Learning Boltzmann Generators via Constrained Mass Transport

Anonymous Author(s)
Affiliation
Address
email

Abstract

Efficient sampling from high-dimensional and multimodal unnormalized probability distributions is a central challenge in many areas of science and machine learning. We focus on Boltzmann generators (BGs) that aim to sample the Boltzmann distribution of physical systems, such as molecules, at a given temperature. Classical variational approaches that minimize the reverse Kullback–Leibler divergence are prone to mode collapse, while annealing-based methods, commonly using geometric schedules, can suffer from mass teleportation and rely heavily on schedule tuning. We introduce *Constrained Mass Transport* (CMT), a variational framework that generates intermediate distributions under constraints on both the KL divergence and the entropy decay between successive steps. These constraints enhance distributional overlap, mitigate mass teleportation, and counteract premature convergence. Across standard BG benchmarks and the here introduced *ELIL tetrapeptide*, the largest system studied without access to samples from molecular dynamics, CMT consistently surpasses state-of-the-art variational methods, achieving more than 2.5× higher effective sample size while avoiding mode collapse.

1 Introduction

We consider the problem of sampling from a target probability measure $p \in \mathcal{P}(\mathbb{R}^d)$ given by $p(x) = \tilde{p}(x)/\mathcal{Z}$ where $\tilde{p} \in C(\mathbb{R}^d, \mathbb{R}_{\geq 0})$ can be evaluated pointwise but the normalization constant $\mathcal{Z} = \int_{\mathbb{R}^d} \tilde{p}(x)\,\mathrm{d}x$ is intractable. Sampling from unnormalized densities arises in many areas, including Bayesian statistics [34], reinforcement learning [20], and the natural sciences [70]. A prominent example is learning Boltzmann generators (BGs) [55], for which $\tilde{p}(x) = \exp(-E(x)/k_B T)$, with E being an energy function, T the temperature, and k_B the Boltzmann constant. BGs enable efficient sampling of thermodynamic ensembles, thereby bypassing costly molecular dynamics (MD) simulations and accelerating the exploration of rare but physically important states. However, learning BGs is challenging as the state space is typically high-dimensional, the target distribution is often highly multimodal, and evaluating $E(x)$ can be very costly, especially when using accurate energies such as those from density-functional theory [12].

A promising alternative to classical MD or Monte Carlo methods [38] is offered by variational approaches [71], which aim to minimize a statistical divergence between a variational probability measure $q \in \mathcal{P}(\mathbb{R}^d)$ and the target p, commonly the reverse Kullback–Leibler (KL) divergence

$$q^* = \operatorname*{arg\,min}_{q \in \mathcal{P}(\mathbb{R}^d)} D_{\mathrm{KL}}(q \,\|\, p), \tag{1}$$

whose unique minimizer is $q^* = p$. Directly minimizing the reverse KL divergence tends to suffer from mode collapse, ignoring low-probability modes of the target [15]. To counteract this, a number of recent approaches have proposed to construct a sequence of intermediate distributions that transport probability mass from a tractable base distribution q_0 to the target. This idea, which dates back several decades to annealed importance sampling [53], is most often realized through a geometric annealing path, which is defined as a sequence of $(q_i)_{i=1}^I$ which follows $q_i \propto q_0^{1-\beta_i}\, \tilde{p}^{\beta_i}$ where the

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

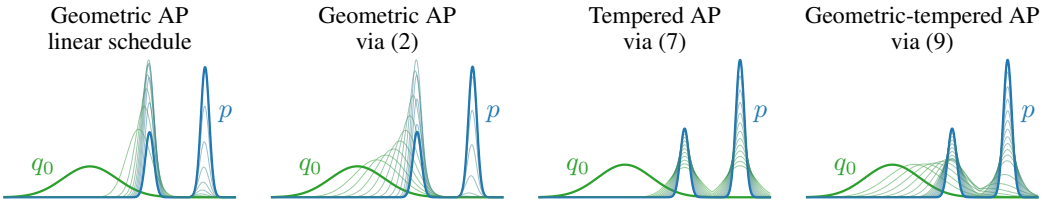


Figure 1: Illustration of the annealing paths (AP) obtained by solving the variational problems (2), (7), or (9). Trust-region–based optimization (2) mitigates the irregularities of naive schedules (e.g., the linear schedule), but the resulting geometric AP suffers from mass teleportation as the right mode of the target distribution p emerges without overlap with earlier intermediate densities. Constraining the entropy decay between successive densities (7) prevents mass teleportation, yet fails to guarantee sufficient overlap between the initial distribution q_0 and subsequent intermediate densities. In contrast, combining both constraints (9) yields APs that both maintain overlap between successive densities and avoid mass teleportation.

corresponding annealing schedule $(\beta_i)_{i=1}^I$ ensures that $q_I = p$. Despite its simplicity, geometric annealing can suffer from mass teleportation, where large portions of the probability mass shift to disjoint regions between successive steps, complicating mass transport [6]. Moreover, its performance critically depends on the choice of annealing schedule [73].

A recent study proposed selecting the annealing schedule using trust-region constraints that bound the KL divergence between successive distributions, thereby avoiding abrupt changes in the stochastic optimal control setting [14]. Adapting this idea to measures that admit densities, we further introduce a constraint that explicitly controls the rate at which the entropy of the variational distribution decreases along the transport path. This added degree of freedom enables deviations from the standard geometric annealing schedule, mitigating issues such as mass teleportation and premature convergence, while fostering greater overlap between consecutive distributions.

We demonstrate that our method, *Constrained Mass Transport* (CMT), consistently outperforms state-of-the-art approaches, often by a substantial margin, when learning Boltzmann generators solely from energy evaluations, without relying on additional MD samples. Furthermore, we introduce a new benchmark system, *ELIL tetrapeptide*, which, to the best of our knowledge, is the largest system studied to date in the setting of learning exclusively from energy evaluations. On this challenging system, CMT achieves a $2.5\times$ larger effective sample size compared to baselines, while avoiding mode collapse.

2 Constrained mass transport

Here, we denote by $\mathcal{P}(\mathbb{R}^d)$ the space of probability measures on $\mathbb{R}^d$ that are absolutely continuous with respect to Lebesgue measure and admit smooth densities. We approach the sampling problem by dividing (1) into a sequence of constrained optimization problems that result in an annealing path of intermediate densities $(q_i)_{i=0}^I$ that bridge between a tractable prior q_0 and the target p.

Trust region constraint. Trust-regions aim at dividing the problem (1) into subproblems by constraining the updated density to be close to the old density in terms of KL divergence. Formally, this is given by the iterative optimization scheme [1]

$$q_{i+1} = \operatorname*{arg\,min}_{q \in \mathcal{P}(\mathbb{R}^d)} D_{\mathrm{KL}}(q|p) \quad \text{s.t.} \quad D_{\mathrm{KL}}(q|q_i) \leq \varepsilon_{\mathrm{tr}}, \tag{2}$$

for $i \in \mathbb{N}$, trust region bound $\varepsilon_{\mathrm{tr}} > 0$ and some $q_0 \in \mathcal{P}(\mathbb{R}^d)$. Due to the convexity of the KL divergence, we can show that in all but the last step we actually have an equality constraint in (2); see App. A. Thus, there exists an $I \in \mathbb{N}$ such that $q_I = q^* \ (= p)$. Under suitable regularity assumptions, we can approach the above constrained optimization problem using a relaxed Lagrangian formalism, i.e.,

$$\mathcal{L}_{\mathrm{tr}}^{(i+1)}(q, \lambda) = D_{\mathrm{KL}}(q|p) + \lambda \left(D_{\mathrm{KL}}(q|q_i) - \varepsilon_{\mathrm{tr}} \right) \tag{3}$$

where $\lambda \geq 0$ is a Lagrange multiplier, and solve the saddle point problems

$$\max_{\lambda \geq 0} \min_{q \in \mathcal{P}(\mathbb{R}^d)} \mathcal{L}_{\mathrm{tr}}^{(i)}(q, \lambda). \tag{4}$$

[1] To ensure that $q \in \mathcal{P}(\mathbb{R}^d)$ we need an additional constraint $\int q(x)\mathrm{d}x = 1$ which we omitted in the main part of the paper for readability.

We note that $\mathcal{L}_{\text{tr}}^{(i)}$ is convex in q by convexity of the KL divergence and (4) concave in λ since it can be expressed as the pointwise minimum $\min_q \mathcal{L}_{\text{tr}}^{(i)}(q, \lambda)$ among a family of linear functions of λ. Thus, (4) has unique optima which we denote by q_{i+1} and λ_i, respectively. Indeed, (2) admits an analytical solution which is characterized by Prop. 2.1. We refer to App. A for a proof and further details on problem (2).

