ENERGY-WEIGHTED FLOW MATCHING: UNLOCKING CONTINUOUS NORMALIZING FLOWS FOR EFFICIENT AND SCALABLE BOLTZMANN SAMPLING

Niclas Dern,* Lennart Redl, Sebastian Pfister, Marcel Kollovieh, David Lüdke & Stephan Günnemann
School of Computation, Information and Technology
Technical University of Munich, Germany

ABSTRACT

Sampling from unnormalized target distributions, e.g. Boltzmann distributions $\mu_{\text{target}}(x) \propto \exp(-E(x)/T)$, is fundamental to many scientific applications yet computationally challenging due to complex, high-dimensional energy landscapes. Existing approaches applying modern generative models to Boltzmann distributions either require large datasets of samples drawn from the target distribution or, when using only energy evaluations for training, cannot efficiently leverage the expressivity of advanced architectures like continuous normalizing flows that have shown promise for molecular sampling. To address these shortcomings, we introduce Energy-Weighted Flow Matching (EWFM), a novel training objective enabling continuous normalizing flows to model Boltzmann distributions using only energy function evaluations. Our objective reformulates conditional flow matching via importance sampling, allowing training with samples from arbitrary proposal distributions. Based on this objective, we develop two algorithms: iterative EWFM (iEWFM), which progressively refines proposals through iterative training, and annealed EWFM (aEWFM), which additionally incorporates temperature annealing for challenging energy landscapes. On benchmark systems, including challenging 55-particle Lennard-Jones clusters, our algorithms demonstrate sample quality competitive with established energy-only methods while requiring up to three orders of magnitude fewer energy evaluations.

1 INTRODUCTION

Understanding the behavior of systems with many interacting particles is a central task in many scientific fields, ranging from molecular dynamics (Allen & Tildesley, 2017; Frenkel & Smit, 2023) to computational chemistry (Shirts & Chodera, 2008; Noé et al., 2019) and protein science (Bryngelson et al., 1995; Dill et al., 2008). In these multi-particle systems, the equilibrium distribution of configurations x (e.g., positions of atoms in a molecule) is often governed by a known energy function $E(x)$, giving rise to a Boltzmann distribution with unnormalized density $\mu_{\text{target}}(x) \propto \exp(-E(x)/T)$, where T is the system's temperature. Generating independent samples from this distribution is essential for computing equilibrium properties, such as the probability of a protein being in a folded state. However, this task remains computationally challenging for complex, high-dimensional systems.

Traditional trajectory-based methods such as Markov Chain Monte Carlo (MCMC) (Hastings, 1970; Andrieu et al., 2003) and molecular dynamics (MD) (Leimkuhler & Matthews, 2013) address this challenge by simulating paths through the system's energy landscape. However, these energy landscapes exhibit numerous local minima (metastable states) separated by high-energy barriers, often causing simulated trajectories to remain trapped within local minima for long periods. This typically leads to prohibitively long simulation times to adequately explore the entire distribution (Allen & Tildesley, 2017; Noé et al., 2019; Łatuszyński et al., 2025; Pompe et al., 2020).

While deep generative models (Lipman et al., 2024; Papamakarios et al., 2021; Yang et al., 2023) offer a modern alternative for learning and sampling from complex distributions, they cannot be di-

*Correspondence to `niclas.dern@tum.de`

rectly applied to Boltzmann distributions as they require training samples from the target distribution — precisely what we seek to generate in the first place. To address this circular problem, Noé et al. (2019) introduced *Boltzmann generators*, a class of deep generative models based on normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016) that primarily leverage the known energy function for training via minimizing the reverse KL divergence between the model and the unnormalized target Boltzmann density. Nevertheless, these methods still require some initial target data to supplement the energy-based training, as the reverse KL divergence alone leads to incomplete target coverage due to its mode-seeking behavior.

This has motivated a line of research into methods that train using only energy function evaluations, without requiring any target samples.[1] Two prominent methods are Flow Annealed Importance Sampling Bootstrap (FAB) (Midgley et al., 2022) and Iterated Denoising Energy Matching (iDEM) (Akhound-Sadegh et al., 2024), though FAB faces scalability challenges for high-dimensional systems and iDEM requires substantial energy evaluations during training.

In parallel, Continuous Normalizing Flows (CNFs) (Chen et al., 2018; Cornish et al., 2020) trained via Flow Matching (Lipman et al., 2022; Albergo et al., 2023) have shown considerable promise for Boltzmann sampling (Klein et al., 2023; Klein & Noé, 2024; Vaitl & Klein, 2025), but current formulations still require large datasets of target samples. This creates a fundamental challenge, as we ideally need both the expressivity of these architectures and the capability of energy-only training.

Contributions. To overcome the reliance on target data for training continuous normalizing flows for Boltzmann sampling, we introduce the Energy-Weighted Flow Matching (EWFM) framework, establishing a new approach for scalable and efficient energy-only Boltzmann generators. Concretely, we make the following contributions:

- We introduce the *EWFM objective* (Sec. 3.1), which reformulates the Conditional Flow Matching (CFM) loss as an expectation over an arbitrary proposal distribution, reweighted by Boltzmann importance weights, enabling CNF training without target samples (see Fig. 1 for a visual comparison).
- Based on this, we develop *iEWFM* (Sec. 3.2), which iteratively refines the proposal using the current model, and *aEWFM* (Sec. 3.3), which additionally incorporates temperature annealing for challenging energy landscapes.
- We demonstrate competitive sample quality on benchmarks including Gaussian mixtures and n-body particle systems up to 55-particle Lennard-Jones (Sec. 4), while requiring orders of magnitude fewer energy evaluations than comparable energy-only methods.

2 Background and preliminaries

We aim to generate i.i.d. samples from a Boltzmann distribution μ_{target} over $\mathbb{R}^d$ defined as:

$$\mu_{\text{target}}(x) = \frac{\exp(-E(x)/T)}{\mathcal{Z}}, \quad \mathcal{Z} = \int_{\mathbb{R}^d} \exp(-E(x)/T)\, dx. \tag{1}$$

Here, $E(x) : \mathbb{R}^d \to \mathbb{R}$ is the energy function of the system, and T is the temperature. The denominator $\mathcal{Z}$ is the partition function, which is generally intractable to compute for high-dimensional systems. Instead, we have access to the energy function $E(x)$ for any configuration x, allowing us to evaluate the unnormalized density $\exp(-E(x)/T)$.

2.1 Boltzmann generators

Boltzmann generators (Noé et al., 2019) are generative models trained to sample from the Boltzmann distribution by learning a transformation from a simple prior to the target, using only the energy function $E(x)$ and temperature T. The original approach trains normalizing flows by minimizing the reverse KL divergence $\text{KL}(q_\theta \| \mu_{\text{target}})$, which can be estimated from model samples alone

[1] The field is currently moving very fast, with multiple recent works improving upon previous methods (see recent work paragraph in Sec. 5).

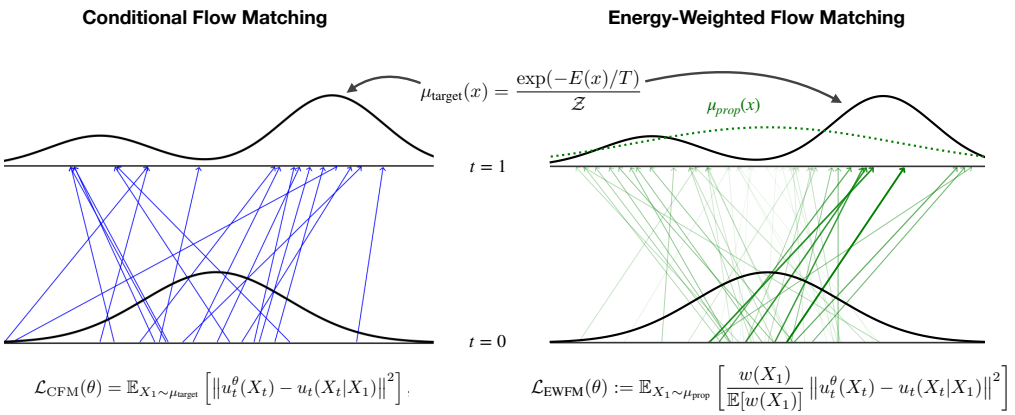

Figure 1: **Conditional Flow Matching vs. Energy-Weighted Flow Matching.** *(Left)* Conditional Flow Matching (CFM) requires samples from the target distribution μ_{target}. The model learns by regressing on points x_t along conditional paths from prior p_0 to target samples. *(Right)* Energy-Weighted Flow Matching (EWFM) reformulates the CFM objective to avoid requiring target samples, instead using an arbitrary proposal distribution. Training points are reweighted by importance weights $w(x_1)$ of their endpoints. High-weight paths (thick lines) are amplified while low-weight paths (thin lines) are suppressed, yielding an equivalent objective that learns the target distribution.

(see Appx. E.1 for details). A key application of trained Boltzmann generators q_θ is computing equilibrium properties via self-normalized importance sampling (SNIS) (Nicoli et al., 2020; Noé et al., 2019), where samples are reweighted with importance weights $w(x) = \exp(-E(x)/T)/q_\theta(x)$.

