proofs

Here we provide the proofs and derivations for the results stated in §3.3.

Conditional and marginal NFM-D.

Proposition 1 (Conditional NFM-D controls marginal NFM-D). *Assume that* $\mathbb{E}_t[\mathcal{I}(x_0; x_t)] < \infty$, *where* $\mathcal{I}$ *denotes mutual information. Then, for every* θ,

$$\mathcal{L}_{\mathrm{D}} = \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right] + \mathbb{E}_t\left[\mathcal{I}(x_0; x_t)\right] \ge \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right].$$

Consequently, the conditional and marginal objectives share the same minimizers,

$$\arg\min_\theta \mathcal{L}_{\mathrm{D}}(\theta) = \arg\min_\theta \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right].$$

Proof. Fix $t \in [0,1]$. By inserting the marginal density $p_t(x_t)$ into the conditional KL, we obtain

$$\begin{aligned} \mathbb{E}_{x_0}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t(\cdot \mid x_0) \,\|\, p^\theta_{1\to t}\right)\right] &= \mathbb{E}_{x_0,x_t}\left[\log\frac{p_t(x_t \mid x_0)}{p^\theta_{1\to t}(x_t)}\right] \\ &= \mathbb{E}_{x_0,x_t}\left[\log\frac{p_t(x_t \mid x_0)}{p_t(x_t)}\right] + \mathbb{E}_{x_0,x_t}\left[\log\frac{p_t(x_t)}{p^\theta_{1\to t}(x_t)}\right]. \end{aligned} \tag{26}$$

The first term is the mutual information $\mathcal{I}(x_0; x_t)$. Since the second integrand depends only on x_t, marginalizing over x_0 gives

$$\mathbb{E}_{x_0,x_t}\left[\log \frac{p_t(x_t)}{p^\theta_{1\to t}(x_t)}\right] = \mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^\theta_{1\to t}\right).$$

Hence, for every t,

$$\mathbb{E}_{x_0}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t(\cdot \mid x_0) \,\middle\|\, p^\theta_{1\to t}\right)\right] = \mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^\theta_{1\to t}\right) + \mathcal{I}(x_0; x_t).$$

Averaging over the time distribution yields

$$\mathcal{L}_{\mathrm{D}} = \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p^\theta_{1\to t}\right)\right] + \mathbb{E}_t\left[\mathcal{I}(x_0; x_t)\right].$$

The upper bound follows from non-negativity of mutual information. Moreover, $\mathbb{E}_t[\mathcal{I}(x_0; x_t)]$ is determined entirely by the prescribed corruption process and is independent of θ. Whenever it is finite, adding this term therefore does not change the set of minimizers. □

Empirical conditional NFM-D is bounded below.

We next show that the likelihood-form objective in eq. (11) remains bounded below when evaluated on the empirical data distribution. Let

$$\widehat{p}_0 = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i},$$

and consider the linear stochastic interpolant

$$x_t = (1-t)x_0 + tx_1, \qquad x_0 \sim \widehat{p}_0, \qquad x_1 \sim p_1 = \mathcal{N}(0, I).$$

For $t > 0$, its conditional distribution is

$$p_t(\cdot \mid x_0) = \mathcal{N}\left((1-t)x_0, t^2 I\right).$$

Proposition 5 (Empirical conditional NFM-D is bounded below)**.** *Suppose the time-sampling density ν associated with $\mathcal{T}_{\mathrm{S}}$ satisfies*

$$\int_0^1 \nu(t)|\log t|\, dt < \infty.$$

Then the empirical likelihood-form objective corresponding to eq. (11),

$$\widehat{\mathcal{L}}^{\mathrm{lik}}_{\mathrm{D}}(\theta) \coloneqq \mathbb{E}_{t\sim\nu,\, x_0\sim\widehat{p}_0,\, x_t\sim p_t(\cdot|x_0)}\left[\frac{1}{2}\left\|(f^\theta_{1,t})^{-1}(x_t)\right\|^2 - \log\left|\det \frac{\partial (f^\theta_{1,t})^{-1}(x_t)}{\partial x_t}\right|\right]$$

satisfies

$$\widehat{\mathcal{L}}^{\mathrm{lik}}_{\mathrm{D}}(\theta) \geq \frac{d}{2} + d\int_0^1 \nu(t)\log t\, dt > -\infty$$

for every θ. Consequently,

$$\inf_\theta \widehat{\mathcal{L}}^{\mathrm{lik}}_{\mathrm{C},1}(\theta) > -\infty.$$

Proof. For fixed $t > 0$ and x_0, expanding the conditional KL gives

$$\begin{aligned}
&\mathbb{E}_{x_t\sim p_t(\cdot|x_0)}\left[\frac{1}{2}\left\|(f^\theta_{1,t})^{-1}(x_t)\right\|^2 - \log\left|\det \frac{\partial (f^\theta_{1,t})^{-1}(x_t)}{\partial x_t}\right|\right] \\
&\qquad = \mathbb{D}_{\mathrm{KL}}\left(p_t(\cdot \mid x_0) \,\middle\|\, p^\theta_{1\to t}\right) + \mathcal{H}\left(p_t(\cdot \mid x_0)\right) - \frac{d}{2}\log(2\pi). \qquad (27)
\end{aligned}$$

Under the linear interpolant,

$$p_t(\cdot \mid x_0) = \mathcal{N}\left((1-t)x_0, t^2 I\right),$$

and therefore

$$\mathcal{H}\left(p_t(\cdot \mid x_0)\right) = \frac{d}{2}\log(2\pi e t^2).$$

Hence, by non-negativity of the KL divergence,

$$\mathbb{E}_{x_t \sim p_t(\cdot|x_0)}\left[\frac{1}{2}\left\|(f_{1,t}^{\theta})^{-1}(x_t)\right\|^2 - \log\left|\det \frac{\partial (f_{1,t}^{\theta})^{-1}(x_t)}{\partial x_t}\right|\right] \geq \frac{d}{2} + d\log t.$$

Averaging over $x_0 \sim \widehat{p}_0$ and $t \sim \nu$ yields

$$\widehat{\mathcal{L}}_{\mathrm{D}}^{\mathrm{lik}}(\theta) \geq \frac{d}{2} + d\int_0^1 \nu(t)\log t\, dt.$$

The right-hand side is finite under the stated assumption and independent of θ, proving the result. □

For uniform time sampling, $\nu(t) = 1$ on $(0,1)$,

$$\int_0^1 \log t\, dt = -1,$$

so

$$\widehat{\mathcal{L}}_{\mathrm{D}}^{\mathrm{lik}}(\theta) \geq -\frac{d}{2}.$$

Single-time NFM-D endpoint control.

We first establish a slightly more general bound.

Lemma 6 (General endpoint bound)**.** *Let*

$$D_\theta(t) \coloneqq \mathbb{D}_{\mathrm{KL}}\left(p_t \,\middle\|\, p_{1\to t}^{\theta}\right), \qquad \overline{D}_\theta \coloneqq \int_0^1 D_\theta(t)\nu(t)\, dt.$$

Assume that there exist $C < \infty$ and $t_0 > 0$ such that

$$D_\theta(t) \geq D_\theta(0) - Ct, \qquad t \in [0, t_0],$$

and that

$$\nu(t) \geq \nu_{\min} > 0, \qquad t \in [0, t_0].$$

Then

$$D_\theta(0) \leq \inf_{0<h\leq t_0}\left\{\frac{\overline{D}_\theta}{\nu_{\min} h} + \frac{C}{2}h\right\}.$$

Proof. For any $h \in (0, t_0]$,

$$\begin{aligned}
\overline{D}_\theta &= \int_0^1 D_\theta(t)\nu(t)\, dt \\
&\geq \int_0^h D_\theta(t)\nu(t)\, dt \\
&\geq \nu_{\min}\int_0^h \left(D_\theta(0) - Ct\right)\, dt \\
&= \nu_{\min}\left(hD_\theta(0) - \frac{C}{2}h^2\right). \qquad (28)
\end{aligned}$$

Rearranging gives

$$D_\theta(0) \leq \frac{\overline{D}_\theta}{\nu_{\min} h} + \frac{C}{2}h.$$

Taking the infimum over $h \in (0, t_0]$ proves the result. □

Proposition 2 (Single-time NFM-D endpoint control)**.** *Assume $t \mapsto \mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)$ is locally Lipschitz at $t = 0$, with constant C for $t \in [0, t_0]$. Suppose the time-sampling density satisfies $\nu(t) \geq \nu_{\min} > 0$ on $[0, t_0]$. If $\mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right] \leq C\nu_{\min}t_0^2/2$, then*

$$\mathbb{D}_{\mathrm{KL}}\left(p_0 \,\|\, p^\theta_{1\to 0}\right) \leq \sqrt{\frac{2C}{\nu_{\min}}\mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right]}.$$

Proof. The local Lipschitz assumption implies

$$D_\theta(t) \geq D_\theta(0) - Ct, \qquad t \in [0, t_0],$$

so Lemma 6 applies. The unconstrained minimizer of

$$\frac{\overline{D}_\theta}{\nu_{\min}h} + \frac{C}{2}h$$

is

$$h^\star = \sqrt{\frac{2\overline{D}_\theta}{C\nu_{\min}}}.$$

The assumption

$$\overline{D}_\theta \leq \frac{C\nu_{\min}t_0^2}{2}$$

ensures $h^\star \leq t_0$. Substituting $h^\star$ into Lemma 6 yields

$$D_\theta(0) \leq \sqrt{\frac{2C}{\nu_{\min}}\overline{D}_\theta}.$$

Finally, using

$$D_\theta(0) = \mathbb{D}_{\mathrm{KL}}\left(p_0 \,\|\, p^\theta_{1\to 0}\right), \qquad \overline{D}_\theta = \mathbb{E}_{\mathcal{T}_{\mathrm{S}}}\left[\mathbb{D}_{\mathrm{KL}}\left(p_t \,\|\, p^\theta_{1\to t}\right)\right],$$

gives the claimed bound. □

Two-time extension of NFM-D.