Proposition 2.1 (Optimal intermediate trust-region densities)**.** *The intermediate optimal densities that solve* (2) *satisfy*

$$q_{i+1}(x, \lambda) = \frac{q_i(x)^{\frac{\lambda}{1+\lambda}} \tilde{p}(x)^{\frac{1}{1+\lambda}}}{\mathcal{Z}_{i+1}(\lambda)}, \quad \textit{with} \quad \mathcal{Z}_{i+1}(\lambda) = \int q_i(x)^{\frac{\lambda}{1+\lambda}} \tilde{p}(x)^{\frac{1}{1+\lambda}} \mathrm{d}x, \tag{5}$$

where q_{i+1} are the unique optima of the Lagrangian corresponding to (2)*.*

The optimal multiplier λ_i that solves (2) is obtained by plugging $q_{i+1}(\lambda)$ in the Lagrangian (3) to obtain the dual function $g_{\text{tr}} \in C(\mathbb{R}, \mathbb{R})$ given by

$$g_{\text{tr}}^{(i+1)}(\lambda) \coloneqq \mathcal{L}_{\text{tr}}^{(i+1)}(q_{i+1}(\lambda), \lambda) = -(1+\lambda) \log \mathcal{Z}_{i+1}(\lambda) - \lambda \varepsilon_{\text{tr}}. \tag{6}$$

Assuming access to $\mathcal{Z}_{i+1}(\lambda)$ one can solve $\lambda_i = \arg\max_{\lambda \geq 0} \; g_{\text{tr}}^{(i+1)}(\lambda)$ to obtain the optimal $q \in \mathcal{P}(\mathbb{R}^d)$ that solves (2) as $q_{i+1} \coloneqq q_{i+1}(\lambda_i)$.

Entropy constraint. In a similar fashion to (2), we can avoid premature convergence by regulating the entropy decay of the model by constructing a sequence of intermediate densities whose change in entropy is constrained. Formally, we aim to solve the following problem

$$q_{i+1} = \underset{q \in \mathcal{P}(\mathbb{R}^d)}{\arg\min} \; D_{\text{KL}}(q|p) \quad \text{s.t.} \quad H(q_i) - H(q) \leq \varepsilon_{\text{ent}}, \tag{7}$$

where $H(q) = -\int q(x) \log q(x) \mathrm{d}x$ is the Shannon entropy and $\varepsilon_{\text{ent}} > 0$ the entropy bound. We can again approach (7) using a Lagrangian formalism by introducing a Lagrangian multiplier $\eta \geq 0$. The analytical solution to (7) is characterized by Prop. 2.2 whose proof can be found in App. A.

Proposition 2.2 (Optimal intermediate densities for entropy constraint)**.** *The intermediate optimal densities, that solve* (7) *satisfy*

$$q_{i+1}(x, \eta) = \frac{\tilde{p}(x)^{\frac{1}{1+\eta}}}{\mathcal{Z}_{i+1}(\eta)}, \quad \textit{with} \quad \mathcal{Z}_{i+1}(\eta) = \int \tilde{p}(x)^{\frac{1}{1+\eta}} \mathrm{d}x, \tag{8}$$

where q_{i+1} are the unique optima of the Lagrangian corresponding to (7)*.*

Despite the potential of (7) for counteracting premature convergence, we identify two challenges depending on the entropy of the initial density $H(q_0)$: First, if $H(q_0) < H(p)$ then the constraint is inactive resulting in $\eta_0 = 0$, reducing (7) to the optimization problem as stated in (1). Second, if $H(q_0) \gg H(p)$ then the KL divergence between q_0 and $q_1 \propto p^{1/1+\eta_0}$ can be arbitrarily large and therefore could cause instabilities due to a lack of overlap between the successive densities. While the former challenge can typically be addressed by initializing q_0 with large entropy, the second can be more intricate. In the following, we explain how this challenge can be addressed by combining the trust-region and entropy constraint.

Combining both constraints. One can straightforwardly combine the constraints in (2) and (7) into a single iterative optimization scheme defined as

$$q_{i+1} = \underset{q \in \mathcal{P}(\mathbb{R}^d)}{\arg\min} \; D_{\text{KL}}(q|p) \quad \text{s.t.} \quad \begin{cases} D_{\text{KL}}(q|q_i) \leq \varepsilon_{\text{tr}}, \\ H(q_i) - H(q) \leq \varepsilon_{\text{ent}}. \end{cases} \tag{9}$$

In analogy to the previous section, we introduce Lagrangian multiplier λ and η for the trust region and entropy constraint, respectively. Indeed, one can again obtain an analytical expression for the evolution of the optimal densities, see Prop. 2.3 and App. A for a proof.

Proposition 2.3 (Optimal intermediate densities for entropy and trust region constraint)**.** *The intermediate optimal densities that solve* (7) *satisfy*

$$q_{i+1}(x, \lambda, \eta) = \frac{q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} \tilde{p}(x)^{\frac{1}{1+\lambda+\eta}}}{\mathcal{Z}_{i+1}(\lambda, \eta)} \quad \textit{with} \quad \mathcal{Z}_{i+1}(\lambda, \eta) = \int q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} \tilde{p}(x)^{\frac{1}{1+\lambda+\eta}}(x) \mathrm{d}x, \tag{10}$$

where q_{i+1} are the unique optima of the Lagrangian corresponding to (9)*.*

Clearly, if $H(q_0) \gg H(p)$, the trust-region constraint ensures that the KL divergence between q_0 and q_1 is at most ε_{tr} and, therefore, for a suitable choice of ε_{tr} ensures that two consecutive densities have sufficient overlap. Lastly, the Lagrangian dual function $g_{\text{tr-ent}} \in C(\mathbb{R}^2, \mathbb{R})$ corresponding to (9),

that is,

$$g_{\text{tr-ent}}^{(i+1)}(\lambda,\eta) \coloneqq -(1+\lambda+\eta)\log \mathcal{Z}_{i+1}(\lambda,\eta) - \lambda\varepsilon_{\text{tr}} - \eta(H(q_i) - \varepsilon_{\text{ent}}), \tag{11}$$

requires solving a two-dimensional convex optimization problem to obtain λ_i, η_i which can be done efficiently in practice; see Sec. 3 for additional details.

Connection to annealing paths. Iteratively solving (2), (7) or (9) induces an *annealing path*, that is, a sequence of densities $(q_i)_{i\in\mathbb{N}}$ that interpolates between q_0 and p. We characterize these paths in Thm. 2.4; See App. A for a proof.

Theorem 2.4 (Annealing paths)**.** *Let $p \in \mathcal{P}(\mathbb{R}^d)$ be the target density and $q_0 \in \mathcal{P}(\mathbb{R}^d)$ some initial density. The intermediate optimal densities, that solve* (2)*,* (7) *and* (9) *satisfy*

$$q_i \propto q_0^{1-\beta_i}\tilde{p}^{\beta_i}, \quad q_i \propto \tilde{p}^{\alpha_i}\ (i \geq 1), \quad \textit{and} \quad q_i \propto q_0^{1-\beta_i}\,(\tilde{p}^{\alpha_i})^{\beta_i}\,, \tag{12}$$

respectively, with β and α being functions of the corresponding Lagrangian multiplier. Moreover, the sequences $(\alpha_i)_{i\in\mathbb{N}_0}$ and $(\beta_i)_{i\in\mathbb{N}_0}$ take values in $[0,1]$, satisfy $\alpha_0 = \beta_0 = 0$ and $\alpha_I = \beta_I = 1$ for some $I \in \mathbb{N}_+$ and $(\beta_i)_{i\in\mathbb{N}_0}$ is monotonically increasing.

In what follows, we refer to the annealing paths in (12) as geometric (G), tempered (T), and geometric-tempered (GT) annealing paths, respectively; see Fig. 1 for an illustration.

3 Learning the intermediate densities

A general recipe. So far, we discussed how one can construct a sequence of intermediate measures $(q_i)_{i\in\mathbb{N}}$ using our constrained mass transport formulation. However, despite having access to the analytical form of q_i, it is typically not possible to sample from it directly. As such, we approximate each q_i by a distribution from a tractable class $\mathcal{Q} \subset \mathcal{P}(\mathbb{R}^d)$ that permits efficient sampling and density evaluation. Given an approximation family $\mathcal{Q}$, we select $\hat{q}_i \in \mathcal{Q}$ to approximate q_i by solving

$$\hat{q}_i = \operatorname*{arg\,min}_{q\in\mathcal{Q}}\ D(q_i, q), \tag{13}$$

where D is an arbitrary statistical divergence between probability measures. This formulation is general: the choice of $\mathcal{Q}$ and D determines the trade-off between expressivity, computational cost, and statistical properties such as mode coverage or robustness.