2.2 Continuous normalizing flows and flow matching

Continuous Normalizing Flows (CNFs) (Chen et al., 2018) define transformations as ODE solutions, removing the requirement for architectural bijectivity inherent to classical normalizing flows. A CNF learns a time-dependent vector field $u_t^\theta(x)$ that induces a flow $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ mapping a base distribution p_0 to a target p_1 via

$$\frac{d}{dt}\psi_t(x) = u_t^\theta(\psi_t(x)), \quad \text{with initial condition} \quad \psi_0(x) = x. \tag{2}$$

The flow defines a probability path $(p_t)_{t\in[0,1]}$ interpolating between p_0 and p_1, and the Instantaneous Change of Variables Formula (Chen et al., 2018) allows exact log-likelihood computation at any generated point. To avoid computationally expensive ODE solving for likelihood computation during training, the flow matching paradigm (Lipman et al., 2022; Albergo et al., 2023) offers an efficient alternative. The core idea is to regress the parameterized vector field u_t^θ onto a target vector field u_t that generates a desired probability path between p_0 and p_1. Although the ideal vector field is generally intractable to compute, this problem is circumvented by instead optimizing the equivalent *conditional* flow matching objective

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,X_t,X_1}\left[\left\|u_t^\theta(X_t) - u_t(X_t|X_1)\right\|^2\right], \text{ where } t \sim U[0,1], X_1 \sim p_1, X_t \sim p_{t|1}(\cdot|X_1). \tag{3}$$

This reduces flow matching to a regression task, enabling simulation-free training. However, this approach requires target samples X_1 — precisely what we lack in Boltzmann sampling.

3 The Energy-Weighted Flow Matching Framework

We now introduce the Energy-Weighted Flow Matching framework, which includes the Energy-Weighted Flow Matching (EWFM) objective that enables CNF training without target samples, and two algorithms that leverage this objective: iterative EWFM and annealed EWFM.

3.1 THE ENERGY-WEIGHTED FLOW MATCHING OBJECTIVE

As established in Sec. 2.2, CFM requires target samples, which are unavailable for Boltzmann sampling. We bridge this gap by reformulating the CFM loss as an expectation over an arbitrary proposal distribution μ_{prop} via importance sampling.

The key insight is that while we cannot sample from the target distribution, we can evaluate its unnormalized density $\exp(-E(x)/T)$. This enables us to rewrite the CFM loss as an expectation over an arbitrary proposal distribution using importance weights. More formally, the conditional flow matching loss can be decomposed as $\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{X_1 \sim \mu_{\text{target}}}[f(X_1; \theta)]$, where $f(x_1; \theta)$ represents the expected loss conditioned on endpoint x_1. By using the relationship

$$\frac{\mu_{\text{target}}(x_1)}{\mu_{\text{prop}}(x_1)} = \frac{w(x_1)}{\mathbb{E}_{X_1' \sim \mu_{\text{prop}}}[w(X_1')]} \tag{4}$$

with unnormalized weights $w(x_1) = \frac{\exp(-E(x_1)/T)}{\mu_{\text{prop}}(x_1)}$, we can take the expectation in the CFM loss with respect to an arbitrary proposal distribution to obtain our Energy-Weighted Flow Matching (EWFM) objective:

$$\begin{aligned} \mathcal{L}_{\text{EWFM}}(\theta; \mu_{\text{prop}}) &:= \mathbb{E}_{t, X_t, X_1}\left[\frac{w(X_1)}{\mathbb{E}_{X_1' \sim \mu_{\text{prop}}}[w(X_1')]} \left\| u_t^\theta(X_t) - u_t(X_t|X_1) \right\|^2\right] \\ &= \mathbb{E}_{X_1 \sim \mu_{\text{prop}}}\left[\frac{w(X_1)}{\mathbb{E}_{X_1' \sim \mu_{\text{prop}}}[w(X_1')]} f(X_1; \theta)\right] = \mathcal{L}_{\text{CFM}}(\theta). \end{aligned} \tag{5}$$

Here, we assume that $t \sim U[0, 1]$, $X_1 \sim \mu_{\text{prop}}$, and $X_t \sim p_{t|1}(\cdot|X_1)$. The mathematical equivalence $\mathcal{L}_{\text{CFM}}(\theta) = \mathcal{L}_{\text{EWFM}}(\theta; \mu_{\text{prop}})$ ensures that minimization of our importance-weighted objective has the same theoretical minimum as the original CFM loss. The derivation is provided in Appx. C.1.

Fig. 1 illustrates the differences between standard CFM and our EWFM approach. A similar reweighting strategy was used by Zhang et al. (2025b) to steer a learned base distribution towards a target proportional to $p_1(x)\exp(-E(x))$ for reinforcement learning applications.

Remark. Crucially, this differs from methods that regress to the marginal score/vector field, such as iDEM (Akhound-Sadegh et al., 2024) or the recently proposed Iterated Energy-based Flow Matching (iEFM) (Woo & Ahn, 2024), which require a nested Monte Carlo importance sampling loop to approximate the marginal score/vector field. Instead, EWFM applies importance sampling directly to the objective. By regressing to the exact conditional vector field, we eliminate the inner estimation loop, reducing the cost to a single energy evaluation per trajectory.

3.2 THE ITERATIVE ENERGY-WEIGHTED FLOW MATCHING ALGORITHM

While the EWFM objective is theoretically sound, its practical estimation via Monte Carlo methods introduces a challenge. If the proposal distribution μ_{prop} differs substantially from the target μ_{target}, the importance weights $w(x)$ will have high variance. This means the Monte Carlo estimate will be dominated by a few samples with extremely large weights, leading to unstable gradients and ineffective training.

To address this, we introduce the *iterative EWFM (iEWFM) algorithm*. The core idea is to use the current generative model q_θ as the proposal distribution for the next training step. We start with an initial proposal distribution (e.g., a simple Gaussian), train a model using EWFM, and then use this trained model as the proposal for the next iteration. Each iteration produces a better approximation to the target distribution, which serves as a higher-quality proposal for subsequent training.

We motivate this iterative strategy through importance sampling theory. The Monte-Carlo estimate of the gradient of the EWFM objective can be written as:

$$\hat{\nabla}_\theta \mathcal{L}_{\text{EWFM}} = \sum_{n=1}^{N} \tilde{w}^{(n)} \phi_\theta(x^{(n)}), \quad \text{where} \quad \tilde{w}^{(n)} = \frac{w(x^{(n)})}{\sum_{m=1}^{N} w(x^{(m)})}. \tag{6}$$

where $\phi_\theta(x_1)$ represents the gradient of the loss conditioned on endpoint x_1. From importance sampling theory, the optimal proposal distribution that minimizes the variance of such estimators

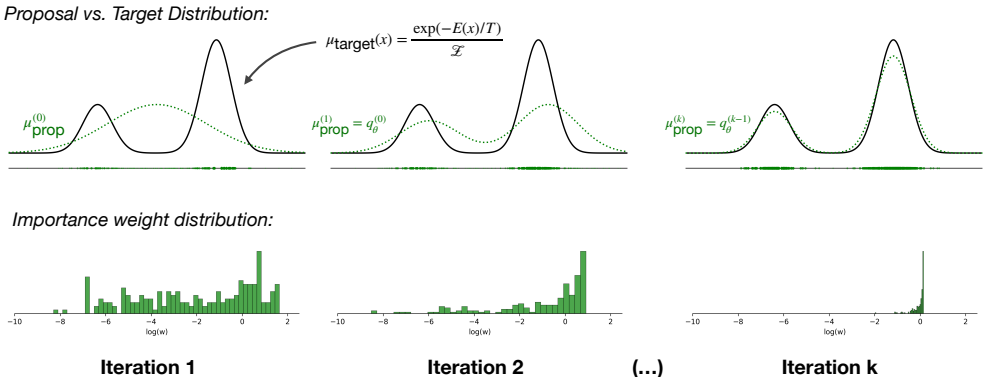

Figure 2: **The iterative EWFM algorithm.** *(Top row)* Shows target distribution μ_{target} (solid black) and proposal distribution μ_{prop} (dotted green) for each iteration, with samples displayed as dots whose size reflects their importance weights. *(Bottom row)* Corresponding distribution of log importance weights. *Iteration 1:* Initial proposal (single Gaussian) poorly matches the target, resulting in highly variable importance weights. *Iteration 2:* Using the previous model as proposal shows improvement — better capturing target modes with more balanced weights. *Iteration k:* After convergence, the proposal closely matches the target, yielding low-variance weights and stable training.

is given by $\mu_{\text{opt}}(x) \propto \mu_{\text{target}}(x) \cdot \|\phi_\theta(x) - \nabla_\theta \mathcal{L}_{\text{EWFM}}\|$. Under the simplifying assumption that the term $\|\phi_\theta(x) - \nabla_\theta \mathcal{L}_{\text{EWFM}}\|$ does not vary substantially across the domain, this reduces to being approximately proportional to the target density (see Appx. C.2 for the complete derivation).

Since our model q_θ is trained to approximate μ_{target}, this motivates our iterative strategy of using the current model as the proposal for the next training step. As training progresses, q_θ becomes a progressively better approximation of the target, leading to lower-variance gradient estimates and more stable training. Fig. 2 illustrates this iterative refinement process. Note that our approach continually refines a single model throughout the iterative training process, rather than training separate models at each iteration.