For an intermediate source time s, the marginal density p_s is generally intractable. We therefore approximate it using a frozen single-time NFM-D model,

$$\bar{p}_s \coloneqq p^{\bar{\theta}}_{1\to s},$$

where $\bar{\theta}$ is held fixed while optimizing the two-time map $f^\theta_{s,t}$. The corresponding surrogate pushforward is

$$p^\theta_{s\to t} \coloneqq (f^\theta_{s,t})_{\#}\bar{p}_s.$$

For $x_t \sim p_t$, define

$$\widetilde{x}_s \coloneqq (f^\theta_{s,t})^{-1}(x_t).$$

By change of variables,

$$p^\theta_{s\to t}(x_t) = \bar{p}_s(\widetilde{x}_s)\left|\det \frac{\partial (f^\theta_{s,t})^{-1}(x_t)}{\partial x_t}\right|. \tag{29}$$

Since

$$\bar{p}_s = (f^{\bar{\theta}}_{1,s})_{\#}p_1,$$

we can further write

$$\bar{p}_s(\widetilde{x}_s) = p_1\left((f^{\bar{\theta}}_{1,s})^{-1}(\widetilde{x}_s)\right)\left|\det \frac{\partial (f^{\bar{\theta}}_{1,s})^{-1}(x_s)}{\partial x_s}\bigg|_{x_s=\widetilde{x}_s}\right|.$$

Defining

$$\Phi \coloneqq (f^{\bar{\theta}}_{1,s})^{-1} \circ (f^\theta_{s,t})^{-1},$$

we therefore obtain

$$\begin{aligned}\log p^{\theta}_{s\to t}(x_t) = &\log p_1(\Phi(x_t)) \\ &+ \log \left| \det \frac{\partial (f^{\bar{\theta}}_{1,s})^{-1}(x_s)}{\partial x_s} \right|_{x_s=\widetilde{x}_s} \\ &+ \log \left| \det \frac{\partial (f^{\theta}_{s,t})^{-1}(x_t)}{\partial x_t} \right|. \end{aligned} \tag{30}$$

Since $p_1 = \mathcal{N}(0, I)$, the corresponding negative log-likelihood, up to constants independent of θ, is

$$\begin{aligned}\mathcal{L}_{\mathrm{D},2} = \mathbb{E}_{s,t,x_t} \Bigg[&\frac{1}{2} \left\| \Phi(x_t) \right\|^2 \\ &- \log \left| \det \frac{\partial (f^{\bar{\theta}}_{1,s})^{-1}(x_s)}{\partial x_s} \right|_{x_s=\widetilde{x}_s} \\ &- \log \left| \det \frac{\partial (f^{\theta}_{s,t})^{-1}(x_t)}{\partial x_t} \right| \Bigg]. \end{aligned} \tag{31}$$

If the frozen single-time model is exact, such that

$$\bar{p}_s = p_s,$$

then the surrogate pushforward coincides with the ideal pushforward

$$p^{\theta}_{s\to t} = (f^{\theta}_{s,t})_{\#} p_s.$$

In this case,

$$\mathbb{E}_{x_t \sim p_t} \left[-\log p^{\theta}_{s\to t}(x_t) \right] = \mathbb{D}_{\mathrm{KL}} \left(p_t \,\middle\|\, p^{\theta}_{s\to t} \right) + \mathcal{H}(p_t), \tag{32}$$

where $\mathcal{H}(p_t)$ is independent of θ.

When $\bar{p}_s$ only approximates p_s, the frozen single-time model instead supplies a tractable surrogate source density. Although $\bar{\theta}$ is held fixed, gradients with respect to the input of $(f^{\bar{\theta}}_{1,s})^{-1}$ are retained, allowing the loss to update $f^{\theta}_{s,t}$ through the full composition.

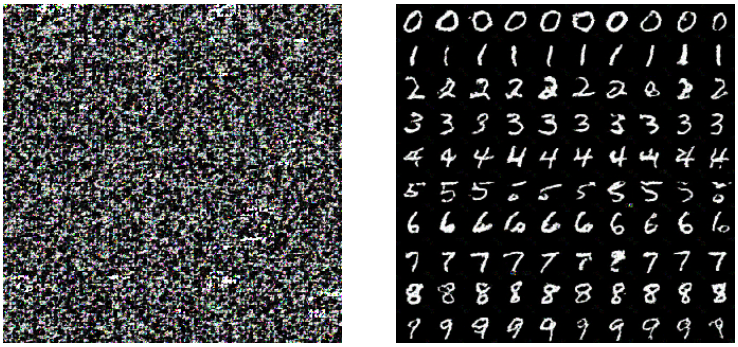
In practice, the single- and two-time maps are evaluations of the same time-conditioned architecture at different time pairs. We jointly optimize

$$\mathcal{L}_{\mathrm{total}}(\theta; \bar{\theta}) = \mathcal{L}_{\mathrm{D}}(\theta) + \lambda \mathcal{L}_{\mathrm{D},2}(\theta; \bar{\theta}),$$

where the single-time objective learns the marginal probability path from the tractable reference p_1, while the two-time objective extends this supervision to transport between intermediate times using the

B EXACT LIKELIHOOD WITHOUT TWEEDIE DENOISING

In Figure 5, we compare guided MNIST samples from an iTARFlow trained via MLE without additive Gaussian noise and an iTARFlow trained with Gaussian noise augmentation and Tweedie denoising at inference time. In this experiment, training without noise produces visibly poor samples, despite retaining the flow's exact invertibility condition. Noise augmentation followed by denoising substantially improves sample quality. However, the flow's tractable density describes its outputs before denoising. The subsequent Tweedie transformation is not guaranteed to be invertible, and its Jacobian determinant is not accounted for by the flow's likelihood calculation. Consequently, the likelihood of the denoised output distribution is not provided by the original flow. This comparison illustrates the practical difficulty, in our setting, of obtaining both high-quality samples and tractable exact likelihoods for the same output distribution.



(a) MLE without noise.

(b) MLE with noise and Tweedie denoising.

Figure 5: Guided MNIST samples from MLE-trained iTARFlow.

C Extended background on Boltzmann Generators

Sampling from equilibrium distributions is a fundamental problem in statistical physics (Frenkel & Smit, 1996; Tuckerman, 2023), since equilibrium configurations provide access to thermodynamic observables and characterize the macroscopic behavior of physical systems (Pathria, 2017). For a configuration x with energy $\mathcal{E}(x)$ at temperature T, the equilibrium distribution is given by the Boltzmann distribution

$$\mu_{\text{target}}(x) = \frac{1}{\mathcal{Z}} \exp\left(-\beta \mathcal{E}(x)\right), \qquad \mathcal{Z} = \int \exp\left(-\beta \mathcal{E}(x)\right) dx, \qquad \beta = \frac{1}{k_B T}, \tag{33}$$

where $\mathcal{Z}$ is the partition function, which is generally intractable for high-dimensional systems.

A particularly important instance of this problem is the sampling of molecular configurations (Zuckerman, 2011). Equilibrium ensembles are required to characterize thermodynamic and conformational properties of molecular systems and are relevant to applications including biomolecular modeling and drug discovery (Frenkel & Smit, 1996). Sampling such systems is challenging because their energy landscapes are typically high-dimensional and multimodal, with metastable regions separated by large free-energy barriers. As a consequence, local sampling methods such as molecular dynamics (MD) (Alder & Wainwright, 1959) and Markov chain Monte Carlo can exhibit long autocorrelation times and slow transitions between metastable states (Wirnsberger et al., 2020; E et al., 2002).

Boltzmann Generators (BGs) (Noé et al., 2019) address this challenge by learning a tractable generative model that approximates the target Boltzmann distribution. In particular, the model defines a distribution $p_\theta(x)$ such that

$$p^\theta(x) \approx \mu_{\text{target}}(x), \tag{34}$$

while providing both a mechanism for efficiently drawing samples $x \sim p^\theta$ and a tractable evaluation of their density $p^\theta(x)$. The former enables direct generation of global proposals in configuration space, while the latter enables the use of Monte Carlo correction methods to account for discrepancies between the learned proposal distribution and the target distribution.