Practical algorithm. In this work, we choose $\mathcal{Q}$ to be a normalizing flow family constructed via push-forwards of a simple base measure. Let $q_z \in \mathcal{P}(\mathbb{R}^d)$ be an easy-to-sample base measure (e.g., a standard Gaussian), and let $\mathcal{F}$ be a class of smooth invertible maps $f : \mathbb{R}^d \to \mathbb{R}^d$. We define

$$\mathcal{Q}_{\text{NF}} \coloneqq \{\, f_{\#}q_z \mid f \in \mathcal{F}\,\}, \quad \text{with} \quad (f_{\#}q_z)(z) = q_z\big(f^{-1}(z)\big)\left|\det \frac{\partial f^{-1}(z)}{\partial z}\right|. \tag{14}$$

with push-forward $f_{\#}q_z$. To fit $\hat{q}_i$ within this family, we take D to be the importance-weighted forward KL divergence

$$\hat{q}_{i+1} = \operatorname*{arg\,min}_{q\in\mathcal{Q}_{\text{NF}}}\ D_{\text{KL}}(q_{i+1}|q) \quad \text{with} \quad D_{\text{KL}}(q_{i+1}|q) \;=\; \mathbb{E}_{x\sim q_i}\left[\frac{q_{i+1}(x)}{q_i(x)}\log\left(\frac{q_{i+1}(x)}{q(x)}\right)\right] \tag{15}$$

This choice offers several advantages. First, forward KL strongly penalizes underestimating the support of q_{i+1}, encouraging mode coverage and reducing the risk of mode collapse. Second, because q_{i+1} is available in closed form from the constrained transport updates (see Prop. 2.1, 2.2 and 2.3), the importance weights $q_{i+1}(x)/q_i(x)$ can be computed solely from q_i and p. Third, the importance-weighted formulation allows us to reuse samples drawn from q_i, enabling a seamless integration of replay buffers, resulting in increased sample efficiency. Lastly, the trust-region constraint controls the variance of the importance weights, keeping it approximately constant, independent of the problem dimension d (see App. A), resulting in a highly scalable algorithm.

Lagrangian dual optimization. Maximizing the concave dual function (11) requires evaluating intermediate normalization constants $\mathcal{Z}_{i+1}$. This can be done efficiently by expressing $\mathcal{Z}_{i+1}$ as an expectation under q_i and using Monte Carlo estimation. For instance, the expression for $\mathcal{Z}_{i+1}$ in (10) can be estimated as

$$\mathcal{Z}_{i+1}(\lambda,\eta) = \mathbb{E}_{x\sim q_i}\left[\left(\frac{\tilde{p}(x)}{q_i(x)^{1+\eta}}\right)^{\frac{1}{1+\lambda+\eta}}\right] \approx \frac{1}{N}\sum_{x_n\sim q_i}\left(\frac{\tilde{p}(x_i)}{q_i(x_i)^{1+\eta}}\right)^{\frac{1}{1+\lambda+\eta}}. \tag{16}$$

We note that samples $x_n \sim q_i$ and the corresponding evaluations $q_i(x_n)$ and $\tilde{p}(x_n)$ are typically already computed when solving (13), so the additional cost of determining the Lagrange multipliers is minimal. Details of the dual optimization procedure are provided in App. C.3, including a code

Algorithm 1 Constrained mass transport

Require: Initial measure q_0, target measure $\tilde{p}$, divergence D, approximation family $\mathcal{Q}$, buffer size N
for $i \leftarrow 0, \ldots, I-1$ **do**
  Draw N samples $x_n \sim q_i$, evaluate $q_i(x_n), \tilde{p}(x_n)$ and initialize buffer $\mathcal{B}^{(i)} = (x_n, q_i(x_n), \tilde{p}(x_n))_{n=1}^N$
  Compute multiplier $\lambda_i, \eta_i = \arg\max_{\lambda,\eta \in \mathbb{R}^+} \; g_{\text{tr-ent}}^{(i+1)}(\lambda, \eta)$ using $\mathcal{B}^{(i)}$
  Compute $q_{i+1} \approx \hat{q}_{i+1} = \arg\min_{q \in \mathcal{Q}} \; D(q_{i+1}, q)$ using $\mathcal{B}^{(i)}$
return $\hat{q}_I \approx p$

example. Lastly, we refer to Algorithm 1 for an algorithmic overview of the constrained measure transport method.

4 Related work

Boltzmann generators. Learning molecular Boltzmann generators [55] purely from energy evaluations has been explored using internal coordinate representations of the system, with both flow-based methods [28, 45, 46, 51, 64] and diffusion-based methods [22, 41, 48]. While flow-based approaches have demonstrated strong performance, their diffusion-based counterparts remain less competitive, often struggling with mode collapse, even on relatively small systems. Alternative approaches operate in Cartesian coordinates [6, 43, 50, 74] allowing for transferability between different systems [44, 75].

Trust regions and entropy constraints. Trust region methods have a long history as robust optimization algorithms that iteratively minimize an objective within an adaptively sized "trust region"; see [25] for an overview. Beyond classical optimization, these methods have been extended to operate over spaces of probability distributions, with applications in reinforcement learning [2–4, 9, 49, 58, 59, 61, 65, 66, 80–82], black-box optimization [1, 72, 78], variational inference [10, 11], and path integral control [35, 76]. The first explicit link between trust-region optimization and geometric annealing paths was established in [14] for path space measures in the setting of stochastic optimal control. Entropy constraints, often introduced as entropy regularization, have also been studied in policy optimization and reinforcement learning, either in the form of soft constraints [5, 52, 57] or hard constraints [1, 7–9, 59]. However, prior work typically constrains the absolute entropy value, which is problematic for inference tasks, since it requires prior knowledge of the target density's entropy. To the best of our knowledge, such methods have not yet been extended to sampling problems. Furthermore, the connection between entropy-constrained optimization and annealing paths has not previously been established.

Improved annealing paths. Research on improving annealing paths (APs) has largely focused on geometric APs in the context of annealed importance sampling (AIS) [53] and their extensions to sequential Monte Carlo (SMC) [27]; see [23, 36, 40, 73]. Beyond the standard geometric AP, alternative constructions have been proposed, such as the moment-averaging path for exponential family distributions [37] and the arithmetic mean path [21]. The geometric path itself can be interpreted as a quasi-arithmetic mean [47] under the natural logarithm, which motivated [17] to propose APs based on the deformed logarithm transformation. A variational characterization of these paths was later analysed in [18]. Related work also explores improved schedules for the diffusion coefficient in ergodic Ornstein–Uhlenbeck processes used to train denoising diffusion models [39, 67]; see, e.g., [13, 54, 79, 83].

5 Numerical evaluation

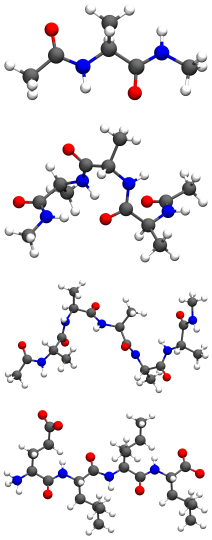
In this section, we compare our approach against state-of-the-art methods on four challenging molecular systems. We provide a brief overview of the experimental setup here, with full details in App. C. Additional experimental results are provided in App. B, including extended performance metrics, an ablation study on the effect of both constraints, and an analysis of the trust-region constraint across systems of different dimensionality.

5.1 Experimental setup

Benchmark problems. Our evaluation covers a range of molecular systems, beginning with the well-studied alanine dipeptide ($d = 60$) [29, 51, 69, 74], and extending to the larger alanine tetrapeptide ($d = 120$) and alanine hexapeptide ($d = 180$), which have only recently been addressed using variational methods [64]. In addition, we introduce a new benchmark, the ELIL tetrapeptide ($d = 219$), which is higher-dimensional and which contains more complex side chain interactions compared to the alanine hexapeptide. To the best of our knowledge, this represents the largest and most complex

Table 1: Results for all systems of varying dimensionality d. Evaluation criteria include the number of target evaluations (*target evals*), the negative log-likelihood (*NLL*), and the effective sample size (*ESS*). Each value is shown as the mean $\pm$ standard error over four independent runs. The best results are highlighted in bold, except for the reverse KL, which is prone to mode collapse, making ESS values not directly comparable.