Amortized training with a sample buffer. Since evaluating the proposal density $q_\theta(x)$ requires solving the reverse-time ODE of the CNF and computing the log-likelihood via the Instantaneous Change of Variables Formula (Chen et al., 2018), which is computationally expensive, we amortize these costs using a sample buffer. We periodically generate samples, pre-compute their log-densities and energies, and reuse them across multiple training steps. This buffering strategy, which has been similarly employed in related work (Midgley et al., 2022; Akhound-Sadegh et al., 2024), preserves the adaptive proposal benefits while improving efficiency. The complete iEWFM algorithm incorporating this buffering approach is presented in Alg. 1, with full implementation details provided in Alg. 2 in the appendix.

3.3 Annealed EWFM: Scaling to Complex Energy Landscapes

While iEWFM provides a robust strategy for the Boltzmann distributions investigated in this work, the quality of the initial proposal can become a limiting factor for more challenging energy landscapes. For such systems, a randomly initialized model forms a poor initial proposal, leading to high-variance gradient estimates that prevent the iterative algorithm from effectively improving. To overcome this bootstrapping problem, we extend iEWFM with temperature annealing. Rather than forcing a randomly initialized proposal to fit a difficult target, we first train at an elevated temperature $T_0 > T$. At this higher temperature, the target distribution $\mu_{T_0}(x) \propto \exp(-E(x)/T_0)$ has flatter energy wells, increasing the likelihood that even a randomly initialized proposal achieves non-negligible overlap with the target, yielding lower-variance importance weights and more stable gradient estimates.

This insight leads to the *annealed Energy-Weighted Flow Matching (aEWFM)* algorithm. We employ a decreasing temperature schedule $T_0 > T_1 > \cdots > T_K = T$, implemented as a geometric

Algorithm 1: Iterative Energy-Weighted Flow Matching (iEWFM) - Simplified

Input: Energy function $E(x)$, temperature T, initial proposal $\mu_{\text{prop}}^{(0)}$
Output: Trained model q_θ
$\mu_{\text{prop}} \leftarrow \mu_{\text{prop}}^{(0)}$; Generate initial buffer $\mathcal{B}$ from μ_{prop};
for *each epoch* **do**
  if *time to refresh buffer* **then**
    $\mu_{\text{prop}} \leftarrow q_\theta$ (use current model as proposal);
    Generate buffer $\mathcal{B}$ from μ_{prop};
  end
  Sample mini-batch $\{x_1^{(n)}\}$ from buffer $\mathcal{B}$;
  Compute importance weights $w^{(n)} = \exp(-E(x_1^{(n)})/T - \log \mu_{\text{prop}}(x_1^{(n)}))$;
  Compute CFM gradients $\hat{\phi}_\theta(x_1^{(n)})$ for each $x_1^{(n)}$;
  Update θ using SNIS gradient $\hat{\nabla}_\theta \mathcal{L}_{\text{EWFM}} = \frac{\sum_n \hat{\phi}_\theta(x_1^{(n)}) w^{(n)}}{\sum_m w^{(m)}}$;
end

Table 1: **Quantitative comparison with established Boltzmann sampling methods.** Results are reported as mean $\pm$ standard deviation over three random seeds. Bold indicates results not significantly outperformed by any other method in the same column ($p < 0.10$, Welch's t-test).

	GMM-40 ($d = 2$)		DW-4 ($d = 8$)		LJ-13 ($d = 39$)		LJ-55 ($d = 165$)	
Method	NLL ↓	$\mathcal{W}_2$ ↓	NLL ↓	$\mathcal{W}_2$ ↓	NLL ↓	$\mathcal{W}_2$ ↓	NLL ↓	$\mathcal{W}_2$ ↓
FAB	7.14 ±0.01	12.00 ±5.73	**7.16** ±0.01	**2.15** ±0.02	**17.52** ±0.17	4.35 ±0.01	200.32 ±62.30	18.03 ±1.21
iDEM	**6.96** ±0.07	**7.42** ±3.44	**7.17** ±0.00	**2.13** ±0.04	**17.68** ±0.14	4.26 ±0.03	125.86 ±18.03	**16.13** ±0.07
EWFM (Ours)	7.05 ±0.05	**3.88** ±0.53	7.76 ±0.09	**2.13** ±0.00	54.19 ±6.05	7.39 ±0.04	-	-
iEWFM (Ours)	7.08 ±0.03	6.68 ±0.67	7.65 ±0.13	2.25 ±0.03	19.38 ±1.27	**4.19** ±0.06	**97.66** ±2.38	16.38 ±0.13
aEWFM (Ours)	7.09 ±0.02	7.06 ±0.49	7.81 ±0.17	2.27 ±0.03	19.41 ±1.11	**4.25** ±0.04	100.89 ±1.46	**16.13** ±0.07

progression. The aEWFM algorithm applies an equivalent iterative scheme to iEWFM by using the model from the previous step (either the previous temperature or the previous iteration at the same temperature) as the proposal.

4 Experimental Results

This section provides an empirical evaluation of our Energy-Weighted Flow Matching framework. We benchmark iEWFM and aEWFM against established methods for Boltzmann sampling without target data, analyzing sample quality, computational efficiency, and the contributions of our algorithmic components.

4.1 Experimental Setup

We evaluate on four benchmarks, namely GMM-40 (2D, 40 components), DW-4 (4-particle double-well, 8D), LJ-13 (13-particle Lennard-Jones, 39D), and LJ-55 (55-particle Lennard-Jones, 165D). The Lennard-Jones systems are particularly challenging due to high dimensionality and sharp, multimodal landscapes. We compare against two established energy-only methods, Flow Annealed Importance Sampling Bootstrap (FAB) (Midgley et al., 2022) and Iterated Denoising Energy Matching (iDEM) (Akhound-Sadegh et al., 2024), as well as EWFM, a simplified variant without iterative refinement. Sample quality is assessed via the 2-Wasserstein Distance ($\mathcal{W}_2$) and Negative Log-Likelihood (NLL), with energy function evaluations as our computational efficiency metric. For iDEM and FAB, we use the quantitative results reported in Akhound-Sadegh et al. (2024). Complete implementation details are provided in Appx. D.1 to D.3.

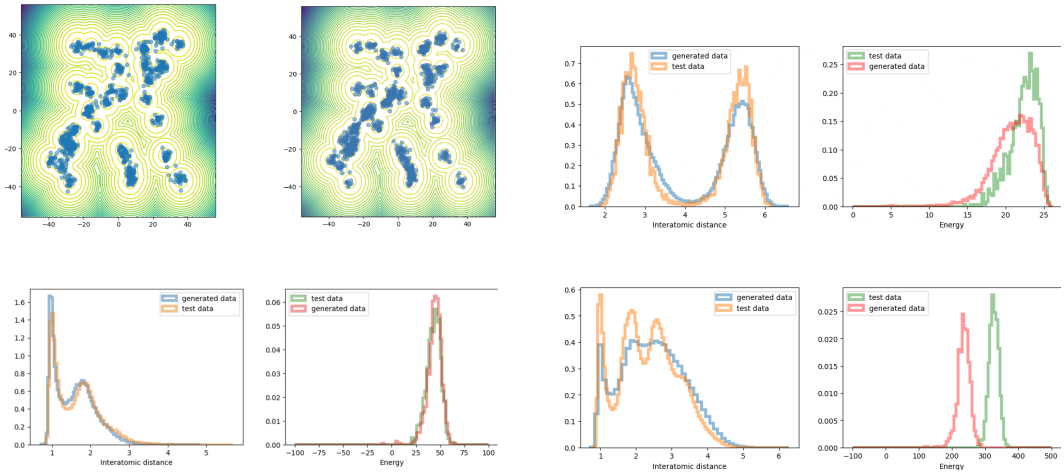

Figure 3: **Sample quality visualization across benchmark systems.** *(Top left)* EWFM samples and *(Top middle)* aEWFM samples for GMM-40, relatively accurately capturing all mixture components. *(Top right)* iEWFM performance on DW-4 showing distributions of interatomic distances and energy values, with limitations in capturing the correct relative weights between peaks. *(Bottom left)* aEWFM on LJ-13 shows excellent agreement with target distributions in both interatomic distance and energy distributions. *(Bottom right)* aEWFM performance on the challenging 165-dimensional LJ-55 system demonstrates relatively good performance despite the system's complexity.

4.2 Results

Table 1 presents our main quantitative results. Our methods achieve competitive performance across benchmarks, with particular strengths on complex systems. On GMM-40, EWFM excels on Wasserstein distance while remaining comparable on NLL, effectively capturing the mixture structure. Performance on DW-4 reveals limitations of our iterative approaches on this intermediate-complexity system, where baselines achieve stronger results. We hypothesize this may be due to potential bias in gradient estimates when using model proposals, though further investigation is needed to confirm this.

For high-dimensional Lennard-Jones systems, our iterative methods demonstrate clear advantages: both iEWFM and aEWFM achieve competitive or superior performance, with particularly strong results on LJ-55, where our methods substantially outperform iDEM on NLL, with aEWFM also matching it on Wasserstein distance. This confirms the effectiveness of our approach on challenging high-dimensional tasks.