One such correction is self-normalized importance sampling (SNIS) (Liu, 2001). For an observable $\phi(x)$, its expectation under the target distribution can be written as

$$\mathbb{E}_{\mu_{\text{target}}}[\phi(x)] = \frac{\mathbb{E}_{p^\theta}\left[w(x)\phi(x)\right]}{\mathbb{E}_{p^\theta}\left[w(x)\right]}, \qquad w(x) = \frac{\exp\left(-\beta \mathcal{E}(x)\right)}{p^\theta(x)}. \tag{35}$$

The unknown partition function $\mathcal{Z}$ cancels in the ratio. Given K independent samples $x^1, \ldots, x^K \sim p^\theta$, the expectation can be approximated as

$$\mathbb{E}_{\mu_{\text{target}}}[\phi(x)] \approx \sum_{i=1}^{K} \bar{w}^i \phi(x_i), \qquad \bar{w}^i = \frac{w(x^i)}{\sum_{j=1}^{K} w(x^j)}. \tag{36}$$

Beyond importance sampling, the learned model can also be combined with other Monte Carlo correction schemes. In particular, sequential Monte Carlo methods (Tan et al., 2025a) can use the generative model as a proposal while progressively bridging between the learned and target distributions through intermediate distributions and resampling steps. We refer to the appendix of (**?**) for further discussion.

C.1 Baselines

We compare NFM against a broad set of baselines covering both single-system and transferable molecular generative modeling. In the single-peptide setting, we include equivariant CNFs (Klein et al., 2023b; Klein & Noé, 2024) (ECNF++ (Tan et al., 2025a)), RegFlow (Rehman et al., 2026b), flow map based variants, FALCON (Rehman et al., 2026a), F2D2 (Ai et al., 2026), and SCALLOP (OuYang et al., 2026). We further compare against models trained by maximum likelihood, including SBG (Tan et al., 2025a) and ArBG (Rehman et al., 2026c). In the transferable setting, we benchmark against Timewarp (Klein et al., 2023a), BioEmu (Lewis et al., 2025), UniSim (Yu et al., 2025), TarFlow (Zhai et al., 2025; Tan et al., 2025b), Prose (Tan et al., 2025b), and Robin (Rehman et al., 2026c).

C.2 METRICS

We evaluate NFM using three complementary metrics that compare generated samples against reference samples obtained from MD. The energy-based metric probes local geometric fidelity, while the torus- and TICA-based metrics assess larger-scale conformational structure and global distributional agreement.

Energy 2-Wasserstein Distance ($\mathcal{E}-\mathcal{W}_2$). To compare the energy distributions induced by generated and reference configurations, we compute the 2-Wasserstein distance between their respective energy marginals. Let $\rho, \nu \in \mathcal{P}(\mathbb{R})$ denote the distributions of energy values under the generated and reference ensembles, respectively, and let $\Pi(\rho, \nu)$ denote the set of couplings with marginals ρ and ν. We define

$$\mathcal{E}\text{-}\mathcal{W}_2(\rho, \nu) := \left(\inf_{\pi \in \Pi(\rho,\nu)} \int_{\mathbb{R}\times\mathbb{R}} |u - v|^2 \, d\pi(u, v) \right)^{1/2}. \tag{37}$$

This metric measures discrepancies in the distribution of molecular energies and is therefore sensitive to local structural distortions that affect the potential energy, such as changes in bond lengths, bond angles, or short-range interactions.

Torus 2-Wasserstein Distance ($\mathbb{T}$-$\mathcal{W}_2$). To compare the global conformational structure of generated and reference ensembles, we measure discrepancies in the space of backbone dihedral angles. For a peptide with L residues, we represent a configuration x by its vector of dihedral angles

$$\text{Dihedrals}(x) = \big(\phi_1, \psi_1, \ldots, \phi_{L-1}, \psi_{L-1}\big), \tag{38}$$

where ϕ_i and ψ_i denote the backbone torsion angles. Since these variables are periodic, Euclidean distances between angles are not appropriate. We therefore define the squared distance between two configurations using the shortest angular displacement on the torus,

$$c_{\mathbb{T}}(x, y)^2 = \sum_{i=1}^{2(L-1)} \left[\big(\text{Dihedrals}(x)_i - \text{Dihedrals}(y)_i + \pi \big) \bmod 2\pi - \pi \right]^2. \tag{39}$$

Let ρ and ν denote the distributions of backbone dihedral angles induced by the generated and reference ensembles, respectively. Their torus 2-Wasserstein distance is defined as

$$\mathbb{T}\text{-}\mathcal{W}_2(\rho, \nu) = \left(\inf_{\pi \in \Pi(\rho,\nu)} \int c_{\mathbb{T}}(x, y)^2 \, d\pi(x, y) \right)^{1/2}. \tag{40}$$

By comparing configurations in torsional space, this metric captures differences in large-scale conformational structure and is particularly informative about mode coverage of the generated ensemble.

TICA 2-Wasserstein Distance (TICA-$\mathcal{W}_2$). To compare the generated and reference ensembles along their slow collective coordinates, we project configurations onto the leading components obtained by time-lagged independent component analysis (TICA). TICA identifies directions of maximal time-lagged autocorrelation from a reference trajectory.

Let w_1, w_2 denote the first two TICA projection vectors. We define the squared distance between two configurations in TICA space as

$$c_{\text{TICA}}(x, y)^2 = \sum_{j=1}^{2} \big(w_j^\top x - w_j^\top y \big)^2. \tag{41}$$

The corresponding 2-Wasserstein distance between the generated and reference distributions ρ and ν is

$$\text{TICA-}\mathcal{W}_2(\rho, \nu) = \left(\inf_{\pi \in \Pi(\rho,\nu)} \int c_{\text{TICA}}(x, y)^2 \, d\pi(x, y) \right)^{1/2}. \tag{42}$$

This metric measures agreement in the slow conformational subspace and is therefore informative about large-scale metastable structure.

D Implementation details

D.1 Transformer Autoregressive Flows

NFMs are compatible with any NF architecture, however, we focus on Transformer Autoregressive Flows (TARFlows) as they have been established as the state-of-the-art NF architectue in terms of sample fidelity and scalability. TARFlows construct exactly-invertible networks by composing affine transformations parametrized by causal transformers. Except for the first block, each block reverses the sequence, alternating the autoregressive direction. In the interest of concision we omit these permutations below; within block b, entry z_j's updated value is given by:

$$z_j^{b+1} = (z_j^b - \mu_\theta^b(z_{<j}^b)) \odot \exp(-\alpha_\theta^b(z_{<j}^b)) \tag{43}$$

where $\mu_\theta^b(z_j^b)$ and $\alpha_\theta^b(z_{<j}^b)$ are obtained by processing the sequence elements proceeding z_j with a causal transformer. It follows immediately that this transformation may be reversed by autoregressively rolling out,

$$z_j^b = z_j^{b+1} \odot \exp(\alpha_\theta^b(z_{<j}^b)) + \mu_\theta^b(z_{<j}^b) \tag{44}$$

D.2 NFM Architecture

Both NFM-D and NFM-R adopt modified versions of the iTARFlow architecture (Lu et al., 2026b). NFM-D introduces time-conditioning via a sinusoidal timestep embedding module and injects this conditioning along with the class embedding into both the attention and the adaptive layernorm (Peebles & Xie, 2023) modules within each block.

NFM-R adapts iTARFlow to more closely follow the iMF architecture. The major changes are completely eliminating adaptive layer norm modules in favor of in-context conditioning and adopting general-purpose Transformer improvements: SwiGLU (Shazeer, 2020), RMSnorm (Zhang & Sennrich, 2019), RoPE (Su et al., 2024) and QK-normalization.

In-context conditioning. Following iMF we abandon adaLN-zero (Peebles & Xie, 2023) in favor of in-context conditioning with *multiple* tokens for each condition. This change is particularly helpful given the need for our image models to condition on class, two timesteps and several CFG related conditions (c.f. § A.1), something which is better handled by introducing all conditioning into attention than simply summing their embeddings in an adaLN-zero module. Furthermore, adaLN-zero is known to be parameter-heavy (Geng et al., 2026; Lu et al., 2026b); so this change potentially enables NFM-R to be more parameter efficient. Key to making in-context conditioning competitive with adaLN-zero is the use of multiple tokens per condition. Given a condition $\alpha \in \mathbb{R}$, we jointly learn its embedding map $g_\theta : \mathbb{R} \mapsto \mathbb{R}^n$ and m embedding offsets $\{\beta_i\}_{i=1}^m, \beta_i \in \mathbb{R}^n$. Then α's in-context conditioning tokens are given by

$$\{g_\theta(\alpha) + \beta_1, \ldots, g_\theta(\alpha) + \beta_m\} \tag{45}$$

D.3 Algorithmic aspects

D.3.1 Improved Mean Flow objective

The Mean Flow (Geng et al., 2025), improved Mean Flow (Geng et al., 2026) and pixel Mean Flow (Lu et al., 2026a) works all derive their objectives from the Mean Flow identity:

$$v_{s,t}(x_s) = v_s(x_s) + (t-s)\frac{d}{ds}v_{s,t}(x_s), \tag{46}$$

where $v_{s,t}(x_s)$ has the usual definition of average velocity $v_{s,t}(x_s) = \frac{1}{t-s}\int_s^t v_\tau(x_\tau)d\tau$. The identity can be derived by beginning with this definition and applying standard rules of Calculus,

$$(t-s)v_{s,t}(x_s) = \int_s^t v_\tau(x_\tau)d\tau \tag{47}$$

$$\frac{d}{ds}(t-s)v_{s,t}(x_s) = \frac{d}{ds}\int_s^t v_\tau(x_\tau)d\tau \tag{48}$$

$$-v_{s,t}(x_s) + (t-s)\frac{d}{ds}v_{s,t}(x_s) = -v_s(x_s) \tag{49}$$

$$v_{s,t}(x_s) = v_s(x_s) + (t-s)\frac{d}{ds}v_{s,t}(x_s) \tag{50}$$

Expanding $\frac{d}{ds}v_{s,t}(x_s)$ we obtain,

$$v_{s,t}(x_s) = v_s(x_s) + (t-s)(\partial_s v_{s,t}(x_s) + \nabla v_{s,t}(x_s)v_s(x_s)) \tag{51}$$