System	Method	target evals ↓	NLL ↓	ESS [%] ↑
Alanine Dipeptide ($d = 60$)	Forward KL	5×10^9	-213.581 ± 0.000	(82.16 ± 0.09) %
	Reverse KL	2.56×10^8	-213.609 ± 0.006	(94.11 ± 0.21) %
	FAB	2.13×10^8	-213.653 ± 0.000	(94.81 ± 0.04) %
	TA-BG	1×10^8	-213.666 ± 0.001	(95.77 ± 0.12) %
	CMT (ours)	1×10^8	$\mathbf{-213.677 \pm 0.000}$	$\mathbf{(97.69 \pm 0.03)}$ %
Alanine Tetra-Peptide ($d = 120$)	Forward KL	4.2×10^9	-330.069 ± 0.001	(45.29 ± 0.08) %
	Reverse KL	2.56×10^8	-329.191 ± 0.122	(74.88 ± 3.65) %
	FAB	2.13×10^8	-330.100 ± 0.002	(63.59 ± 0.23) %
	TA-BG	1×10^8	-330.143 ± 0.002	(64.87 ± 0.21) %
	CMT (ours)	1×10^8	$\mathbf{-330.167 \pm 0.002}$	$\mathbf{(68.49 \pm 0.14)}$ %
Alanine Hexa-Peptide ($d = 180$)	Forward KL	4.2×10^9	-501.598 ± 0.005	(10.97 ± 0.11) %
	Reverse KL	2.56×10^8	-497.378 ± 0.277	(22.22 ± 1.44) %
	FAB	4.2×10^8	-501.268 ± 0.008	(14.64 ± 0.08) %
	TA-BG	4×10^8	-501.582 ± 0.010	(15.89 ± 0.13) %
	CMT (ours)	4×10^8	$\mathbf{-501.761 \pm 0.008}$	$\mathbf{(29.20 \pm 0.24)}$ %
ELIL Tetra-Peptide ($d = 219$)	Forward KL	4.2×10^9	-597.571 ± 0.004	(5.80 ± 0.04) %
	Reverse KL	2.56×10^8	-583.381 ± 3.033	(1.25 ± 0.45) %
	FAB	8.43×10^8	-597.370 ± 0.006	(7.30 ± 0.08) %
	TA-BG	8×10^8	-597.830 ± 0.022	(10.12 ± 0.38) %
	CMT (ours)	8×10^8	$\mathbf{-598.440 \pm 0.003}$	$\mathbf{(25.91 \pm 0.19)}$ %



molecular system investigated using variational approaches to date. A detailed description of all benchmark systems is provided in App. C.2.

Baseline methods. Our main baselines are Flow Annealed Importance Sampling Bootstrap (FAB) [51] and Temperature-Annealed Boltzmann Generators (TA-BG) [64], which currently define the state of the art for learning molecular systems on internal coordinate representations. For reference, we also include reverse and forward KL training; the latter leverages ground truth samples obtained from molecular dynamics (MD) simulations (see App. C.2). To ensure a fair comparison, all methods use neural spline flows [31] and identical architectures.

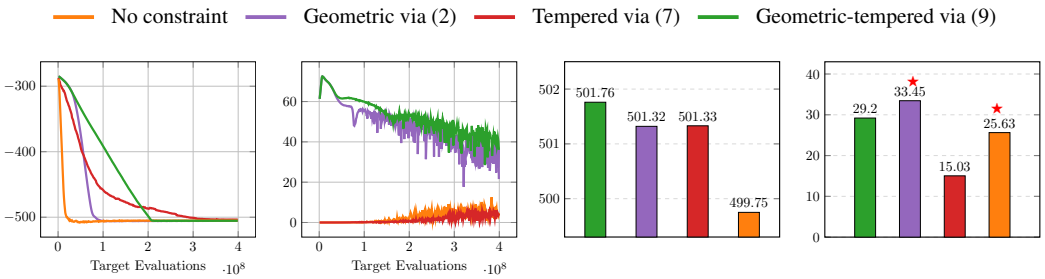
Performance criteria. We evaluate methods primarily using two criteria. First, the negative log-likelihood (NLL), computed with ground truth MD samples. Up to an additive constant, the NLL corresponds to the forward KL divergence and is therefore well suited for detecting mode collapse [15]. Second, we consider the effective sample size (ESS), defined as $\mathrm{ESS}(q, p) := \left(\mathbb{E}_{x \sim q}\left[\left({}^{p(x)}/_{q(x)}\right)^2\right]\right)^{-1}$. ESS is a common measure of sample quality, but it is known to be less reliable for assessing mode collapse [15]. For details on all metrics, we refer to App. C.4. Since evaluating the target density of molecular systems is typically expensive, we also report the number of target evaluations required by each method.

Finally, we also consider Ramachandran plots as a qualitative criterion for assessing mode collapse. These plots visualize low-dimensional projections of important molecular configurations, making it possible to assess whether the generated samples capture all relevant modes of the distribution or fail to represent certain regions of the state space. For more details on Ramachandran plots, we refer to [64].

5.2 Results

Main results. The main findings are summarized in Tab. 1. While the performance gap between our method and the baselines is less pronounced for smaller systems, it widens substantially for the larger ones. In particular, on alanine hexapeptide and ELIL tetrapeptide, our method attains approximately twice the ESS of competing approaches, while also avoiding mode collapse, as reflected in improved NLL values.

Ablation study for constraints. Additionally, we investigate the effect of different constraint choices on the performance of the alanine hexapeptide system. Specifically, we compare four settings: using both constraints, each constraint individually, and no constraint (which corresponds to importance-weighted forward KL minimization). The results are summarized in Fig. 2 and Fig. 3. Fig. 2a



(a) $H(\hat{q}_i)$ (b) $\mathrm{ESS}(\hat{q}_{i-1}, q_i)$ [%] (c) $-\mathrm{NLL}$ (d) $\mathrm{ESS}(\hat{q}_I, p)$ [%]

Figure 2: Impact of the trust-region and entropy constraint visualized on alanine hexapeptide. Fig. 2a visualizes the model entropy over the course of the training. Analogously, Fig. 2b shows the importance-weight effective sample size (ESS) between successive intermediate densities. Figs. 2c and 2d depict the final log-likelihood and ESS to the target density, respectively. The variants in Fig. 2d marked with "★" exhibit visible mode-collapse on the Ramachandran plots; see Fig. 3. The ESS is therefore not directly comparable to the other methods. We denote by $\hat{q}_i$ the variational approximation of the intermediate density q_i.

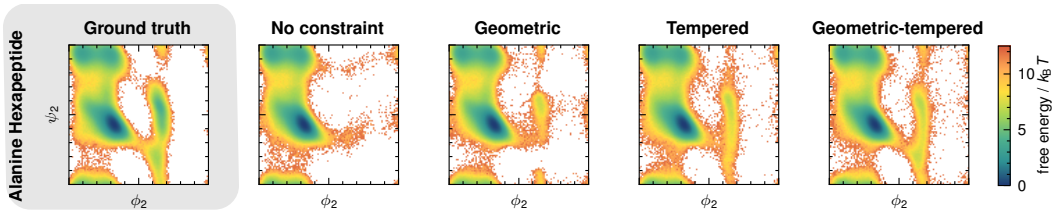


Figure 3: Ramachandran plots for alanine hexapeptide with trust-region and entropy constraints selectively enabled or disabled. Using a single or no constraint leads to mode collapse, whereas combining both constraints avoids it. See App. C.4 for details.

shows that omitting the trust region constraint causes entropy to decrease rapidly, which lead to mode collapse during training. Moreover, using only the entropy constraint yields unstable training, as evidenced by violations of the prescribed linear entropy decay. In contrast, incorporating a trust region constraint stabilizes training, as reflected in Fig. 2b, where it produces a substantially higher ESS between successive intermediate densities. Fig. 3 shows Ramachandran plots of alanine hexapeptide with the constraints selectively enabled or disabled. Visible signs of mode collapse appear in all cases except for the tempered (7) and geometric-tempered (9) variants, with the most accurate Ramachandran plot observed in the latter. Overall, our findings indicate that both constraints are necessary to achieve high ESS values while simultaneously avoiding mode collapse.

6 Conclusion

We have introduced *Constrained Mass Transport* (CMT), a variational framework for constructing intermediate distributions that transport probability mass from a tractable base measure to a complex, unnormalized target. By enforcing constraints on both the KL divergence and the entropy decay between successive steps, CMT balances exploration and convergence, thereby mitigating mass teleportation, reducing mode collapse, and promoting smooth distributional overlap. Our empirical evaluation across established Boltzmann generator benchmarks and the here proposed *ELIL tetrapeptide*, learned purely from energy evaluations without access to molecular dynamics samples, demonstrates that CMT consistently outperforms existing annealing-based and variational baselines, achieving over $2.5\times$ higher effective sample size while preserving mode diversity.

Promising directions for future work include exploring alternative approximation families $\mathcal{Q}$ and divergences D for learning intermediate densities,which may yield further performance gains. Another interesting avenue is to apply our method in Cartesian coordinate representations, which enables transferability across different molecular systems [44, 75].

References

[1] A. Abdolmaleki, R. Lioutikov, J. R. Peters, N. Lau, L. Pualo Reis, and G. Neumann. Model-based relative entropy stochastic search. *Advances in Neural Information Processing Systems*, 28, 2015.

[2] A. Abdolmaleki, J. T. Springenberg, J. Degrave, S. Bohez, Y. Tassa, D. Belov, N. Heess, and M. Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018.