The comparison between our variants reveals the importance of the iterative proposal scheme. While EWFM matches or outperforms iEWFM on simple systems (likely benefiting from exact density evaluation of the simple baseline proposal), it fails to converge on LJ-13, demonstrating that iterative refinement becomes essential for complex energy landscapes. Both iEWFM and aEWFM perform similarly across most systems, with aEWFM's robust performance indicating potential for our method to scale to even larger systems.

Table 2: **Energy evaluations required during training.** EWFM variants require fewer energy evaluations than iDEM while being comparable to FAB across benchmark systems.

Method	GMM	DW-4	LJ-13	LJ-55
FAB	1×10^7	1×10^8	6×10^8	6×10^6
iDEM	3×10^{10}	5×10^{10}	5×10^{10}	1×10^9
EWFM variants	3×10^7	3×10^7	1×10^7	1×10^7

Energy Evaluation Efficiency. A key advantage of our method is its energy evaluation efficiency, which is particularly important in real-world applications where energy function evaluations can be computationally expensive. As shown in Table 2, our methods require up to three orders of magnitude fewer energy evaluations than iDEM across all systems. For example, on LJ-13, iDEM requires 5×10^{10} evaluations while our methods need only 1×10^7 — a 5000-fold reduction. This

efficiency likely stems from the EWFM objective providing more informative learning signals per evaluation than Monte-Carlo based objectives, as in iDEM, and our buffer-based strategy reusing samples across training steps.

5 Related Work

Boltzmann generators (Noé et al., 2019) are deep generative models designed to overcome the computational expense of classical sampling methods for equilibrium distributions. Modern approaches can be broadly categorized by their data requirements: methods requiring target samples versus those using only energy evaluations. Among methods requiring target samples, we focus on flow matching approaches, which have recently shown particular promise for Boltzmann sampling. Additional and more detailed discussion is provided in Appx. E.

Flow Matching-Based Methods Requiring Target Data. Recent flow matching approaches for Boltzmann sampling (Klein et al., 2023; Klein & Noé, 2024; Vaitl & Klein, 2025; Yu et al., 2024) have shown promising results for molecular systems. However, all these methods require large datasets of target samples, which is often not feasible in practice.

Methods Using Only Energy Evaluations. Among methods using only energy evaluations, FAB (Midgley et al., 2022) and iDEM (Akhound-Sadegh et al., 2024) are two prominent approaches. FAB combines normalizing flows with annealed importance sampling but faces scalability challenges for high-dimensional systems, while iDEM trains diffusion models via Monte Carlo score estimators, scaling to large systems but requiring substantial energy evaluations. Beyond these, energy-only sampling has also been pursued through stochastic optimal control (Zhang & Chen, 2021; Vargas et al., 2023; Berner et al., 2022), with Richter & Berner (2023) providing a unifying perspective. Further approaches include LFIS (Tian et al., 2024), BNEM (OuYang et al., 2025), iEFM (Woo & Ahn, 2024), NETS (Albergo & Vanden-Eijnden, 2024), Annealing Flow Generative Models (Wu & Xie, 2024), reverse diffusive KL methods (He et al., 2024), SCLD (Chen et al., 2025), and Underdamped Diffusion Bridges (Blessing et al., 2025b). A common theme among those methods is iterative refinement, which our iEWFM algorithm adopts with theoretical motivation from a variance-reduction perspective (Midgley et al., 2022).

Recent Work. Several recent works have made contributions to energy-based sampling. Approaches leveraging stochastic optimal control include Adjoint Sampling (Havens et al., 2025), the Adjoint Schrödinger Bridge Sampler (Liu et al., 2025), and Trust Region Constrained Measure Transport (Blessing et al., 2025a). Methods using annealing strategies include PTSD (Rissanen et al., 2025), PITA (Akhound-Sadegh et al., 2025), and Temperature-Annealed Boltzmann Generators (Schopmans & Friederich, 2025). Other recent contributions include Tilt Matching (Potaptchik et al., 2025), FALCON (Rehman et al., 2025), VT-DIS (Zhang et al., 2025a), and SGDS (Kim et al., 2025). As these methods appeared recently, we leave systematic comparison for future work.

6 Conclusion

We introduced iterative Energy-Weighted Flow Matching (iEWFM) and annealed EWFM (aEWFM), novel methods for training continuous normalizing flows as Boltzmann generators without target samples. Our evaluation demonstrates competitive sample quality compared to established energy-only methods while requiring up to three orders of magnitude fewer energy evaluations. On high-dimensional systems (LJ-13, LJ-55), our methods perform comparably or better than iDEM, suggesting potential for real-world energy landscapes.

Limitations. Several limitations remain. The primary trade-off for our energy evaluation efficiency is the computational cost of CNF density calculations, which dominates wall-clock training time despite our buffering strategy. Additionally, performance gaps on the 8-dimensional DW-4 system compared to FAB and iDEM suggest potential bias in gradient estimates when using the previous model as a proposal distribution, though we lack a full understanding of this behavior.

We discuss future directions and extensions to our framework that we explored in Appx. F.

REFERENCES

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, et al. Iterated denoising energy matching for sampling from boltzmann densities. *arXiv preprint arXiv:2402.06121*, 2024.

Tara Akhound-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. *arXiv preprint arXiv:2506.16471*, 2025.

Michael S Albergo and Eric Vanden-Eijnden. Nets: A non-equilibrium transport sampler. *arXiv preprint arXiv:2410.02711*, 2024.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 2017.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50:5–43, 2003.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

Denis Blessing, Julius Berner, Lorenz Richter, Carles Domingo-Enrich, Yuanqi Du, Arash Vahdat, and Gerhard Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference, 2025a. URL `https://arxiv.org/abs/2508.12511`.

Denis Blessing, Julius Berner, Lorenz Richter, and Gerhard Neumann. Underdamped diffusion bridges with applications to sampling, 2025b. URL `https://arxiv.org/abs/2503.01006`.

Joseph D Bryngelson, José Nelson Onuchic, Nicholas D Socci, and Peter G Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled langevin diffusions, 2025. URL `https://arxiv.org/abs/2412.07081`.

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

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2133–2143. PMLR, jul 2020. URL `https://proceedings.mlr.press/v119/cornish20a.html`.

Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37(1):289–316, 2008.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Aaron Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Brandon Wood, Daniel Levine, Bin Hu, Brandon Amos, Brian Karrer, et al. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. *arXiv preprint arXiv:2504.11713*, 2025.

Jiajun He, Wenlin Chen, Mingtian Zhang, David Barber, and José Miguel Hernández-Lobato. Training neural samplers with reverse diffusive kl divergence. *arXiv preprint arXiv:2410.12456*, 2024.

Minkyu Kim, Kiyoung Seong, Dongyeop Woo, Sungsoo Ahn, and Minsu Kim. On scalable and efficient training of diffusion samplers. *arXiv preprint arXiv:2505.19552*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Leon Klein and Frank Noé. Transferable boltzmann generators. *arXiv preprint arXiv:2406.14426*, 2024.

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

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Krzysztof Łatuszyński, Matthew T Moores, and Timothée Stumpf-Fétizon. Mcmc for multi-modal distributions. *arXiv preprint arXiv:2501.05908*, 2025.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky TQ Chen. Adjoint schr\" odinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025.

Laurence Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Se (3) equivariant augmented coupling flows. *Advances in Neural Information Processing Systems*, 36:79200–79225, 2023.

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.

Kim A Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2):023304, 2020.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

RuiKang OuYang, Bo Qiang, and José Miguel Hernández-Lobato. Bnem: A boltzmann sampler based on bootstrapped noised energy matching, 2025. URL https://arxiv.org/abs/2409.09787.

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

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

Emilia Pompe, Chris Holmes, and Krzysztof Łatuszyński. A framework for adaptive mcmc targeting multimodal distributions. 2020.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S Albergo. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025.

Danyal Rehman, Tara Akhound-Sadegh, Artem Gazizov, Yoshua Bengio, and Alexander Tong. Falcon: Few-step accurate likelihoods for continuous flows. *arXiv preprint arXiv:2512.09914*, 2025.

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

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. *arXiv preprint arXiv:2307.01198*, 2023.

Severi Rissanen, RuiKang OuYang, Jiajun He, Wenlin Chen, Markus Heinonen, Arno Solin, and José Miguel Hernández-Lobato. Progressive tempering sampler with diffusion, 2025. URL `https://arxiv.org/abs/2506.05231`.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Henrik Schopmans and Pascal Friederich. Temperature-annealed boltzmann generators. *arXiv preprint arXiv:2501.19077*, 2025.

Michael R Shirts and John D Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of chemical physics*, 129(12), 2008.

Yifeng Tian, Nishant Panda, and Yen Ting Lin. Liouville flow importance sampler. *arXiv preprint arXiv:2405.06672*, 2024.

Lorenz Vaitl and Leon Klein. Path gradients after flow matching. *arXiv preprint arXiv:2505.10139*, 2025.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. *arXiv preprint arXiv:2302.13834*, 2023.

Dongyeop Woo and Sungsoo Ahn. Iterated energy-based flow matching for sampling from boltzmann densities. *arXiv preprint arXiv:2408.16249*, 2024.

Dongze Wu and Yao Xie. Annealing flow generative models towards sampling high-dimensional and multi-modal distributions. *arXiv preprint arXiv:2409.20547*, 2024.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

Ziyang Yu, Wenbing Huang, and Yang Liu. Force-guided bridge matching for full-atom time-coarsened dynamics of peptides, 2024. URL `https://arxiv.org/abs/2408.15126`.