Equation 51 suggests that we may learn $v_{s,t}(x_s)$ by using stopgrads and matching the RHS. However, when training from scratch we do not have access to the underlying $v_s(x_s)$. In improved Mean Flow the outer $v_s(x_s)$ is replaced by the *sample-conditional* instantaneous velocity[1] and the inner $v_s(x_s)$ by $v^\theta_{s,s}(x)$ following the boundary condition,

$$\mathcal{L}_{\text{iMF}} = \mathbb{E}_{s,t,x_0,x_1}\left[\|(x_1 - x_0) + \text{sg}((t-s)(\partial_s v^\theta_{s,t}(x_s) + \nabla v^\theta_{s,t}(x_s)v^\theta_{s,s}(x_s))) - v^\theta_{s,t}(x_s)\|^2\right] \tag{52}$$

D.3.2 Conditioning and Guidance

NFMs can be made conditional by incorporating additional arguments (e.g., class label). NFM-D follows TARFlow-style (**?**) guidance; training with conditioning-dropout, and at inference performs *online* guidance *directly on the sample* (§ D.3.2). In contrast, NFM-R utilizes iMF-style guidance wherein the average velocity $v_{s,t}$ is conditional on the classifier-free guidance (CFG) interval (Ho & Salimans, 2022; Kynkäänniemi et al., 2024) $[t_{\min}, t_{\max}]$ and guidance strength ω. For training this entails adjusting Equation 3.3 with guidance of varying ω, $t_{\min}$ and $t_{\max}$ (§ D.3.2). At inference time, this enables few-step sampling with flexible CFG settings at no additional cost.

NFM-R guidance. NFM-R utilizes the flexible guidance introduced in iMF. The original Mean Flow (Geng et al., 2025) derived a relationship between the average velocity and the guided instantaneous velocity typically employed in diffusion and flow matching work (Ho & Salimans, 2022). We begin by defining the guided instantaneous velocity for fixed ω:

$$v^{\text{cfg}}_t(x_t|c) \coloneqq \omega v_t(x_t|c) + (1-\omega)v^u(x_t, t), \tag{53}$$

which is a linear combination of class-conditional and class-unconditional instantaneous velocities:

$$v_t(x_t|c) \coloneqq \mathbb{E}_{p(x_0,x_1|x_t,c)}[x_1 - x_0] \text{ and } v^u_t(x_t) \coloneqq \mathbb{E}_c[v_t(x_t|c)] \tag{54}$$

Next, from the MeanFlow Identity (Equation 46), it follows that the guided average velocity satisfies,

$$v^{\text{cfg}}_{s,t}(x_s|c) = v^{\text{cfg}}_s(x_s|c) + (t-s)\frac{d}{ds}v^{\text{cfg}}_{s,t}(x_s|c) \tag{55}$$

Hence, it follows that $v^{\text{cfg}}_{s,s}(x_s|c) = v^{\text{cfg}}_s(x_s|c)$. Next, define $v^{\text{cfg}}_{s,t}(x_s) \coloneqq \mathbb{E}_c[v^{\text{cfg}}_{s,t}(x_s|c)]$. Then by leveraging $v^u_s(x_s) = \mathbb{E}_c[v^{\text{cfg}}_s(x_s|c)]$, we have that $v^{\text{cfg}}_{s,s}(x_s) = v^u_s(x_s)$. Therefore,

[1] We adopt a linear interpolant in this work but this objective supports any stochastic interpolant.

$$v_s^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s|c) = v_s^u(x_s) - v_{s,s}^{\text{cfg}}(x_s) \tag{56}$$
$$v_s^{\text{cfg}}(x_s|c) - v_s^u(x_s) = v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s) \tag{57}$$
$$\omega(v_s(x_s|c) - v_s^u(x_s, s)) = v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s) \tag{58}$$
$$v_s(x_s|c) - v_s^u(x_s, s) = \frac{1}{\omega}(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) \tag{59}$$
$$v_t^u(x_s, s) = v_s(x_s|c) - \frac{1}{\omega}(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) \tag{60}$$
$$v_s^u(x_s, s) + (v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) = v_s(x_s|c) + (1 - \frac{1}{\omega})(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) \tag{61}$$
$$v_s^u(x_s, s) + (v_s^{\text{cfg}}(x_s|c) - v_s^u(x_s, s)) = v_t(x_s|c) + (1 - \frac{1}{\omega})(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) \tag{62}$$
$$v_s^{\text{cfg}}(x_s|c) = v_s(x_s|c) + (1 - \frac{1}{\omega})(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)) \tag{63}$$

Equation 63 allows the guided *marginal* instantaneous velocity to be targeted by utilizing the model's own current estimates of $v_{s,s}^{\text{cfg}}(x_s|c)$ and $v_{s,s}^{\text{cfg}}(x_s)$, where $v_{s,s}^{\text{cfg}}(x_s) \approx v_{s,s}^{\theta}(x_s|\emptyset)$. Furthermore, it can be beneficial to only apply guidance within a specific time interval $[s_{\min}, s_{\max}]$ (Kynkäänniemi et al., 2024),

$$v_s^{\text{cfg}}(x_s|c) = \begin{cases} v_s(x_s|c) + (1 - \frac{1}{\omega})(v_{s,s}^{\text{cfg}}(x_s|c) - v_{s,s}^{\text{cfg}}(x_s)), & s \in [s_{\min}, s_{\max}] \\ v_s(x_s|c), & \text{otherwise.} \end{cases} \tag{64}$$

In order to enable flexible guidance at inference time, iMF makes the network conditional on ω, $s_{\min}$ and $s_{\max}$. During training ω is sampled from a power distribution $\omega \sim p(\omega) \propto \omega^{-\beta}$, $s_{\min} \sim \mathcal{U}[0.0, 0.5]$ and $s_{\max} \sim \mathcal{U}[0.5, 1.0]$. For brevity we use $\Omega \coloneqq \{\omega, s_{\min}, s_{\max}\}$ to denote all the cfg conditioning. From Equations 55 and 63 we can derive

$$v_s^{\text{cfg}}(x_s|c, \Omega) - v_{s,t}^{\text{cfg}}(x_s|c, \Omega) + (t - s)\frac{d}{ds}v_{s,t}^{\text{cfg}}(x_s|c, \Omega) = 0, \tag{65}$$
$$v_s(x_s|c, \Omega) + (1 - \frac{1}{\omega})(v_{s,s}^{\text{cfg}}(x_s|c, \Omega) - v_{s,s}^{\text{cfg}}(x_s)) - v_{s,t}^{\text{cfg}}(x_s|c, \Omega) + (t - s)\frac{d}{ds}v_{s,t}^{\text{cfg}}(x_s|c, \Omega) = 0. \tag{66}$$

Replacing $v_s(x_s|c, \Omega)$ with the *sample-conditional* instantaneous velocity and $v_{s,s}(x_s|c, \Omega)$ as in Appendix A.1 and applying stopgrads we obtain the guided objective,

$$\mathcal{L}_{\text{iMF}}^{\text{cfg}} = \mathbb{E}_{s,t,x_0,x_1,c,\Omega}[\|(x_1 - x_0) + (1 - \frac{1}{\omega})\text{sg}((v_{s,s}^{\theta}(x_s|c, \Omega) - v_{s,s}^{\theta}(x_s)) \tag{67}$$
$$+ \text{sg}((t - s)\partial_s v_{s,t}^{\theta}(x_s|c, \Omega) \tag{68}$$
$$+ (t - s)\nabla v_{s,t}^{\theta}(x_s|c, \Omega)v_{s,s}^{\theta}(x_s|c, \Omega)) \tag{69}$$
$$- v_{s,t}^{\theta}(x_s|c, \Omega)\|^2] \tag{70}$$

NFM-D Guidance. NFM-D utilizes the guidance introduced in TARFlow where the parameters of each affine transformation are a linear combination of guided and unguided μ and α. Given conditioning c, null conditioning $\emptyset$ and guidance strength w, the guided $\tilde{z}_i^t$ (block t, sequence position i) is computed via

$$\tilde{z}_j^b = z_j^{b+1} \odot \exp(\tilde{\alpha}_j^b(\tilde{z}_{<j}^b; c, w)) + \tilde{\mu}_j^b(\tilde{z}_{<j}^b; c, w). \tag{71}$$

Where the guided affine parameters $\tilde{\mu}_i^t$ and $\tilde{\alpha}_i^t$ are given by,

$$\tilde{\mu}_j^b(\tilde{z}_{<j}^b; c, w) = (1 + w)\mu_j^b(\tilde{z}_{<j}^t; c) - w\mu_j^b(\tilde{z}_{<j}^b; \emptyset), \tag{72}$$
$$\tilde{\alpha}_j^b(\tilde{z}_{<j}^b; c, w) = (1 + w)\alpha_j^b(\tilde{z}_{<j}^b; c) - w\alpha_j^b(\tilde{z}_{<j}^b; \emptyset). \tag{73}$$

D.3.3 NUMERICAL INVERTIBILITY

Noise annealing and `FP64` inference. Theoretically, NFM-R should be exactly invertible; however, in practice, its regression objective combined with the denoiser parametrization can lead to a loss of *numerical* invertibility. As a mitigation strategy, we apply additive Gaussian noise to x_0 before computing the interpolant and anneal its contribution to zero over the course of training (see §D.4). Through this strategy, it is possible to recover numerical invertibility in `FP64` at inference. This strategy enabled the NFM-R CIFAR-10 to be finetuned with REINFORCE to preferentially produce green frogs (§ E.1.2).