[3] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[4] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[5] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[6] T. Akhound-Sadegh, J. Lee, J. Bose, V. D. Bortoli, A. Doucet, M. M. Bronstein, D. Beaini, S. Ravanbakhsh, K. Neklyudov, and A. Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

[7] R. Akrour, A. Abdolmaleki, H. Abdulsamad, J. Peters, and G. Neumann. Model-free trajectory-based policy optimization with monotonic improvement. *Journal of machine learning research*, 19(14):1–25, 2018.

[8] R. Akrour, G. Neumann, H. Abdulsamad, and A. Abdolmaleki. Model-free trajectory optimization for reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970. PMLR, 2016.

[9] R. Akrour, J. Pajarinen, J. Peters, and G. Neumann. Projections for approximate policy iteration algorithms. In *International Conference on Machine Learning*, pages 181–190. PMLR, 2019.

[10] O. Arenz, P. Dahlinger, Z. Ye, M. Volpp, and G. Neumann. A unified perspective on natural gradient variational inference with gaussian mixture models. *arXiv preprint arXiv:2209.11533*, 2022.

[11] O. Arenz, M. Zhong, and G. Neumann. Trust-region variational inference with gaussian mixture models. *Journal of Machine Learning Research*, 21(163):1–60, 2020.

[12] N. Argaman and G. Makov. Density functional theory: An introduction. *American Journal of Physics*, 68(1):69–79, Jan. 2000.

[13] R. Benita, M. Elad, and J. Keshet. Spectral analysis of diffusion models with application to schedule design. *arXiv preprint arXiv:2502.00180*, 2025.

[14] D. Blessing, J. Berner, L. Richter, C. Domingo-Enrich, Y. Du, A. Vahdat, and G. Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference, 2025.

[15] D. Blessing, X. Jia, J. Esslinger, F. Vargas, and G. Neumann. Beyond ELBOs: A large-scale evaluation of variational methods for sampling. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4205–4229. PMLR, 21–27 Jul 2024.

[16] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. Seventh printing with corrections, 2009.

[17] R. Brekelmans, V. Masrani, T. Bui, F. Wood, A. Galstyan, G. V. Steeg, and F. Nielsen. Annealed importance sampling with q-paths. *arXiv preprint arXiv:2012.07823*, 2020.

[18] R. Brekelmans and F. Nielsen. Variational representations of annealing paths: Bregman information under monotonic embedding. *Information Geometry*, 7(1):193–228, 2024.

[19] R. P. Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[20] O. Celik, Z. Li, D. Blessing, G. Li, D. Palenicek, J. Peters, G. Chalvatzaki, and G. Neumann. Dime: Diffusion-based maximum entropy reinforcement learning. *arXiv preprint arXiv:2502.02316*, 2025.

[21] J. Chen, D. Lu, Z. Xiu, K. Bai, L. Carin, and C. Tao. Variational inference with holder bounds. *arXiv preprint arXiv:2111.02947*, 2021.

[22] J. Choi, Y. Chen, M. Tao, and G.-H. Liu. Non-equilibrium annealed adjoint sampler. *arXiv preprint arXiv:2506.18165*, 2025.

[23] N. Chopin, F. R. Crucinio, and A. Korba. A connection between tempering and entropic mirror descent. *arXiv preprint arXiv:2310.11914*, 2023.

[24] N. Chopin, O. Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

[25] A. R. Conn, N. I. Gould, and P. L. Toint. *Trust region methods*. SIAM, 2000.

[26] T. M. Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[27] P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

[28] M. Dibak, L. Klein, A. Krämer, and F. Noé. Temperature steerable flows and boltzmann generators. *Physical Review Research*, 4(4):L042005, 2022.

[29] M. Dibak, L. Klein, A. Krämer, and F. Noé. Temperature steerable flows and Boltzmann generators. *Physical Review Research*, 4(4):L042005, Oct. 2022.

[30] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, Sept. 1987.

[31] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[32] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. nflows: normalizing flows in PyTorch, Nov. 2020.

[33] P. Eastman, R. Galvelis, R. P. Peláez, C. R. A. Abreu, S. E. Farr, E. Gallicchio, A. Gorenko, M. M. Henry, F. Hu, J. Huang, A. Krämer, J. Michel, J. A. Mitchell, V. S. Pande, J. P. Rodrigues, J. Rodriguez-Guerra, A. C. Simmonett, S. Singh, J. Swails, P. Turner, Y. Wang, I. Zhang, J. D. Chodera, G. De Fabritiis, and T. E. Markland. OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials. *J. Phys. Chem. B*, 128(1):109–116, Jan. 2024.

[34] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[35] V. Gómez, H. J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 482–497. Springer, 2014.

[36] S. Goshtasbpour, V. Cohen, and F. Perez-Cruz. Adaptive annealed importance sampling with constant rate progress. In *International Conference on Machine Learning*, pages 11642–11658. PMLR, 2023.

[37] R. B. Grosse, C. J. Maddison, and R. R. Salakhutdinov. Annealing between distributions by averaging moments. *Advances in Neural Information Processing Systems*, 26, 2013.

[38] J. Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

[39] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[40] A. Jasra, D. A. Stephens, A. Doucet, and T. Tsagaris. Inference for lévy-driven stochastic volatility models via adaptive sequential monte carlo. *Scandinavian Journal of Statistics*, 38(1):1–22, 2011.

[41] M. Kim, K. Seong, D. Woo, S. Ahn, and M. Kim. On scalable and efficient training of diffusion samplers. *arXiv preprint arXiv:2505.19552*, 2025.

[42] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, Jan. 2017.

[43] L. Klein, A. Krämer, and F. Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910, 2023.

[44] L. Klein and F. Noé. Transferable boltzmann generators. *Advances in Neural Information Processing Systems*, 37:45281–45314, 2024.

[45] J. Köhler, M. Invernizzi, P. De Haan, and F. Noé. Rigid body flows for sampling molecular crystal structures. In *International Conference on Machine Learning*, pages 17301–17326. PMLR, 2023.

[46] J. Köhler, A. Krämer, and F. Noé. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021.

[47] A. N. Kolmogorov and G. Castelnuovo. *Sur la notion de la moyenne*. G. Bardi, tip. della R. Accad. dei Lincei, 1930.

[48] G.-H. Liu, J. Choi, Y. Chen, B. K. Miller, and R. T. Chen. Adjoint schr\" odinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025.

[49] W. Meng, Q. Zheng, Y. Shi, and G. Pan. An off-policy trust region policy optimization method with monotonic improvement guarantee for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2223–2235, 2021.

[50] L. Midgley, V. Stimper, J. Antorán, E. Mathieu, B. Schölkopf, and J. M. Hernández-Lobato. Se (3) equivariant augmented coupling flows. *Advances in Neural Information Processing Systems*, 36:79200–79225, 2023.

[51] L. I. Midgley, V. Stimper, G. N. Simm, B. Schölkopf, and J. M. Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.

[52] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016.

[53] R. M. Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[54] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[55] F. Noé, S. Olsson, J. Köhler, and H. Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[56] F. Noé and co workers. bgflow: Boltzmann generators and normalizing flows in pytorch, 2025. Commit fbba56fac3eb88f6825d2bd4f745ee75ae9715e1.

[57] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih. Pgq: Combining policy gradient and q-learning. *CoRR*, 2016.

[58] F. Otto, P. Becker, N. A. Vien, H. C. Ziesche, and G. Neumann. Differentiable trust region layers for deep reinforcement learning. *arXiv preprint arXiv:2101.09207*, 2021.

[59] J. Pajarinen, H. L. Thai, R. Akrour, J. Peters, and G. Neumann. Compatible natural gradient policy search. *Machine Learning*, 108:1443–1466, 2019.

[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[61] J. Peters, K. Mulling, and Y. Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1607–1612, 2010.

[62] D. J. Rezende, G. Papamakarios, S. Racaniere, M. Albergo, G. Kanwar, P. Shanahan, and K. Cranmer. Normalizing Flows on Tori and Spheres. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8083–8092. PMLR, Nov. 2020.

[63] H. Schopmans and P. Friederich. Conditional Normalizing Flows for Active Learning of Coarse-Grained Molecular Representations. In *Forty-First International Conference on Machine Learning*, June 2024.

[64] H. Schopmans and P. Friederich. Temperature-annealed boltzmann generators. In *Forty-second International Conference on Machine Learning*, 2025.

[65] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[66] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[67] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[68] V. Stimper, L. I. Midgley, G. N. C. Simm, B. Schölkopf, and J. M. Hernández-Lobato. Alanine dipeptide in an implicit solvent at 300K. Aug. 2022.

[69] V. Stimper, B. Schölkopf, and J. M. Hernandez-Lobato. Resampling Base Distributions of Normalizing Flows. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 4915–4936. PMLR, May 2022.