Fengzhe Zhang, Laurence I Midgley, and José Miguel Hernández-Lobato. Efficient and unbiased sampling from boltzmann distributions via variance-tuned diffusion models. *arXiv preprint arXiv:2505.21005*, 2025a.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. *arXiv preprint arXiv:2503.04975*, 2025b.

APPENDIX OVERVIEW

This appendix provides additional supporting material for the main text. We organize the content as follows:

- **Appx. A:** Reproducibility and ethics statements, including pointers to relevant appendix sections for reproducing our results.
- **Appx. B:** A visual illustration of the fundamental Boltzmann sampling problem.
- **Appx. C:** Technical details for the key components of our Energy-Weighted Flow Matching framework, including the complete mathematical derivation of the EWFM objective, the optimal proposal derivation for iEWFM, the complete iEWFM algorithm, details on the amortized training strategy using sample buffers, and weight clipping techniques for stabilizing training.
- **Appx. D:** More detailed specifications of the benchmark systems used in our evaluation, descriptions of the evaluation metrics employed, the full set of implementation details and hyperparameters for reproducibility, and the computational environment used.
- **Appx. E:** Descriptions of the initial Boltzmann generator framework, Flow Annealed Importance Sampling Bootstrap (FAB), and Iterated Denoising Energy Matching (iDEM).
- **Appx. F:** Future directions and extensions to our framework that we explored: mixture model proposals for more efficient density evaluation, alternative gradient estimation strategies for improved stability, and hybrid approaches that incorporate small amounts of target data when available.
- **Appx. G:** A disclosure of the usage of LLMs during writing and discovery.

A REPRODUCIBILITY AND ETHICS STATEMENTS

Reproducibility. To support reproducibility, we provide a mathematical derivation of our objective in Appx. C.1, detailed benchmark descriptions and metrics in Appx. D.1 and D.2, implementation details including model architecture and temperature annealing schedule in Appx. D.3, hyperparameter choices in Table 3, and computational environment specifications including GPU types and training times in Appx. D.4.

Ethics. We believe there are no significant ethical concerns stemming from our work as it focuses on methodological advances in probabilistic modeling, does not involve human subjects or sensitive data, and poses no immediate societal risks. Nevertheless, we advocate for responsible implementation and use of our methods.

B THE BOLTZMANN SAMPLING PROBLEM

Fig. 4 visualizes the fundamental challenge of Boltzmann sampling. The left panel shows a two-dimensional energy landscape $E(x)/T$ with two distinct low-energy regions separated by a high-energy barrier. The right panel displays the corresponding target Boltzmann distribution $\mu_{\text{target}}(x) \propto \exp(-E(x)/T)$, where probability mass concentrates precisely in these low-energy regions.

This visualization reveals why traditional trajectory-based methods struggle: to transition between the two modes, a sampling trajectory must cross the high-energy barrier, which occurs rarely. Consequently, methods like MCMC often become trapped in one mode for extended periods, failing to adequately explore the full distribution.

C METHODOLOGICAL DETAILS

This section provides technical details for the key components of our Energy-Weighted Flow Matching framework. We begin with the complete mathematical derivation of the EWFM objective, then present the optimal proposal derivation for iEWFM, provide the complete iEWFM algorithm, cover the amortized training strategy using sample buffers, and finally discuss weight clipping techniques for stabilizing training.

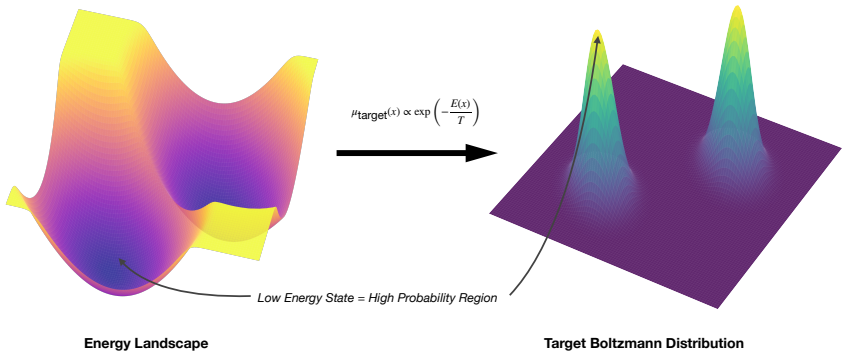

Figure 4: **Illustration of the Boltzmann sampling problem.** *(Left)* A two-dimensional energy landscape $E(x)/T$ with energy values shown in the third dimension, revealing two distinct low-energy regions separated by an energy barrier. *(Right)* The corresponding Boltzmann distribution $\mu_{\text{target}}(x) \propto \exp(-E(x)/T)$, where probability density (shown in the third dimension) is concentrated in low-energy regions. The goal of Boltzmann sampling is to generate samples from this target distribution.

C.1 Detailed Derivation of the EWFM Objective

In the following, we present the step-by-step mathematical derivation that establishes the theoretical foundation for the EWFM objective introduced in Sec. 3.1.

The EWFM objective addresses the fundamental limitation of conditional flow matching (CFM): its reliance on target samples. The problem is that Boltzmann sampling seeks to generate such samples without access to initial samples. In our strategy we use the fact that Boltzmann distributions have a known unnormalized density $\exp(-E(x)/T)$ and use methods usually applied in importance sampling to transform the CFM loss to not require target samples.

Now let $f(x_1;\theta) = \mathbb{E}_{t\sim U[0,1], X_t\sim p_{t|1}(\cdot|x_1)}\left[\left\|u_t^\theta(X_t) - u_t(X_t|x_1)\right\|^2\right]$ denote the expected loss conditioned on an endpoint x_1. The CFM loss can then be written as

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{X_1\sim\mu_{\text{target}}}[f(X_1;\theta)]. \tag{7}$$

The key insight is that $f(x_1;\theta)$ depends only on the endpoint x_1 and not on the distribution from which x_1 is sampled. We can therefore use a method usually applied in importance sampling to transform the expectation from the intractable target distribution to an arbitrary proposal distribution μ_{prop}. Concretely:

$$\mathbb{E}_{X_1\sim\mu_{\text{target}}}[f(X_1;\theta)] = \int f(x_1;\theta)\,\mu_{\text{target}}(x_1)\,dx_1 \tag{8}$$

$$= \int f(x_1;\theta)\,\frac{\mu_{\text{target}}(x_1)}{\mu_{\text{prop}}(x_1)}\,\mu_{\text{prop}}(x_1)\,dx_1 \tag{9}$$

$$= \mathbb{E}_{X_1\sim\mu_{\text{prop}}}\left[\frac{\mu_{\text{target}}(X_1)}{\mu_{\text{prop}}(X_1)}\,f(X_1;\theta)\right]. \tag{10}$$

For this to be well-defined, we need the support condition $\text{supp}(\mu_{\text{target}}) \subseteq \text{supp}(\mu_{\text{prop}})$.

We now express the density ratio in terms of quantities we can evaluate. Substituting the Boltzmann form $\mu_{\text{target}}(x) = \frac{\exp(-E(x)/T)}{\mathcal{Z}}$ yields

$$\frac{\mu_{\text{target}}(X_1)}{\mu_{\text{prop}}(X_1)} = \frac{\exp(-E(X_1)/T)}{\mathcal{Z}\mu_{\text{prop}}(X_1)} = \frac{w(X_1)}{\mathcal{Z}}, \tag{11}$$

where $w(x) := \frac{\exp(-E(x)/T)}{\mu_{\text{prop}}(x)}$. Further, the partition function satisfies $\mathcal{Z} = \int \exp(-E(x)/T)\,dx = \mathbb{E}_{X_1'\sim\mu_{\text{prop}}}[w(X_1')]$ by the same argument as in Eq. (8).

Taken together, this gives us that

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{X_1 \sim \mu_{\text{target}}}[f(X_1;\theta)] = \mathbb{E}_{X_1 \sim \mu_{\text{prop}}}\left[\frac{w(X_1)}{\mathbb{E}_{X_1' \sim \mu_{\text{prop}}}[w(X_1')]} f(X_1;\theta)\right] := \mathcal{L}_{\text{EWFM}}(\theta;\mu_{\text{prop}}). \tag{12}$$

where $t \sim U[0,1]$, $X_1 \sim \mu_{\text{prop}}$, and $X_t \sim p_{t|1}(\cdot|X_1)$. This establishes the mathematical equivalence $\mathcal{L}_{\text{CFM}}(\theta) = \mathcal{L}_{\text{EWFM}}(\theta;\mu_{\text{prop}})$, ensuring that minimization of our importance-weighted objective yields the same model parameters as the original CFM loss. In practice, EWFM requires only sampling from a proposal μ_{prop}, evaluating its density, and computing energies $E(x)$, i.e., avoiding the need for target samples entirely.

C.2 Optimal Proposal Derivation for iEWFM

This section provides the detailed mathematical motivation for the iterative proposal refinement strategy in iEWFM, expanding on the overview presented in Sec. 3.2.