D.3.4 NFM PARAMETRIZATION (C): GATED

When the base NF is composed of affine coupling transformations as in TARFlows, the identity condition can be enforced while retaining invertibility for every time pair through gating with $(s-t)$,

$$\text{(C)}: \qquad \left[f^{\theta,b}_{s,t}\left(z^b\right)\right]_j = \left(z^b_j - (s-t)\mu^b_\theta\left(z^b_{<j}; s, t\right)\right) \odot \exp\left(-(s-t)\alpha^b_\theta\left(z^b_{<j}; s, t\right)\right). \tag{74}$$

We term this the *gated dual-time NFM*, as each layer i reduces to the identity at $s = t$ while remaining explicitly invertible.

D.4 PSEUDOCODE

Algorithm 1 NFM-R: training loss.
Note: `jvp` returns the function output and its JVP.

```
# net: denoiser NF network
# x,c: training and condition batch


# ReFlow-paired data x0 and noise x1
x0, x1, c = sample_reflow_batch()
sigma = sigma_init
if anneal:
    frac = min((step // steps_per_epoch) / anneal_epochs, 1)
    sigma = (1 - frac) * (sigma_init - sigma_min) + sigma_min
x0 = x0 + sigma * randn_like(x0)

s, t, diag = sample_times()
w, a, b = sample_cfg(diag)
z = (1 - s) * x0 + s * x1
v_pair = x1 - x0

# Average velocity from data prediction
def v_fn(z, s, t, w, c, a, b):
    return (z - net(z, s, t, w, c, a, b)) / s

# Unfiltered and interval-filtered instantaneous velocities
v_c = v_fn(z, s, s, w, c, a, b)
v_u = v_fn(z, s, s, w, None, 0, 1)
w_s = where((a <= s) & (s <= b), w, 1)
v_s = v_fn(z, s, s, w_s, c, 0, 1)
v_target = where(diag, v_c, v_s)
v_g = v_pair + (1 - 1 / w) * (v_target - v_u)

# Class dropout; retain the conditional JVP tangent
n_drop = sum(bernoulli(p_drop, len(c)))
drop = arange(len(c)) < n_drop
c_train = where(drop, null, c)
v_g = where(drop, v_pair, v_g)

# JVP in z and s; hold t, w, c_train, a, b fixed
v, dv_ds = jvp(v_fn, (z, s, t, w, c_train, a, b),(v_s, 1, 0, 0, 0, 0, 0))
V = v + (s - t) * stopgrad(dv_ds)
loss =  metric(V, stopgrad(v_g)) + aux_perceptual_losses(z,s,s,w,c,a,b)
```

Algorithm 2 One-time NFM-D: training loss.
Note: the encoding destination is fixed at $s = 1$ (noise).

```
# net: invertible nf.
# x,c: training and condition batch

# Data batch and independent Gaussian noise
x0, c = sample_batch()
x1 = randn_like(x0)
t = uniform(0, t_max, len(c))
x_t = (1 - s) * x0 + s * x1

# Label dropout for classifier-free guidance
mask = bernoulli(p_drop, len(c))
c = where(mask, null, c)

# Map the interpolant to the noise distribution at s=1.
z, logdet = net.invert(x_t, 1, t, c)

# Gaussian prior term and log-Jacobian correction
loss = 0.5 * mean(z**2) - mean(logdet)
```

Algorithm 3 Two-time NFM-D: training loss.

```
# net: invertible nf.
# x,c: training and condition batch
# gamma: one-time loss weighting

# Data batch and independent Gaussian noise
x0, c = sample_batch()
x1 = randn_like(x0)
s,t = sample_times()
x_s = (1 - s) * x0 + s * x1

# Label dropout for classifier-free guidance
mask = bernoulli(p_drop, len(c))
c = where(mask, null, c)


#Map from s to 1:
z, log_det_s_1 = net.invert(x_s, 1, s, c)
one_time_loss = 0.5 * mean(z**2) - mean(log_det_s_1)

#Map from s to t:
pred_x_t, log_det_s_t = net(x_s, s, t, c)

#Freeze and map t to 1:
frozen_net = freeze(net)
y, log_det_t_1 = frozen_net.invert(pred_x_t, 1, t, c)
two_time_loss = 0.5 * mean(y**2) - mean(log_det_t_1) - mean(log_det_s_t)

loss = gamma*one_time_loss + two_time_loss
```

D.5 Reinforcement Learning Finetuning

We perform *on-policy* reinforcement learning finetuning for both NFM-D and NFM-R. When NFM models are numerically invertible, we are able to train *without* reward gradients and with an exact KL-penalty computed against the untuned model. This is achieved via a KL-penalized REINFORCE with a leave-one-out baseline (RLOO), thus relying only on $\nabla_\theta \log p^\theta_{1\to 0}(x)$:

$$J^{\mathrm{RLOO}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \log p^\theta_{1\to 0}(x_i)\mathrm{sg}(\mathrm{A_i}) \tag{75}$$

$$\text{with } A_i = H_i - \frac{1}{n-1}\sum_{j\neq i} H_j \tag{76}$$

$$\text{and } H_i = R(x_i) - \beta(\log p^\theta_{1\to 0}(x_i) - \log p^\theta_{\mathrm{ref}}(x_i)),\ \beta > 0. \tag{77}$$

When NFM-R becomes numerically non-invertible, the use of the score and KL-penalties becomes infeasible. Instead, we utilize reward gradients and replace the KL-constraint with maximum entropy regularization,

$$J^{\mathrm{maxent}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} R(x_i) - \beta \log p^\theta_{1\to 0}(x_i),\ \beta > 0 \tag{78}$$

$$\nabla_\theta J^{\mathrm{maxent}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta [R(x_i) - \beta \log p^\theta_{1\to 0}(x_i)] \tag{79}$$

CIFAR-10. For our CIFAR-10 RL experiments we train NFM-R and NFM-D to preferentially produce green frogs. We use the $J^{\mathrm{RLOO}}(\theta)$ objective to train the final bias in each iTARFlow block for 5 steps. Noise-annealed NFM-R uses learning rate 2×10^{-2}, $\beta = 1 \times 10^{-6}$ and `FP64` precision and trains without guidance. NFM-D uses learning rate 1×10^{-2}, $\beta = 1$, `FP32` precision and optimizes the guided policy directly. The reward used to incentivize green frogs is:

$$r_{\mathrm{green}}(x) = \mathrm{sigmoid}\left(\frac{\mathrm{g_{score}(x)} - \mu_{\mathrm{g}}}{\sigma_{\mathrm{g}}}\right) \tag{80}$$

where $g_{\mathrm{score}}(x)$ center-crops the image and computes the 'green score' via the pixel average of

$$2 \cdot x_{\mathrm{green_channel}} - x_{\mathrm{red_channel}} - x_{\mathrm{blue_channel}}. \tag{81}$$

Finally, μ_g and σ_g are the empirical average and standard deviation of the green score before post-training. See Figure 8 for before and after.

ImageNet-64. The ImageNet-64 experiments use a mixture of different rewards; NFM-R uses $R(x) = \mathrm{Skywork(x)} + 0.1 \times \mathrm{ImageReward(x)}$ and NFM-D uses $R(x) = \mathrm{ImageReward(x)} + 0.1 \times \mathrm{AestheticScore(x)} + 0.2 \times \mathrm{PickScore(x)}$. NFM-R uses $J^{\mathrm{maxent}}(\theta)$ with $\beta = 0.2$, learning rate 3×10^{-6}, batch size 256, `FP32` precision and trains without guidance for 40 steps. NFM-D uses $J^{\mathrm{RLOO}}(\theta)$ with $\beta = 0.01$, learning rate 1×10^{-6}, batch size 448, `FP32` and trains without guidance for 20 steps. See Figure 2 for before and after, and Figures 3 and 9 for metrics over the course of training.

E Additional Results

Hardware. Experiments were conducted on NVIDIA L40S, H100, GH200, RTX Pro 6000 Blackwell, and RTX Pro 5000 Blackwell.

E.1 Image Generation

Datasets. We evaluate our models on both CIFAR-10 and ImageNet-64×64. For NFM-R we use ReFlow couplings from EDM (Karras et al., 2022) for CIFAR-10 and from EDM 2 (Karras et al., 2024) for ImageNet-64.

Architecture. NFMs implement modified iTARFlow architectures (§ D.2). All model size configurations are based off the original iTARFlow and the NFM-R recipe follows pixel Mean Flow (pMF) (Lu et al., 2026a).

Table 5: Architectural hyperparameters used for the image experiments.