[70] G. Stoltz, M. Rousset, et al. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[71] M. Struwe and M. Struwe. *Variational methods*, volume 991. Springer, 2000.

[72] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient natural evolution strategies. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 539–546, 2009.

[73] S. Syed, A. Bouchard-Côté, K. Chern, and A. Doucet. Optimised annealed sequential monte carlo samplers. *arXiv preprint arXiv:2408.12057*, 2024.

[74] C. B. Tan, J. Bose, C. Lin, L. Klein, M. M. Bronstein, and A. Tong. Scalable Equilibrium Sampling with Sequential Boltzmann Generators. In *Forty-Second International Conference on Machine Learning*, June 2025.

[75] C. B. Tan, M. Hassan, L. Klein, S. Syed, D. Beaini, M. M. Bronstein, A. Tong, and K. Neklyudov. Amortized sampling with transferable normalizing flows. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

[76] D. Thalmeier, H. J. Kappen, S. Totaro, and V. Gómez. Adaptive smoothing for path integral control. *Journal of Machine Learning Research*, 21(191):1–37, 2020.

[77] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[78] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[79] C. Williams, A. Campbell, A. Doucet, and S. Syed. Score-optimal diffusion schedules. *Advances in Neural Information Processing Systems*, 37:107960–107983, 2024.

[80] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in neural information processing systems*, 30, 2017.

[81] H. Xu, J. Xuan, G. Zhang, and J. Lu. Trust region policy optimization via entropy regularization for kullback–leibler divergence constraint. *Neurocomputing*, 589:127716, 2024.

[82] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

[83] L. Zhang. The cosine schedule is fisher-rao-optimal for masked discrete diffusion models. *arXiv preprint arXiv:2508.04884*, 2025.

[84] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

Appendix

A Proofs

Proof of Props. 2.1 to 2.3. We divide the proof into two parts and start with the most general formulation using both constraints, referring to Prop. 2.3. Part 2 will then derive the solution with just the trust-region (Prop. 2.1) and just the entropy constraint (Prop. 2.2), as they can be interpreted as special cases of the general formulation.

Part 1: Consider the constrained optimization problem

$$q_{i+1} = \arg\min_{q \in \mathcal{P}(\mathbb{R}^d)} D_{\mathrm{KL}}(q|p) \quad \text{s.t.} \quad D_{\mathrm{KL}}(q|q_i) \leq \varepsilon_{\mathrm{tr}}, \quad H(q_i) - H(q) \leq \varepsilon_{\mathrm{ent}}, \quad \int \mathrm{d}q = 1. \tag{17}$$

and its corresponding Lagrangian

$$\mathcal{L}_{\mathrm{tr}}^{(i+1)}(q, \lambda, \eta, \omega) = D_{\mathrm{KL}}(q|p) + \lambda(D_{\mathrm{KL}}(q|q_i) - \varepsilon_{\mathrm{tr}}) + \eta(H(q_i) - H(q) - \varepsilon_{\mathrm{ent}}) + \omega \left(\int \mathrm{d}q - 1 \right). \tag{18}$$

Using the convexity of the Kullback-Leibler (KL) divergence in its arguments, the convexity of the negative Shannon entropy, and that the integral is a linear functional, the objective from Eq. (17) and its Lagrangian, given by Eq. (18) are convex in q. Using that $\mathcal{P}(\mathbb{R}^d)$ is continuous, there always exists a measure $\tilde{q} \neq q_i$ with $D_{\mathrm{KL}}(\tilde{q}|q_i) < \varepsilon_{\mathrm{tr}}$ and $H(q_i) - H(\tilde{q}) < \varepsilon_{\mathrm{ent}}$ that satisfies the inequality constraints strictly. Following [16, 5.2.3], Slater's condition holds, implying strong duality. We can therefore instead solve the dual problem.

We start by setting up the Euler-Lagrange equation, given by

$$\frac{\partial}{\partial q} \mathcal{L}_{\mathrm{tr}}^{(i+1)}(q, \lambda, \eta, \omega) = 0,$$

using

$$\begin{aligned} \mathcal{L}_{\mathrm{tr}}^{(i+1)}(q, \lambda, \eta, \omega) = \int q(x) \big((1 + \lambda + \eta) \log q(x) - \log p(x) - \lambda \log q_i(x) + \omega \big) \mathrm{d}x \\ - \lambda \varepsilon_{\mathrm{tr}} + \eta(H(q_i) - \varepsilon_{\mathrm{ent}}) - \omega \end{aligned}$$

and solve for q. Hence, we get

$$\begin{aligned} \frac{\partial}{\partial q} \mathcal{L}_{\mathrm{tr}}^{(i+1)}(q, \lambda, \eta, \omega) = (1 + \lambda + \eta)(\log q + 1) - \log p - \lambda \log q_i + \omega = 0 \\ \Leftrightarrow \log q = \log \left(q_i^{\frac{\lambda}{1+\lambda+\eta}} p^{\frac{1}{1+\lambda+\eta}} \right) - \left(\frac{\omega}{1 + \lambda + \eta} + 1 \right). \end{aligned} \tag{19}$$

Using this, we can further determine ω using

$$\begin{aligned} &\int \mathrm{d}q = \int q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} p(x)^{\frac{1}{1+\lambda+\eta}} \mathrm{d}x / \exp\left(\frac{\omega}{1 + \lambda + \eta} + 1 \right) = 1 \\ &\Leftrightarrow \omega = (1 + \lambda + \eta)(\log \bar{\mathcal{Z}}_{i+1}(\lambda, \eta) - 1), \quad \text{with} \quad \bar{\mathcal{Z}}_{i+1}(\lambda, \eta) = \int q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} p(x)^{\frac{1}{1+\lambda+\eta}}(x) \mathrm{d}x. \end{aligned}$$

Substituting ω back into (19) and simplifying the fraction using $\tilde{p} = \mathcal{Z} p$ yields

$$q_{i+1}(x, \lambda, \eta) = \frac{q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} \tilde{p}(x)^{\frac{1}{1+\lambda+\eta}}}{\mathcal{Z}_{i+1}(\lambda, \eta)} \quad \text{with} \quad \mathcal{Z}_{i+1}(\lambda, \eta) = \int q_i(x)^{\frac{\lambda}{1+\lambda+\eta}} \tilde{p}(x)^{\frac{1}{1+\lambda+\eta}}(x) \mathrm{d}x,$$

which uses the unnormalized target $\tilde{p}$, proving Prop. 2.3.

Part 2: Setting $\varepsilon_{\mathrm{ent}} = \infty$ or $\varepsilon_{\mathrm{tr}} = \infty$ effectively deactivates the respective constraint, yielding the trust-region objective (2) or the entropy objective (7) respectively. This is equivalent to setting the Lagrangian multipliers $\eta = 0$ or $\lambda = 0$, proving Prop. 2.1 and Prop. 2.2 respectively. □

Proof of Thm. 2.4. We divide the proof into three parts and start with the most general formulation using both constraints. The first two parts will show form and monotonicity while part three will derive the special cases with just the trust-region and just the entropy constraint.

Part 1: Given are the sequences of Lagrangian multipliers $(\lambda_i)_{i \in \mathbb{N}_0} \geq 0$ and $(\eta_i)_{i \in \mathbb{N}_0} \geq 0$. We now aim to proof that the sequence $(q_i)_{i \in \mathbb{N}_0}$, given by

$$\tilde{q}_i = \begin{cases} q_0 & , \quad i = 0 \\ \tilde{q}_{i-1}^{\frac{\lambda_{i-1}}{1+\lambda_{i-1}+\eta_{i-1}}} \tilde{p}^{\frac{1}{1+\lambda_{i-1}+\eta_{i-1}}} & , \quad i \geq 1 \end{cases}, \tag{20}$$

takes the form

$$\tilde{q}_i = q_0^{1-\beta_i}(\tilde{p}^{\alpha_i})^{\beta_i} \quad \text{with} \quad \beta_i = 1 - \prod_{j=0}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j}$$

$$\text{and} \quad \alpha_i = \begin{cases} 0 & , \quad i = 0 \\ 1 - \frac{1}{\beta_i} \sum_{k=0}^{i-1} \frac{\eta_k}{1+\lambda_k+\eta_k} \prod_{j=k+1}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j} & , \quad i \geq 1. \end{cases}$$

We use the common convention that empty products evaluate to one.

Base case ($i = 0$): The simplest case

$$\tilde{q}_0 = q_0^{1-\beta_0}(\tilde{p}^{\alpha_0})^{\beta_0}$$

holds due to $\beta_0 = 0$ (using empty product convention).