The gradient of the EWFM objective takes the form:

$$\nabla_\theta \mathcal{L}_{\text{EWFM}}(\theta;\mu_{\text{prop}}) = \frac{\mathbb{E}_{X_1 \sim \mu_{\text{prop}}}[\phi_\theta(X_1) w(X_1)]}{\mathbb{E}_{X_1' \sim \mu_{\text{prop}}}[w(X_1')]}, \tag{13}$$

where $\phi_\theta(x_1) = \mathbb{E}_{t \sim U[0,1], X_t \sim p_{t|1}(\cdot|x_1)}\left[\nabla_\theta \|u_t^\theta(X_t) - u_t(X_t|x_1)\|^2\right]$ is the gradient of the loss conditioned on x_1. If the normalization term in the denominator is estimated from the same samples, which is useful for computational efficiency, the gradient estimator takes the form of a self-normalized importance sampling problem. Given N samples $\{x_1^{(n)}\}_{n=1}^N$ from μ_{prop}, the corresponding SNIS estimator is

$$\hat{\nabla}_\theta \mathcal{L}_{\text{EWFM}} = \sum_{n=1}^{N} \tilde{w}^{(n)} \phi_\theta(x_1^{(n)}), \quad \text{where} \quad \tilde{w}^{(n)} = \frac{w(x_1^{(n)})}{\sum_{m=1}^{N} w(x_1^{(m)})}. \tag{14}$$

From importance sampling theory (Owen, 2013), the optimal proposal distribution that minimizes the variance of the SNIS estimator is

$$\mu_{\text{opt}}(x) \propto \mu_{\text{target}}(x) \cdot \|\phi_\theta(x) - \nabla_\theta \mathcal{L}_{\text{EWFM}}\|. \tag{15}$$

While we cannot sample from μ_{opt} directly, as it depends on the gradient we seek, its form suggests an effective approximation strategy. If we make the (relatively strong) simplifying assumption that the difference $\|\phi_\theta(x) - \nabla_\theta \mathcal{L}_{\text{EWFM}}\|$ does not vary substantially across the domain, the optimal proposal becomes approximately the target density, i.e. $\mu_{\text{opt}}(x) \approx \mu_{\text{target}}(x)$. Since our model q_θ is trained to approximate μ_{target}, this motivates using q_θ as the proposal distribution, forming the theoretical motivation for the iterative refinement strategy.

C.3 Amortized Training with Sample Buffer

A direct implementation of the iterative scheme would be computationally expensive, as evaluating the proposal density $q_\theta(x)$ for each new sample during each gradient step requires solving the reverse-time ODE of the CNF, and computing the corresponding importance weights requires a new energy evaluation. To make this practical, we amortize these costs using a *sample buffer*.

The buffering strategy works as follows: periodically, we generate a fixed set of N_{buffer} samples and pre-compute both their log-densities under the current proposal q_θ and their energies $E(x)$. These samples, log-densities, and energy values are cached in a buffer $\mathcal{B} = \{x_j, \log q_\theta(x_j), E(x_j)\}_{j=1}^{N_{\text{buffer}}}$. For multiple subsequent training steps, we then use this static buffer as our proposal distribution by drawing mini-batches from it. By reusing samples multiple times, this buffering approach reduces computational cost and the number of energy evaluations approximately by the average number of times each sample is reused.

The use of a buffer introduces several hyperparameters that represent trade-offs between computational cost and statistical accuracy. The buffer size N_{buffer} and mini-batch size N_{batch} control the approximation quality and per-step computational cost, while the buffer refresh rate determines how

Algorithm 2: Iterative Energy-Weighted Flow Matching (iEWFM) - Detailed

Input: Energy function $E(x)$; target temperature T; initial parameters θ; total epochs E_{total}; epochs per buffer refresh E_{refresh}; buffer size N_{buffer}; batch size N_{batch}; initial proposal $\mu_{\text{prop}}^{(0)}$; prior p_0.
Output: Final trained model parameters θ.
$\mu_{\text{prop}} \leftarrow \mu_{\text{prop}}^{(0)}$;
Generate initial buffer $\mathcal{B}$ by sampling $\{x_j, \log \mu_{\text{prop}}(x_j), E(x_j)\}_{j=1}^{N_{\text{buffer}}}$ from $\mu_{\text{prop}}^{(0)}$;
for $e = 1$ **to** E_{total} **do**
  if $(e-1) \pmod{E_{refresh}} == 0$ ***and*** $e > 1$ **then**
    $\mu_{\text{prop}} \leftarrow q_\theta$;
    Generate buffer $\mathcal{B}$ by sampling $\{x_j, \log \mu_{\text{prop}}(x_j), E(x_j)\}_{j=1}^{N_{\text{buffer}}}$ from μ_{prop};
  end
  Sample mini-batch $\{(x_1^{(n)}, \log \mu_{\text{prop}}(x_1^{(n)}))\}_{n=1}^{N_{\text{batch}}}$ from $\mathcal{B}$;
  Compute importance weights $w^{(n)} = \exp(-E(x_1^{(n)})/T - \log \mu_{\text{prop}}(x_1^{(n)}))$ for $n = 1, \ldots, N_{\text{batch}}$;
  for $n = 1$ **to** N_{batch} **do**
    Sample $t \sim \mathcal{U}(0,1), x_0 \sim p_0$;
    $x_t \leftarrow (1-t)x_0 + t x_1^{(n)}$;
    $\hat{\phi}_\theta(x_1^{(n)}) \leftarrow \nabla_\theta \left\| u_t^\theta(x_t) - u_t(x_t | x_1^{(n)}) \right\|^2$;
  end
  Estimate full gradient $\hat{\nabla}_\theta \mathcal{L}_{\text{EWFM}} = \frac{\sum_{n=1}^{N_{\text{batch}}} \hat{\phi}_\theta(x_1^{(n)}) w^{(n)}}{\sum_{m=1}^{N_{\text{batch}}} w^{(m)}}$;
  Update parameters $\theta \leftarrow \text{OptimizerUpdate}(\theta, \hat{\nabla}_\theta \mathcal{L}_{\text{EWFM}})$;
end
return θ;

frequently the buffer is updated to remain consistent with the changing model q_θ. We are currently sampling with replacement from the buffer, but we have also experimented with sampling without replacement, which did not seem to improve performance.

This buffering approach is similar to strategies employed in related work (Midgley et al., 2022; Akhound-Sadegh et al., 2024). The specific hyperparameter values used in our experiments are detailed in Appx. D. You can find the full iEWFM algorithm, including the buffering strategy, in Alg. 2.

C.4 Stabilizing Training with Weight Clipping

To further mitigate the issue of the SNIS gradient estimator becoming unstable when sampling from distributions with sharp energy landscapes, we investigated weight clipping strategies to cap the influence of samples with very high importance weights.

We investigated two different clipping strategies to address this issue. The first approach directly clips the negative energy values from above:

$$w_{\text{clipped}}^{(1)}(x) = \frac{\exp\left(\min\left(-E(x)/T, \tau_1\right)\right)}{\log \mu_{\text{prop}}(x)} = \exp\left(\min\left(-E(x)/T, \tau_1\right) - \log \mu_{\text{prop}}(x)\right), \tag{16}$$

while the second approach clips the combined log-importance weight term from above:

$$w_{\text{clipped}}^{(2)}(x) = \exp\left(\min\left(-E(x)/T - \log \mu_{\text{prop}}(x), \tau_2\right)\right), \tag{17}$$

where τ_1 and τ_2 are set to high percentile thresholds (e.g., the 99th percentile) of their respective unclipped terms, preventing the largest importance weights from dominating the gradient estimates. Based on our results during hyperparameter tuning, we adopted the second approach as it performs better, especially when the proposal distribution's likelihood evaluations are less reliable due to the Hutchinson trace estimator used for density evaluation in CNFs.

D EXPERIMENTAL DETAILS

To ensure comparability with other recent work, our experimental setup generally follows that of Akhound-Sadegh et al. (2024), employing benchmark systems and metrics commonly used in recent literature on generative Boltzmann sampling. For the performance of the baselines (both iDEM and FAB), we directly cite results from Akhound-Sadegh et al. (2024). We re-implemented an equivalent evaluation pipeline to Akhound-Sadegh et al. (2024) to evaluate our models.

The following subsections provide detailed information on the benchmark systems utilized, the evaluation metrics employed, and the hyperparameters used in our experiments.

D.1 BENCHMARK SYSTEMS

We evaluate our methods on four classical benchmark systems for Boltzmann generators, covering a range of complexities and dimensionalities. For more detailed descriptions, see Akhound-Sadegh et al. (2024).

GMM-40 This represents a two-dimensional Gaussian Mixture Model with 40 components arranged on a grid (Midgley et al., 2022). The energy function is the negative log-probability of the mixture: $E(x) = -\log\left(\sum_{i=1}^{40} \frac{1}{40}\mathcal{N}(x|\mu_i, \Sigma_i)\right)$. Despite its low dimensionality, this system challenges models to capture multiple distinct modes.

DW-4 This system describes four particles in a 2-dimensional space (8D total dimensions) interacting via a double-well potential (Köhler et al., 2020). Following previous work, we set the parameters to $a = 0$, $b = -4$, $c = 0.9$, $d_0 = 4$, and $\tau = 1$. For evaluation, we use ground truth data from Klein et al. (2023).