System	Patch Size	Blocks	Layers	Channels	Parameters (M)
CIFAR-10	2	8	6	256	40M
ImageNet-64	2	8	8	384	120M

Training configuration. Following pMF, we use the Muon optimizer with Muon and AdamW weight decays of 0.1 and 1×10^{-4} respectively, and AdamW parameters $(\beta_1, \beta_2) = (0.9, 0.95)$. For MLE iTARFlow we ablated the choice of optimizer Table 7 on CIFAR-10 and used AdamW on both datasets. We train the MLE iTARFlow, NFM-D and noise annealed NFM-R CIFAR-10 models for 960 epochs adopting a constant learning rate of 4×10^{-4}. The noise annealed NFM-R begins with $\sigma = 0.5$ noise injected into the data, and this is annealed to 0 over the first 300 epochs. The numerically non-invertible NFM-R model follows a different training strategy; first we train solely with a flow matching loss (equivalently, only sampling $s = t$ in Algorithm 1) for 450 epochs at learning rate of 4×10^{-4}, next we resume training from EMA weights with the full iMF loss for 200 epochs with the same learning rate. This strategy is substantially cheaper and produces better samples.

Similarly for ImageNet-64, NFM-R begins with flow matching training at a learning rate of 4×10^{-4} for 200 epochs, and resumes training with the full iMF loss for just 25 epochs at a lower learning rate of 1×10^{-4}. On ImageNet-64 both MLE iTARFlow and NFM-D train for 300 epochs with a linear warmup of 25 000 steps up to a learning rate of 4×10^{-4}. For NFM-D we found it beneficial to bring the learning rate down to 1×10^{-4} with a cosine decay from epoch 150 to 200, after the decay this learning rate is maintained until the end.

Evaluation. During training we periodically evaluate the models; for MLE iTARFlow and NFM-D we use a guidance of 2.8 and for NFM-R we use the default guidance settings from pMF. After training, we select the best checkpoint and sweep the guidance setting to find each model's best performance.

Baselines. We primarily compare against iTARFlow (Lu et al., 2026b). To the best of our knowledge, no existing TARFlow works report conditional CIFAR-10 results. Hence, to ensure a fair comparison we ablated improved TARFlow over several learning rates, EMA decays, noise augmentation levels and guidance levels (Table 6). Furthermore, we also investigated whether the iTARFlow recipe could similarly benefit from the Muon optimizer over AdamW (Table 7). Finally, for ImageNet 64×64 we followed the noise level of 0.05 reported in the original TARFlow (Zhai et al., 2025) paper and performed two experiments with learning rates of 1×10^{-4} and 4×10^{-4}.

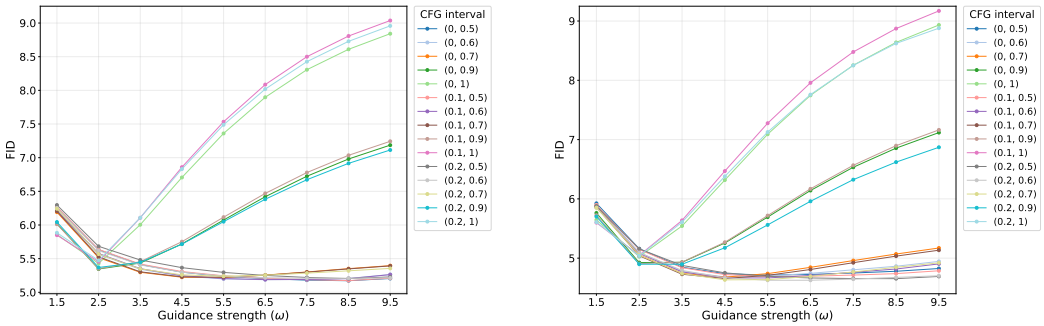
E.1.1 CIFAR10

Table 6: MLE iTARFlow AdamW results on conditional CIFAR-10, results are selected across the learning rates $\{10^{-4}, 4 \times 10^{-4}, 10^{-3}\}$ and EMAs $\{0.9999, 0.9995, 0.999\}$. Results computed with the default MLE iTARFlow guidance of 2.8.

Setting	Variant	FID ↓	FLD ↓	KID ↓	Prec. ↑	Rec. ↑	Invertible? ↓
No noise	Guidance	40.83	23.21	2.758	0.744	0.025	✓
	No guidance	55.52	26.90	4.249	0.835	0.017	✓
0.05 noise	Guidance and Denoise	6.72	8.22	0.457	0.595	0.293	✗
	No guidance	12.04	12.00	1.277	0.627	0.340	✗
	No denoise	48.34	10.05	0.923	0.676	0.122	✓
	No guid., no denoise	71.48	15.16	2.256	0.734	0.129	✓
0.1 noise	Guidance and Denoise	7.59	8.87	0.531	0.598	0.233	✗
	No guidance	13.99	13.22	1.513	0.625	0.273	✗
	No denoise	79.60	12.46	1.778	0.773	0.039	✓
	No guid., no denoise	112.64	18.39	3.886	0.862	0.023	✓
0.2 noise	Guidance and Denoise	11.46	9.89	0.800	0.631	0.159	✗
	No guidance	17.19	14.08	1.782	0.654	0.199	✗
	No denoise	122.25	17.74	4.027	0.886	0.003	✓
	No guid., no denoise	163.44	24.66	6.864	0.948	0.001	✓
0.3 noise	Guidance and Denoise	15.17	11.13	1.008	0.628	0.126	✗
	No guidance	22.22	15.54	2.096	0.658	0.149	✗
	No denoise	165.36	23.71	6.915	0.938	0.000	✓
	No guid., no denoise	198.75	29.67	9.531	0.967	0.000	✓

Table 7: MLE iTARFlow results on conditional CIFAR-10 at 0.05 input noise, comparing the AdamW and Muon optimizers. The AdamW row is selected across the learning rates $\{10^{-4}, 4 \times 10^{-4}, 10^{-3}\}$; the Muon row across the same learning rates and Muon weight decays $\{0, 0.1\}$. Both select over EMAs $\{0.9999, 0.9995, 0.999\}$ and steps by FID. Results computed with the default MLE iTARFlow guidance of 2.8.

Optimizer	Variant	FID ↓	FLD ↓	KID ↓	Prec. ↑	Rec. ↑	Invertible?
AdamW	Guidance and Denoise	6.72	8.22	0.457	0.595	0.293	✗
	No guidance	12.04	12.00	1.277	0.627	0.340	✗
	No denoise	48.34	10.05	0.923	0.676	0.122	✓
	No guid., no denoise	71.48	15.16	2.256	0.734	0.129	✓
Muon	Guidance and Denoise	7.54	7.69	0.394	0.622	0.275	✗
	No guidance	11.62	11.35	1.177	0.630	0.322	✗
	No denoise	44.69	9.34	0.854	0.682	0.116	✓
	No guid., no denoise	66.66	14.18	2.075	0.732	0.121	✓



(a) 1 sampling step.

(b) 2 sampling steps.

Figure 6: NFM-R (*noise annealed*) CIFAR-10 FID as the CFG strength (ω) and the CFG interval $[t_{\min}, t_{\max}]$ vary.

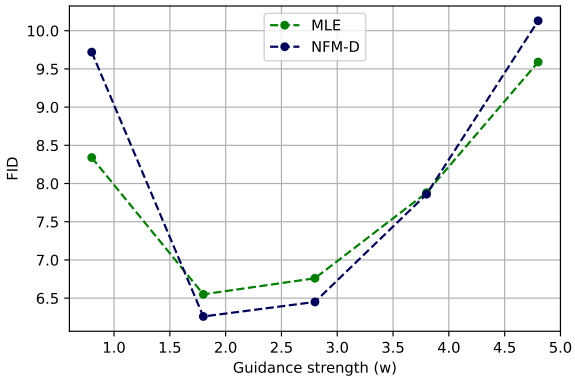


Figure 7: NFM-D and improved TARFlow MLE guidance scale (w) vs FID on CIFAR-10

E.1.2 Reinforcement Learning experiments

CIFAR-10: Frogs experiment.



(a) NFM-D, before. (b) NFM-D, after. (c) NFM-R, before. (d) NFM-R, after.

Figure 8: Samples from NFMs pre- and post-REINFORCE fine-tuning with a reward favouring green frogs.

In Figure 8 we sample frogs under a fixed noise seed before and after fine-tuning. NFM-D demonstrates a clear shift in model outputs, while NFM-R a subtle adjustment.

ImageNet-64: Text-alignment experiments.

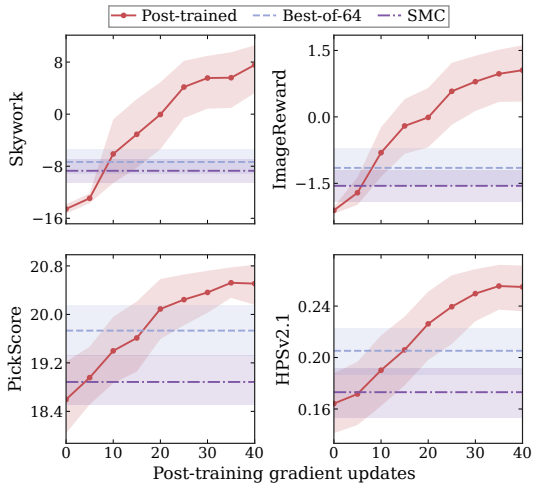


Figure 9: Post-trained NFM-R vs. best-of-64 and SMC baselines. Prompt: *bighorn sheep standing on a sandy beach at the sea*

In Figure 9 we plot the performance of NFM-R against best-of-N and SMC baselines over 10 seeds and observe a clear improvement over these baselines across all four rewards.