Inductive step: We start from Eq. (20) and transform it using the assumption that $\tilde{q}_i = q_0^{1-\beta_i}(\tilde{p}^{\alpha_i})^{\beta_i}$ holds for some arbitrary but fixed $i \in \mathbb{N}_0$, yielding

$$\begin{aligned} \tilde{q}_{i+1} &= \tilde{q}_i^{\frac{\lambda_i}{1+\lambda_i+\eta_i}} \tilde{p}^{\frac{1}{1+\lambda_i+\eta_i}} \\ &= \left(q_0^{1-\beta_i}(\tilde{p}^{\alpha_i})^{\beta_i}\right)^{\frac{\lambda_i}{1+\lambda_i+\eta_i}} \tilde{p}^{\frac{1}{1+\lambda_i+\eta_i}} \\ &= q_0^{1-\beta_{i+1}} \tilde{p}^{\alpha_i\beta_i\frac{\lambda_i}{1+\lambda_i+\eta_i} + \frac{1}{1+\lambda_i+\eta_i}}. \end{aligned}$$

Using

$$\begin{aligned} \beta_i \frac{\lambda_i}{1+\lambda_i+\eta_i} &= \frac{\lambda_i}{1+\lambda_i+\eta_i} - \prod_{j=0}^{i} \frac{\lambda_j}{1+\lambda_j+\eta_j} \\ &= 1 - \frac{1+\eta_i}{1+\lambda_i+\eta_i} - \prod_{j=0}^{i} \frac{\lambda_j}{1+\lambda_j+\eta_j} \\ &= \beta_{i+1} - \frac{1+\eta_i}{1+\lambda_i+\eta_i}, \end{aligned}$$

we now can rewrite the exponent of p yielding

$$\begin{aligned} &\alpha_i\beta_i\frac{\lambda_i}{1+\lambda_i+\eta_i} + \frac{1}{1+\lambda_i+\eta_i} \\ &= \left(\beta_i - \sum_{k=0}^{i-1} \frac{\eta_k}{1+\lambda_k+\eta_k} \prod_{j=k+1}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j}\right) \frac{\lambda_i}{1+\lambda_i+\eta_i} + \frac{1}{1+\lambda_i+\eta_i} \\ &= \beta_{i+1} - \frac{1+\eta_i}{1+\lambda_i+\eta_i} - \sum_{k=0}^{i-1} \frac{\eta_k}{1+\lambda_k+\eta_k} \prod_{j=k+1}^{i} \frac{\lambda_j}{1+\lambda_j+\eta_j} + \frac{1}{1+\lambda_i+\eta_i} \\ &= \beta_{i+1} - \sum_{k=0}^{i} \frac{\eta_k}{1+\lambda_k+\eta_k} \prod_{j=k+1}^{i} \frac{\lambda_j}{1+\lambda_j+\eta_j}, \\ &= \alpha_{i+1}\beta_{i+1}. \end{aligned}$$

again using the convention that an empty product evaluates to one. It directly follows

$$q_{i+1} \propto \tilde{q}_{i+1} = {q_0}^{1-\beta_{i+1}}(\tilde{p}^{\alpha_{i+1}})^{\beta_{i+1}},$$

which completes the induction.

Part 2: It remains to show that $(\alpha_i)_{i\in\mathbb{N}_0}$ and $(\beta_i)_{i\in\mathbb{N}_0}$ take values in $[0, 1]$ and are monotonically increasing with $\alpha_0 = \beta_0 = 0$ and $\alpha_I = \beta_I = 1$ for some $I \in \mathbb{N}_+$.

The first case ($\alpha_0 = \beta_0 = 0$) holds by definition. Assuming that there exists an $I \in \mathbb{N}_+$, such that $\lambda_{I-1} = \eta_{I-1} = 0$,

$$\beta_i = 1 - \prod_{j=0}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j} \overset{i\geq I}{=} 1$$

and

$$\alpha_i = 1 - \frac{1}{\beta_i} \sum_{k=0}^{i-1} \frac{\eta_k}{1+\lambda_k+\eta_k} \prod_{j=k+1}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j} \overset{i\geq I}{=} 1$$

follow directly for all $i \geq I$. Both the trust-region and entropy constraints become inactive at the optimum and do not prevent $(q_i)_{i\in\mathbb{N}0}$ from reaching the unique optimum p ($\varepsilon_{\mathrm{tr}}, \varepsilon_{\mathrm{ent}} > 0$). Consequently, both Lagrange multipliers will eventually vanish, motivating the existence of some $I \in \mathbb{N}+$, such that $\lambda_{I-1} = \eta_{I-1} = 0$.

We now show monotonicity of $(\beta_i)_{i\in\mathbb{N}_0}$ using $(\lambda_i)_{i\in\mathbb{N}_0} \geq 0$ and $(\eta_i)_{i\in\mathbb{N}_0} \geq 0$. Let $i \in \mathbb{N}_0$ be arbitrary. As a direct consequence of

$$\begin{aligned}
\beta_{i+1} - \beta_i &= \prod_{j=0}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j} - \prod_{j=0}^{i} \frac{\lambda_j}{1+\lambda_j+\eta_j} \\
&= \left(\prod_{j=0}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j}\right)\left(1 - \frac{\lambda_i}{1+\lambda_i+\eta_i}\right) \\
&= \left(\prod_{j=0}^{i-1} \frac{\lambda_j}{1+\lambda_j+\eta_j}\right)\left(\frac{1+\eta_i}{1+\lambda_i+\eta_i}\right) \overset{\substack{\lambda_j,\eta_j\geq 0 \\ \forall j\in\mathbb{N}_0}}{\geq} 0,
\end{aligned}$$

the sequence $(\beta_i)_{i\in\mathbb{N}_0}$ must be monotonically increasing.

Part 3: We now consider the special cases in which only the trust-region constraint or the entropy constraint is active by setting the Lagrange multiplier sequence of the other constraint to zero.

We first consider only the trust-region constraint (2), which corresponds to setting the Lagrangian multiplier of the entropy constraint to zero, i.e., $\eta_i = 0$ for all $i \in \mathbb{N}_0$. In this scenario, α_i simplifies to $\alpha_0 = 0$ and $\alpha_i = 1$ for all $i \geq 1$. Consequently, and using $\beta_0 = 0$, the iterates take the form

$$q_i \propto \tilde{q}_i = q_0^{1-\beta_i} \tilde{p}^{\beta_i}, \quad i \in \mathbb{N}_0,$$

as claimed.

Analogously, the trust-region constraint can be rendered inactive by setting $\lambda_i = 0$ for all $i \in \mathbb{N}_0$, leaving only the entropy constraint active, corresponding to Eq. (7). In this case, $\beta_0 = 0$ and $\beta_i = 1$ for all $i \geq 1$, yielding

$$q_i \propto \tilde{q}_i = \begin{cases} q_0 & , \quad i = 0, \\ \tilde{p}^{\alpha_i} & , \quad i \geq 1, \end{cases}$$

which concludes the proof. □

Proof of bounded importance-weight variance its effect on the effective sample size. In this section, we show that using the trust-region constraint yields an approximate lower bound for the effective sample between any two consecutive distributions q_i and q_{i+1}. This approximate lower bound only depends on $\varepsilon_{\mathrm{tr}}$.

The variance of the importance weights

$$\frac{q_{i+1}(x)}{q_i(x)} = \begin{cases} \frac{1}{\mathcal{Z}_{i+1}(\lambda_i)} \left(\frac{\tilde{p}(x)}{q_i(x)}\right)^{\frac{1}{1+\lambda_i}} & \text{with trust-region constraint (2)} \\ \frac{1}{\mathcal{Z}_{i+1}(\lambda_i,\eta_i)} \left(\frac{\tilde{p}(x)}{q_i(x)^{1+\eta_i}}\right)^{\frac{1}{1+\lambda_i+\eta_i}} & \text{with trust-region + entropy constraint (9)} \end{cases}$$

between two normalized consecutive distributions is closely connected to the effective sample size via

$$\mathrm{ESS}(q_i, q_{i+1}) = \frac{1}{1 + \mathrm{Var}_{q_i}\left(\frac{q_{i+1}(x)}{q_i(x)}\right)},$$

also explained in App. C.4. The relation $\mathrm{Var}_{q_i}(q_{i+1}(x)/q_i(x)) = \chi^2(q_{i+1}|q_i)$ [24] and the well-known Taylor approximation $\chi^2(q_{i+1}|q_i) \approx 2D_{\mathrm{KL}}(q_{i+1}|q_i)$ [26] lets use rewrite the effective sample size in terms of the KL divergence between q_{i+1} and q_i yielding

$$\mathrm{ESS}(q_i, q_{i+1}) \approx \frac{1}{1 + 2D_{\mathrm{KL}}(q_{i+1}|q_i)}$$

as approximation for the effective sample size. This approximation is justified under the assumption that q_{i+1} is close to q_i, a condition that is satisfied by the design of the problem for a small trust-region bound $\varepsilon_{\mathrm{tr}} > 0$. Due to q_{i+1} being the optimal solution to an objective with the constraint $D_{\mathrm{KL}}(q|q_i) \leq \varepsilon_{\mathrm{tr}}$, the constraint must also hold for $q = q_{i+1}$ resulting in the approximate lower bound

$$\mathrm{ESS}(q_i, q_{i+1}) \gtrapprox \frac{1}{1 + 2\varepsilon_{\mathrm{tr}}} \tag{21}$$

for the effective sample size of the importance weights with equality in all but the last step.