LJ-13 and LJ-55 These systems feature clusters of particles governed by the Lennard-Jones potential with harmonic confinement, modeling attractive and repulsive forces between particles. LJ-13 involves 13 particles (39D total), and LJ-55 involves 55 particles (165D total). These systems are particularly challenging due to their high dimensionality and sharp, multi-modal energy landscapes. We use MCMC samples from Klein et al. (2023) as ground truth.

D.2 EVALUATION METRICS

Sample quality is assessed using two complementary metrics that capture different aspects of distributional similarity:

2-Wasserstein Distance ($\mathcal{W}_2$) The $\mathcal{W}_2$ distance quantifies the minimum cost to transform one probability distribution into another:

$$\mathcal{W}_2(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \, d\pi(x, y)\right)^{1/2} \tag{18}$$

where $\Pi(\mu, \nu)$ is the set of all joint distributions with marginals μ and ν. We estimate this distance by computing the Wasserstein distance between the empirical distributions of generated and ground truth samples using the Python Optimal Transport package (Flamary et al., 2021) with Euclidean distance. Lower $\mathcal{W}_2$ values indicate a closer distributional match.

Negative Log-Likelihood (NLL) The NLL measures how likely a set of ground truth test samples is under the learned generative model. Following the evaluation pipeline from Akhound-Sadegh et al. (2024), we first generate a large dataset of samples from our trained model (100,000 for GMM-40, DW-4, and LJ-13; 10,000 for LJ-55). We then train a separate evaluation CNF on these generated samples. The NLL is then the negative log-probability of the ground truth test data under this evaluation CNF, computed using the Instantaneous Change of Variables Formula. For the GMM-40, DW-4, and LJ-13 tasks, we use exact computation of the divergence term, while for the high-dimensional LJ-55 system, we use the Hutchinson trace estimator. Lower NLL values indicate better sample quality.

While we employ this evaluation pipeline for comparability with Akhound-Sadegh et al. (2024), it has limitations: the optimal checkpoint for the evaluation CNF is selected using a validation set of samples from the target distribution, and we also found that there were slight discrepancies between NLL values under our original model versus the evaluation model. We note that we do not find this evaluation procedure optimal and plan to use different metrics that do not require training a second model in future versions of this paper.

Omission of Effective Sample Size (ESS) We omit the Effective Sample Size metric used in Akhound-Sadegh et al. (2024) as we found it to be relatively unstable during our evaluations. Specifically, the ESS values varied significantly depending on whether we used the density from our original model or the density from the evaluation CNF model, and also showed considerable variation between different runs. Additionally, we note that Akhound-Sadegh et al. (2024) evaluated the ESS on only 16 test samples for all tasks, which may not provide reliable estimates.

Energy Function Evaluations As energy evaluations can be computationally expensive for large systems (Klein & Noé, 2024; Havens et al., 2025), we report the total number of energy function evaluations during training as an important efficiency metric.

D.3 Implementation Details and Hyperparameters

Baselines We compare our algorithms against two state-of-the-art methods for energy-only Boltzmann sampling: Flow Annealed Importance Sampling Bootstrap (FAB) (Midgley et al., 2022), which combines normalizing flows with Annealed Importance Sampling, and iterated Denoising Energy Matching (iDEM) (Akhound-Sadegh et al., 2024), which employs a score-matching approach. Both methods enable training Boltzmann generators without target data. For both baselines, we report performance metrics from Akhound-Sadegh et al. (2024).

In addition to our main proposed methods (iEWFM and aEWFM), we also evaluate a simplified variant we call EWFM, which represents an ablation of iEWFM without the iterative refinement component. Instead of using the current model as a proposal, EWFM employs a simple, fixed proposal distribution throughout training (e.g., a standard Gaussian). This allows us to assess the specific contribution of the iterative scheme to the overall performance.

Model Architectures To ensure direct comparability with baseline results, we use the same network architectures as Akhound-Sadegh et al. (2024) used for score estimation to parameterize our vector fields: a multi-layer perceptron (MLP) with sinusoidal positional embeddings for the GMM task, and an Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021) for the DW-4, LJ-13, and LJ-55 systems. All models were optimized using the Adam optimizer (Kingma & Ba, 2014).

Training Details We employ weight clipping on the top percentiles of importance weights to stabilize training, with clipping percentiles ranging from 97.5% to 99.9%. We found that the choice of clipping percentile was often one of the most impactful hyperparameters for training stability and final performance. When training our models with either iEWFM or aEWFM, we evaluate model densities using the Hutchinson trace estimator for the divergence calculations, which leads to imperfect density estimates but is required to keep training computationally tractable. For evaluation, we use the final checkpoint at the end of training without any model selection based on validation performance.

For aEWFM, we employ a geometric temperature annealing schedule starting from $T_{\text{init}} = 10.0$ and decreasing every 2 epochs until reaching the target temperature of 1.0, using a total of 100 epochs to anneal across all systems. The buffer refresh rate is set to once per epoch across all experiments, though for the more challenging LJ systems, we reduced the refresh rate and increased the number of mini-batches per epoch to save computational cost. Hyperparameters were chosen through a structured search over key parameters, including learning rates, buffer sizes, and clipping percentiles.

Table 3: **Key hyperparameters across the various benchmark systems.** The main hyperparameters (learning rate through mini-batches per epoch) are used identically for EWFM, iEWFM, and aEWFM. The annealing schedule parameters are only used for aEWFM, while iEWFM and EWFM use a fixed temperature of 1.0 throughout training. For DW-4, the clipping percentile notation 97.5%/99.9% indicates that 97.5% was used for aEWFM and iEWFM while 99.9% was used for EWFM.

Hyperparameter	**GMM-40**	**DW-4**	**LJ-13**	**LJ-55**
Learning Rate	5×10^{-4}	1×10^{-3}	5×10^{-4}	5×10^{-4}
Buffer Size (N_{buffer})	5000	5000	5000	500
Batch Size (N_{batch})	5000	5000	5000	500
Total Training Epochs	5000	2500	2500	2500
Mini-batches per Epoch	10	10	20	20
Annealing Schedule (for aEWFM)				
Initial Temperature (T_{init})	10.0	10.0	10.0	10.0
Epochs per Temperature	2	2	2	2
Total Annealing Epochs	100	100	100	100
Weight Clipping				
Clipping Percentile	99.9%	97.5%/99.9%	99.9%	98.0%

D.4 Computational Environment

Experiments were conducted on a variety of NVIDIA GPUs. The computationally intensive LJ-55 experiments utilized an H100 GPU. The LJ-13 experiments were performed on an A100 GPU, while the smaller GMM-40 and DW-4 systems were trained on a GTX 1080 Ti GPU.

For reference, the training times were as follows: LJ-55 (H100) took 25-27 hours, LJ-13 (A100) took 30-31 hours, DW-4 (GTX 1080 Ti) took 10 hours, and GMM-40 (GTX 1080 Ti) took 5 hours. We note that both wall-clock time and energy evaluation counts could likely be improved further by reducing the number of training epochs, as convergence was often achieved in approximately one-half of the total training time, with only marginal improvements observed in later stages.

E Further Details on Related Work

This section provides extended descriptions of the methods discussed in Sec. 5, including the initial Boltzmann generator framework, flow matching-based methods requiring target data, energy-only methods (FAB, iDEM, and others).

E.1 The Initial Boltzmann Generator Method

The core contribution of Boltzmann generators, as introduced by Noé et al. (2019), is to adapt the training of generative models to the setting where only the energy function $E(x)$ and temperature T are known. The initial training objective is to minimize the reverse KL divergence, $\text{KL}(q_\theta \| \mu_{\text{target}})$, which, unlike the forward KL, can be estimated using samples from the model q_θ itself:

$$\begin{aligned}\text{KL}(q_\theta \| \mu_{\text{target}}) &= \int q_\theta(x) \log \frac{q_\theta(x)}{\mu_{\text{target}}(x)} dx \\ &= \mathbb{E}_{X \sim q_\theta} \left[\log q_\theta(X) + E(X)/T\right] + \log \mathcal{Z}. \end{aligned} \tag{19}$$

Since $\log \mathcal{Z}$ is constant with respect to the model parameters θ, it can be ignored during optimization. Note that the reverse KL divergence has the known limitation of being "mode-seeking", meaning that optimizing it can lead to the model collapsing to only a subset of the modes of a multi-modal target distribution. To stabilize training, Noé et al. (2019) use a combination of the forward and reverse KL divergences:

$$\lambda \text{KL}(\mu_{\text{target}} \| q_\theta) + (1 - \lambda) \text{KL}(q_\theta \| \mu_{\text{target}}) \tag{20}$$

where λ is a hyperparameter, which is often annealed during training. Here, the forward KL term is estimated using a small set of training samples from the target distribution.

E.2 Flow Annealed Importance Sampling Bootstrap (FAB)

Flow Annealed Importance Sampling Bootstrap (FAB) (Midgley et al., 2022; 2023) augments a normalizing flow q_θ with an Annealed Importance Sampling (AIS) bootstrap to minimize the mass-covering α-divergence (with $\alpha = 2$) between the target μ_{target} and the flow:

$$\mathcal{D}_2\big(\mu_{\text{target}} \parallel q_\theta\big) \;=\; \tfrac{1}{2}\int_{\mathbb{R}^d} \frac{\mu_{\text{target}}(x)^2}{q_\theta(x)}\,\mathrm{d}x, \tag{21}$$

whose minimizer yields the lowest possible variance of importance weights. This choice of divergence addresses the mode-seeking behavior of the reverse KL divergence used in earlier methods.