E.2 BOLTZMANN GENERATORS

Datasets. We evaluate our models in both system-specific and transferable settings. For the system-specific setting, we consider tri-alanine (AL3), tetra-alanine (AL4), and hexa-alanine (AL6), comprising three, four, and six residues, respectively. We additionally evaluate on Chignolin (sequence GYDPETGTWG), a 10-residue miniprotein. For the transferable setting, we use ManyPeptidesMD.

Architecture. We use the iTARFlow architecture for NFM, with modifications to TARFlow detailed in § D.2. Table 8 reports the configurations used in our experiments. The system-specific configurations follow SBG (Tan et al., 2025a), except that AL4 uses the smaller AL3 configuration and Chignolin uses the smaller AL6 configuration. The transferable configuration follows transferable TARFlow (**?**).

Table 8: Architectural hyperparameters used for NFM.

System	Layers	Blocks	Channels	Parameters (M)
AL3	6	6	256	29.5
AL4	6	6	256	29.5
AL6	6	6	384	66.5
Chignolin	6	6	384	66.5
Transferable	8	8	384	118.6

Training configuration. Following ArBG (Rehman et al., 2026c), we use the Muon optimizer, departing from the original SBG training configuration. We use a learning rate of 2×10^{-4}, Nesterov momentum $\mu = 0.95$, Muon and AdamW weight decays of 0.1 and 1×10^{-4}, respectively, and AdamW parameters $(\beta_1, \beta_2) = (0.9, 0.999)$. We adopt the same cosine-decay learning rate schedule with warmup as SBG (Tan et al., 2025a). For the system-specific models, we sweep over 2,000, 2,500, and 3,000 training epochs, while the transferable model is trained for 1,000 epochs. To improve training throughput, we additionally increase the batch size to 2,048. We ablate these training choices in this section.

E.2.1 SINGLE PEPTIDES

AL3, AL4, and AL6. Table 9 compares NFM-D to different models on AL3, AL4, and AL6. First, NFM-D outperforms SBG. Additionally, it is strong on every baseline, closing the gap to ArBG. We believe there is the need to introduce more single system datasets, including much larger systems (up to 20 residues), that contains equilibrium distributions, as many results are statistically indistinguishable (mostly regarding $\mathbb{T}$-$\mathcal{W}_2$).

Table 9: Benchmark results for continuous flows on AL3, AL4, and AL6. We report ArBG in gray as an autoregressive model reference. Bold values indicate the best result between SBG and NFM-D.

	Tri-alanine (AL3)		Tetrapeptide (AL4)		Hexa-alanine (AL6)	
Algorithm	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$
ArBG (Rehman et al., 2026c)	0.271 ± 0.113	0.311 ± 0.009	0.886 ± 0.076	0.593 ± 0.008	0.722 ± 0.063	1.085 ± 0.069
ECNF++ (Tan et al., 2025a)	2.206 ± 0.813	0.962 ± 0.253	5.638 ± 0.483	1.002 ± 0.061	10.668 ± 0.285	1.902 ± 0.055
RegFlow (Rehman et al., 2026b)	0.853 ± 0.105	1.577 ± 0.140	3.277 ± 0.546	2.342 ± 0.102	—	—
FALCON-A (Rehman et al., 2026a)	1.385 ± 0.182	0.343 ± 0.004	2.929 ± 0.068	1.094 ± 0.034	1.211 ± 0.105	$\mathbf{1.163 \pm 0.112}$
FALCON (Rehman et al., 2026a)	$\underline{0.544 \pm 0.013}$	0.452 ± 0.011	$\mathbf{0.686 \pm 0.047}$	$\mathbf{0.858 \pm 0.077}$	$\underline{0.892 \pm 0.311}$	$\mathbf{1.256 \pm 0.132}$
F2D2 (Ai et al., 2026)	2.155 ± 0.033	$\underline{0.219 \pm 0.001}$	4.209 ± 0.066	1.435 ± 0.023	10.140 ± 0.065	$\mathbf{1.143 \pm 0.039}$
SCALLOP (OuYang et al., 2026)	1.596 ± 0.028	$\mathbf{0.212 \pm 0.001}$	1.545 ± 0.033	1.399 ± 0.039	5.845 ± 0.101	$\mathbf{1.120 \pm 0.014}$
SBG (Tan et al., 2025a)	$\underline{0.598 \pm 0.084}$	0.503 ± 0.029	1.007 ± 0.382	1.039 ± 0.069	1.189 ± 0.357	1.444 ± 0.140
NFM-D (Ours)	$\mathbf{0.166 \pm 0.012}$	0.428 ± 0.025	$\mathbf{0.651 \pm 0.145}$	$\mathbf{0.894 \pm 0.091}$	$\mathbf{0.325 \pm 0.027}$	$\mathbf{1.135 \pm 0.004}$

Finally, we report energy histograms and Ramachandran plots for AL3, AL4 and AL6 in fig. 10, fig. 11, fig. 12, and fig. 13.

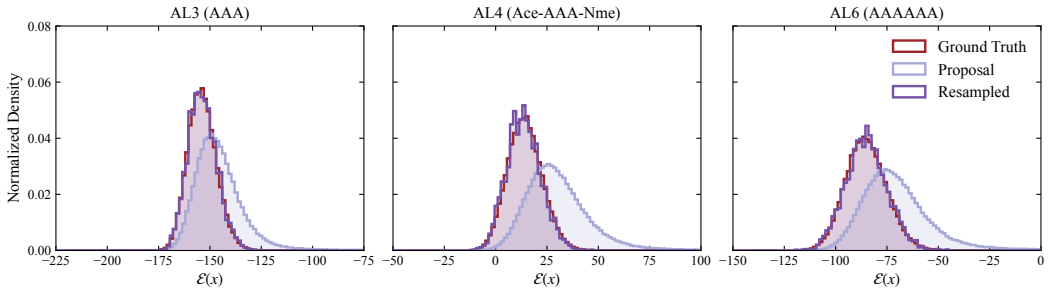


Figure 10: Energy histograms for the true ground truth MD data, the proposal, and the resampled distribution for AL3, AL4, and AL6.

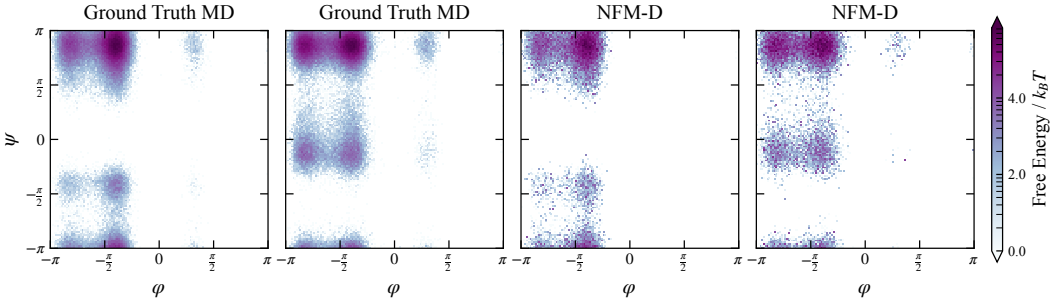


Figure 11: Ramachandran plots for tri-alanine (AL3). Left: reference MD data; right: angular distributions generated by NFM-D.

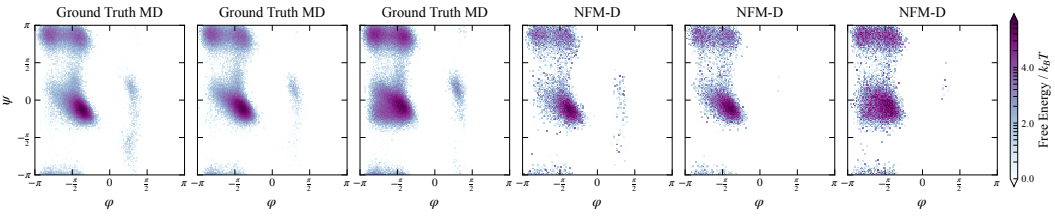


Figure 12: Ramachandran plots for tetra-alanine (AL4). Left: reference MD data; right: angular distributions generated by NFM-D.

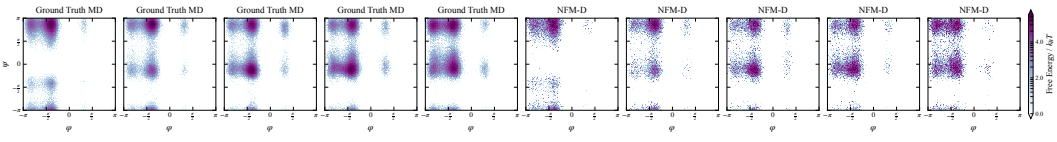


Figure 13: Ramachandran plots for hexa-alanine (AL6). Left: reference MD data; right: angular distributions generated by NFM-D.

Comparison between NFM-D and NFM-R. Table 10 compares NFM-R and NFM-D on AL3. We observe that the raw metrics are comparable, showing that both NFM variants can successfully train NFs. After resampling, however, NFM-D substantially outperforms NFM-R. We attribute this improvement to the better likelihood estimates produced by NFM-D, which enable more effective resampling.

Table 10: Comparison of the best NFM-R and NFM-D models on AL3 using raw and resampled metrics. Best results are in **bold**.

	Raw		Resampled	
Algorithm	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$
NFM-R	$> 10^9$	**0.31**	0.54	0.51
NFM-D	$> 10^9$	0.32	**0.15**	**0.40**

Scaling to decapeptides. Table 11 compares SBG and NFM-D on Chignolin. The effect of the architecture and training setting is ablated. We train a smaller model compared to SBG to avoid overfitting.

Table 11: Comparison of SBG and NFM-D on Chignolin. SBG-iTARFlow refers to our reimplementation of SBG, used to control for architectural, optimizer, and training-recipe changes. Best results are in **bold**; second-best results are underlined.

Algorithm	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$W_2 \downarrow$	Parameters (M)
SBG SMC	10.819 ± 7.206	3.778 ± 0.440	114
SBG-iTARFlow IS	17.015 ± 11.781	2.845 ± 0.069	66.5
SBG-iTARFlow SMC	10.874 ± 10.179	3.558 ± 0.918	66.5
NFM-D IS (Ours)	$\underline{5.118 \pm 0.262}$	$\mathbf{2.712 \pm 0.050}$	66.5
NFM-D SMC (Ours)	$\mathbf{3.419 \pm 0.268}$	$\underline{2.809 \pm 0.119}$	66.5

E.3 Transferable Samplers

Comparison with MLE training. Table 12 reports the effects of different training settings, architectural choices, and loss functions on transferable TARFlow. Comparing the third row with the preceding baselines, we find that training with Muon and a larger batch size yields modest improvements across most metrics. Replacing TARFlow with the iTARFlow architecture leads to a substantially larger gain, while switching to the NFM-D loss further improves the quality of the learned energy representation.

Table 12: Quantitative results on unseen 4- and 8-residue peptides, with a budget of 10^4 energy evaluations at $T = 1$. Rows marked with † denote our reimplementation, used to control for differences in optimizer and training settings. Best results are in **bold**; second-best results are underlined.

Sequence length $\rightarrow$	4AA (30 systems)			8AA (30 systems)		
Model	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	TICA-$\mathcal{W}_2 \downarrow$
MLE-TARFlow (original)	1.260	0.924	0.492	11.298	2.733	1.087
MLE-TARFlow (rerun)	1.414	0.933	0.585	12.126	2.808	1.107
MLE-TARFlow†	1.362	0.843	0.424	12.723	2.654	0.994
MLE-iTARFlow	$\underline{1.068}$	$\underline{0.757}$	**0.378**	$\underline{8.569}$	$\underline{2.379}$	$\underline{0.944}$
NFM-D (Ours)	**0.907**	**0.750**	$\underline{0.387}$	**7.396**	**2.315**	**0.890**

TICA plots for additional unseen octapeptides. We complement Figure 4 by reporting additional TICA plots for unseen octapeptides in Figure 14. This furher demonstrate the zero-shot performance of NFM-D.

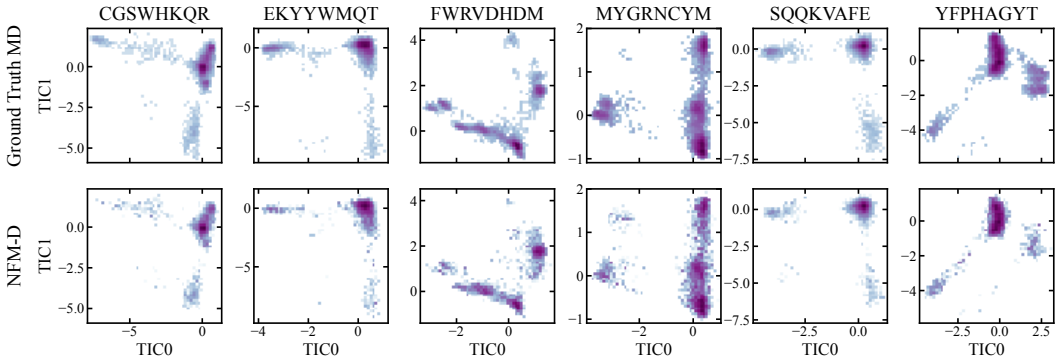


Figure 14: TICA projections for the ground truth MD, and NFM-D on six unseen octapeptides (reported in the titles).

Peptide-level performance. For each 4AA and 8AA peptide in the ManyPeptidesMD test set, we report $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, and TICA-$\mathcal{W}_2$. NFM-D generally achieves lower discrepancies than TARFlow, although the size of the improvement varies across sequences and metrics.

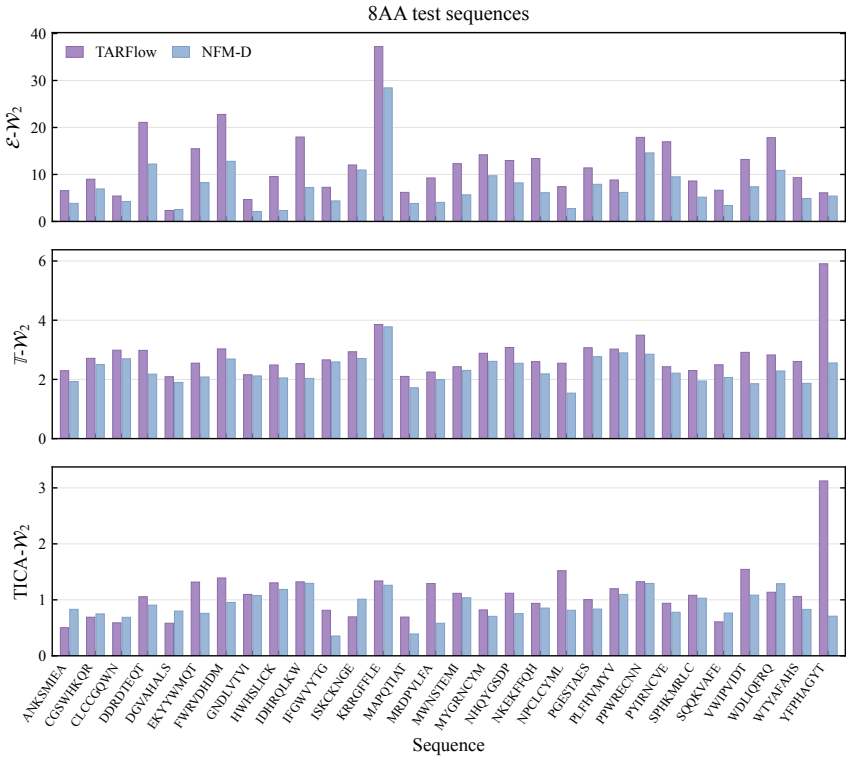


Figure 15: Per-sequence $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, and TICA-$\mathcal{W}_2$ on the 8AA ManyPeptidesMD test set. Bars compare TARFlow and NFM-D; lower values indicate better agreement with the reference distribution.

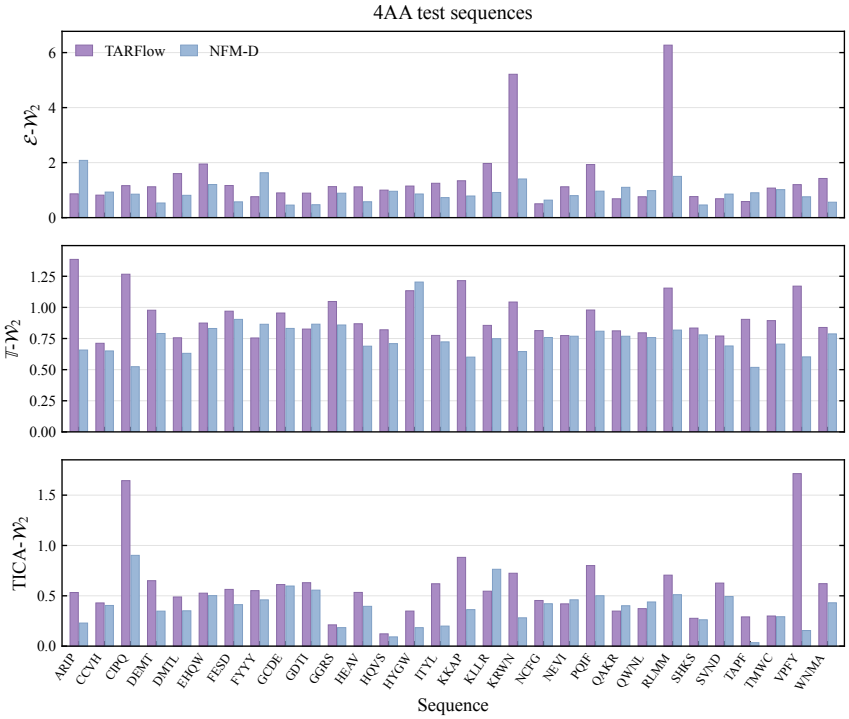


Figure 16: Per-sequence $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, and TICA-$\mathcal{W}_2$ on the 4AA ManyPeptidesMD test set. Bars compare TARFlow and NFM-D; lower values indicate better agreement with the reference distribution.