During training, AIS is run with the current flow as the initial distribution and a target density $g(x) \propto \mu_{\text{target}}(x)^2/q_\theta(x)$; this makes the AIS path focus on regions that contribute most to $\mathcal{D}_2$, similar to how our iterative approach uses the current model as the proposal for the next iteration to minimize the variance of the estimator. Each AIS run returns pairs $\{(x, w_{\text{AIS}})\}_{i=1}^N$, used to update the flow through a self-normalized surrogate loss:

$$\mathcal{S}'(\theta) \;=\; -\,\mathbb{E}_{\text{AIS}}\big[\bar{w}_{\text{AIS}}\,\log q_\theta(x)\big], \qquad \bar{w}_{\text{AIS}} \;=\; \frac{w_{\text{AIS}}}{\sum_{i=1}^N w_{\text{AIS},i}}. \tag{22}$$

Furthermore, a prioritized replay buffer lets FAB reuse past AIS samples, reducing computational costs. FAB requires significantly fewer energy evaluations than previous methods, though it has limited scalability to more complex systems.

E.3 Iterated Denoising Energy Matching (iDEM)

Iterated Denoising Energy Matching (iDEM) (Akhound-Sadegh et al., 2024) trains a diffusion-based sampler by replacing standard score-matching with a Denoising Energy Matching objective. This is done by constructing a Monte-Carlo estimator of the score for noised points $x_t \sim \mathcal{N}(x_1, \sigma_t^2\mathbf{I})$:

$$\hat{s}_K(x_t,t) = \frac{\frac{1}{K}\sum_{i=1}^K \exp\left(-\mathcal{E}\left(x_{1|t}^{(i)}\right)\right)\nabla \exp\left(-\mathcal{E}\left(x_{1|t}^{(i)}\right)\right)}{\frac{1}{K}\sum_{j=1}^K \exp\left(-\mathcal{E}\left(x_{1|t}^{(j)}\right)\right)}, \quad x_{1|t}^{(1)},\ldots,x_{1|t}^{(K)} \sim \mathcal{N}\left(x_t, \sigma_t^2\right), \tag{23}$$

and fits a score network via the loss $\mathcal{L}_{\text{DEM}}(\theta) = \mathbb{E}_{t,x_t}[\|s_\theta(x_t,t) - \hat{s}_K(x_t,t)\|^2]$. Training proceeds in two coupled loops with a replay buffer strategy. Similar to our approach, iDEM uses an iterative refinement strategy where the model is progressively improved by using samples from the current model to train the next iteration. The method requires gradients of the energy function and, while it scales well to high-dimensional systems, requires relatively many energy evaluations. Notably, iDEM was the first method to successfully train using only energy evaluations on the challenging 55-particle Lennard-Jones system.

E.4 Flow Matching-Based Methods Requiring Target Data

Klein et al. (2023) first sampled molecular equilibrium distributions in Cartesian coordinates using equivariant flow matching with graph neural networks. This was extended to transferable Boltzmann generators that generalize across chemical space without retraining (Klein & Noé, 2024). Furthermore, Vaitl & Klein (2025) demonstrated that fine-tuning such models with path gradients can significantly improve sampling efficiency. Relatedly, Yu et al. (2024) proposed Force-guided Bridge Matching, a conditional bridge-matching framework that employs a hybrid approach combining data and energy evaluations in a two-stage training procedure.

E.5 Further Energy-Only Methods

Beyond FAB and iDEM, several other approaches train generative models using only energy evaluations. NETS (Albergo & Vanden-Eijnden, 2024) uses neural transport samplers to construct efficient sampling schemes. He et al. (2024) target the reverse diffusive KL divergence as a training objective to mitigate mode-seeking behavior. Sequential Controlled Langevin Diffusions (SCLD) (Chen et al., 2025) connects diffusion models with Sequential Monte Carlo via controlled Langevin dynamics.

Underdamped Diffusion Bridges (Blessing et al., 2025b) construct bridge processes based on underdamped Langevin SDEs. Iterated Energy-based Flow Matching (iEFM) (Woo & Ahn, 2024) is the only other flow matching approach using only energy evaluations. iEFM adapts the iDEM framework by deriving Monte Carlo estimators for target vector fields; however, it requires significantly more energy evaluations during training and has only been demonstrated on smaller systems (GMM-40, DW-4).

F Future Work and Further Extensions

Beyond the results presented in the main text, several directions for future work remain. Evaluation on larger molecular systems (di-, tetra-, hexapeptides) and systematic comparison with recent methods would help assess the relative performance of our approach. Additionally, investigating single-model fine-tuning versus separate model retraining (as done in TA-BG) would provide insights into optimal temperature annealing strategies.

In the following, we detail three extensions to our framework that we explored: mixture model proposals for computational efficiency, alternative gradient estimation strategies for improved stability, and hybrid approaches that incorporate small amounts of target data when available.

F.1 Mixture Model Proposals for Improved Efficiency

While using the current model q_θ as the proposal distribution in iEWFM is well-motivated, it is computationally expensive, and the likelihood evaluations can be inaccurate, potentially leading to unreliable importance weights.

To address these limitations, we explored approximating the proposal distribution more efficiently using buffer samples from the current model q_θ. One approach is to use kernel density estimation (KDE) to approximate the current model distribution based on the buffer samples. Given buffer samples $\{x_i\}_{i=1}^N$ from the current model q_θ, we construct the smoothed proposal distribution as

$$\mu_{\text{prop}}(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i), \tag{24}$$

where K_h is a kernel function with bandwidth h. For instance, using a Gaussian kernel yields a Gaussian mixture model as the proposal distribution. This "smoothed" proposal distribution μ_{prop} provides efficient sampling and density evaluations.

However, our experiments revealed that the quality we achieve with this approach depends strongly on the kernel bandwidth h, which controls the smoothness of the KDE approximation. We found an interesting trade-off: smaller bandwidths improved the Wasserstein distance but worsened the negative log-likelihood performance, suggesting potential bias in the learned distribution. Due to these concerns, we ultimately did not adopt this approach for our main experiments, though we believe it represents an interesting direction for future work.

F.2 Alternative Gradient Estimation Strategy

An alternative strategy for estimating the EWFM objective, which we did not implement in this work, involves rewriting it as a set of nested expectations. This approach may improve the stability of the estimation, particularly for target distributions with sharp modes. The objective can be expressed as

$$\mathcal{L}_{\text{EWFM}} = \mathbb{E}_{t\sim U[0,1], X_t\sim p_t}\left[\mathbb{E}_{X_1\sim p_{1|t}(\cdot|X_t)}\left[\frac{w(X_1)}{\mathbb{E}_{X_1'\sim\mu_{\text{prop}}}[w(X_1')]}\left\|u_t^\theta(X_t) - u_t(X_t|X_1)\right\|^2\right]\right], \tag{25}$$

where the outer expectation is over the marginal path p_t and the inner is over the posterior path $p_{1|t}$. One could then generate the required samples as follows: first draw X_t by interpolating between a prior sample and a proposal sample, then draw X_1 for the inner expectation by sampling from the posterior path conditioned on X_t. However, this formulation is incompatible with our efficient sample buffer strategy, as the inner expectation requires repeated sampling of new X_1 samples conditioned on X_t, which prevents pre-computation of the proposal log-densities and energy evaluations.

F.3 COMBINING WITH TARGET SAMPLES

While our methods are designed for settings without target samples, a potential extension could be to leverage a small dataset if one exists. This might help stabilize the initial stages of training, particularly for the iEWFM algorithm. Following the original Boltzmann generator framework (Noé et al., 2019), one could explore a hybrid loss function

$$\mathcal{L}_{\text{hybrid}}(\theta; \lambda) = (1 - \lambda)\mathcal{L}_{\text{EWFM}}(\theta; \mu_{\text{prop}}) + \lambda \mathcal{L}_{\text{CFM}}(\theta), \tag{26}$$

where $\lambda \in [0, 1]$ is a weighting parameter. The $\mathcal{L}_{\text{EWFM}}$ component would still benefit from the iterative proposal scheme. To mitigate overfitting to the small dataset, it may be beneficial to anneal the hyperparameter λ during training, starting with a higher value and gradually decreasing it to zero.

It is important to note that this hybrid approach cannot be directly combined with our aEWFM algorithm. The available target samples are from the distribution at the final target temperature T, not the higher intermediate temperatures $T_i > T$ used in the annealing schedule. Integrating target samples into aEWFM would require a more complex procedure, such as using importance resampling to adapt the target samples to different temperatures.

G LLM USAGE DISCLOSURE

During the preparation of this manuscript, large language models (LLMs) were used in a supporting role. For writing, we employed LLMs to provide grammar and style suggestions; no proofs or derivations were generated by LLMs. In addition, we used advanced reasoning models to challenge our own interpretations and conclusions.

For retrieval and discovery, we employed LLM-based Deep Research tools to help identify potentially relevant references and to occasionally summarize external work for scoping purposes.

Overall, LLMs served as an auxiliary tool for writing polish, exploratory literature discovery, and internal cross-checking, without contributing novel technical content.