This approximate lower bound justifies the use of Monte Carlo approximations in Sec. 3, helping to stabilize training independent of the problem's dimensionality. □

Proof of uniqueness and tightness of the trust-region solution. Closely following [14], we now establish the uniqueness of the trust-region solution and show that the trust-region constraint is tight for all but the the final step. Specifically, we show

$$D_{\mathrm{KL}}(q_i|p) < \varepsilon_{\mathrm{tr}} \implies q_i = p$$
$$q_i = \arg\min D_{\mathrm{KL}}(q|p) \quad \text{s.t.} \quad D_{\mathrm{KL}}(q|q_{i-1}) \leq \varepsilon_{\mathrm{tr}} \quad \text{is unique}$$

If $D_{\mathrm{KL}}(q_i \,\|\, p) < \varepsilon_{\mathrm{tr}}$, the KKT conditions imply that the Lagrangian multiplier satisfies $\lambda_i = 0$, so the constraint is inactive. Consequently, q_i must solve the strictly convex unconstrained problem

$$\min_{q \in \mathcal{P}(\mathbb{R}^d)} D_{\mathrm{KL}}(q|p),$$

which has the unique minimizer p. Since p is feasible ($D_{\mathrm{KL}}(p|p) = 0 \leq \varepsilon_{\mathrm{tr}}$), it follows that $q_i = p$.

Uniqueness of q_i further follows from the convexity of the feasible set $\{q \in \mathcal{P} | D_{\mathrm{KL}}(q|q_i) \leq \varepsilon_{\mathrm{tr}}\}$ together with the strict convexity of the objective in q when p is fixed. □

B Extended numerical evaluation

System	Method	Target Evals ↓	NLL ↓	ESS [%] ↑	Ram KL ↓	Ram KL w. RW ↓
Alanine Dipeptide ($d = 60$)	Forward KL	5×10^9	-213.581 ± 0.000	(82.16 ± 0.09) %	$(2.21 \pm 0.05) \times 10^{-3}$	$(1.99 \pm 0.07) \times 10^{-3}$
	Reverse KL	2.56×10^8	-213.609 ± 0.006	(94.11 ± 0.21) %	$(1.75 \pm 0.28) \times 10^{-2}$	$(1.65 \pm 0.29) \times 10^{-2}$
	FAB	2.13×10^8	-213.653 ± 0.000	(94.81 ± 0.04) %	$(1.50 \pm 0.03) \times 10^{-3}$	$\mathbf{(1.25 \pm 0.01) \times 10^{-3}}$
	TA-BG	1×10^8	-213.666 ± 0.001	(95.77 ± 0.12) %	$(1.94 \pm 0.07) \times 10^{-3}$	$(1.36 \pm 0.02) \times 10^{-3}$
	CMT (ours)	1×10^8	$\mathbf{-213.677 \pm 0.000}$	$\mathbf{(97.69 \pm 0.03)}$ %	$\mathbf{(1.49 \pm 0.02) \times 10^{-3}}$	$(1.41 \pm 0.02) \times 10^{-3}$
Alanine Tetra-Peptide ($d = 120$)	Forward KL	4.2×10^9	-330.069 ± 0.001	(45.29 ± 0.08) %	$(2.26 \pm 0.06) \times 10^{-3}$	$(2.50 \pm 0.03) \times 10^{-3}$
	Reverse KL	2.56×10^8	-329.191 ± 0.122	(74.88 ± 3.65) %	$(3.00 \pm 0.35) \times 10^{-1}$	$(2.87 \pm 0.40) \times 10^{-1}$
	FAB	2.13×10^8	-330.100 ± 0.002	(63.59 ± 0.23) %	$(6.89 \pm 0.25) \times 10^{-3}$	$\mathbf{(1.25 \pm 0.01) \times 10^{-3}}$
	TA-BG	1×10^8	-330.143 ± 0.002	(64.87 ± 0.21) %	$(2.47 \pm 0.23) \times 10^{-3}$	$(1.71 \pm 0.07) \times 10^{-3}$
	CMT (ours)	1×10^8	$\mathbf{-330.167 \pm 0.002}$	$\mathbf{(68.49 \pm 0.14)}$ %	$\mathbf{(1.99 \pm 0.05) \times 10^{-3}}$	$(1.65 \pm 0.07) \times 10^{-3}$
Alanine Hexa-Peptide ($d = 180$)	Forward KL	4.2×10^9	-501.598 ± 0.005	(10.97 ± 0.11) %	$(4.16 \pm 0.26) \times 10^{-3}$	$(7.69 \pm 0.03) \times 10^{-3}$
	Reverse KL	2.56×10^8	-497.378 ± 0.277	(22.22 ± 1.44) %	$(5.41 \pm 0.38) \times 10^{-1}$	$(5.32 \pm 0.38) \times 10^{-1}$
	FAB	4.2×10^8	-501.268 ± 0.008	(14.64 ± 0.08) %	$(2.09 \pm 0.02) \times 10^{-2}$	$(1.12 \pm 0.02) \times 10^{-2}$
	TA-BG	4×10^8	-501.582 ± 0.010	(15.89 ± 0.13) %	$\mathbf{(8.33 \pm 0.59) \times 10^{-3}}$	$\mathbf{(8.15 \pm 0.44) \times 10^{-3}}$
	CMT (ours)	4×10^8	$\mathbf{-501.761 \pm 0.008}$	$\mathbf{(29.20 \pm 0.24)}$ %	$(1.25 \pm 0.04) \times 10^{-2}$	$(1.21 \pm 0.01) \times 10^{-2}$
ELIL Tetra-Peptide ($d = 219$)	Forward KL	4.2×10^9	-597.571 ± 0.004	(5.80 ± 0.04) %	$(4.12 \pm 0.03) \times 10^{-3}$	$(9.38 \pm 0.06) \times 10^{-3}$
	Reverse KL	2.56×10^8	-583.381 ± 3.033	(1.25 ± 0.45) %	$(1.22 \pm 0.28) \times 10^{0}$	$(1.14 \pm 0.31) \times 10^{0}$
	FAB	8.43×10^8	-597.370 ± 0.006	(7.30 ± 0.08) %	$(2.56 \pm 0.10) \times 10^{-2}$	$(9.01 \pm 0.57) \times 10^{-3}$
	TA-BG	8×10^8	-597.830 ± 0.022	(10.12 ± 0.38) %	$(7.35 \pm 0.89) \times 10^{-3}$	$(7.65 \pm 0.59) \times 10^{-3}$
	CMT	8×10^8	$\mathbf{-598.440 \pm 0.003}$	$\mathbf{(25.91 \pm 0.19)}$ %	$\mathbf{(5.74 \pm 0.05) \times 10^{-3}}$	$\mathbf{(5.22 \pm 0.10) \times 10^{-3}}$

Table 2: Comparison of metrics obtained for all four peptide systems. The table reports the number of target evaluations (Target Evals), the negative log-likelihood (NLL), the effective sample size (ESS), the average forward KL divergence to the ground-truth Ramachandran plots (Ram KL) and its importance-weighted version (Ram KL w. RW.), along with the corresponding system and method. All values are presented as the mean and standard error across four independent experiments. The best-performing variational method for each metric is highlighted in bold. Reverse KL was excluded, as it tends to suffer from mode collapse, making ESS values not directly comparable.

Complementing the results of Sec. 5, this section reports additional metrics for the main method comparison in Tab. 2 (see also the corresponding Ramachandran plots in Fig. 4), an ablation study on the effect of both constraints (see Tab. 3 and Fig. 5), and an ablation study on the trust-region constraint and its effect on bounding importance-weight variance across different system sizes and trust-region bounds $\varepsilon_{\mathrm{tr}}$ (see Fig. 6).

Main results. We begin with Tab. 2, which introduces two additional metrics (RAM KL and RAM KL w. RW) to quantify the discrepancy between the ground truth and method-generated Ramachandran plots. Substantially elevated RAM KL values serve as robust indicators of mode collapse, as exemplified by the results for reverse KL training, where the RAM KL values are consistently at least an order of magnitude higher than those observed for other methods. Corresponding Ramachandran plots for the different methods are shown in Fig. 4.

Ablation study for constraints. Tab. 3 presents the performance of our method under different configurations, with the trust-region and entropy constraints selectively enabled or disabled. In

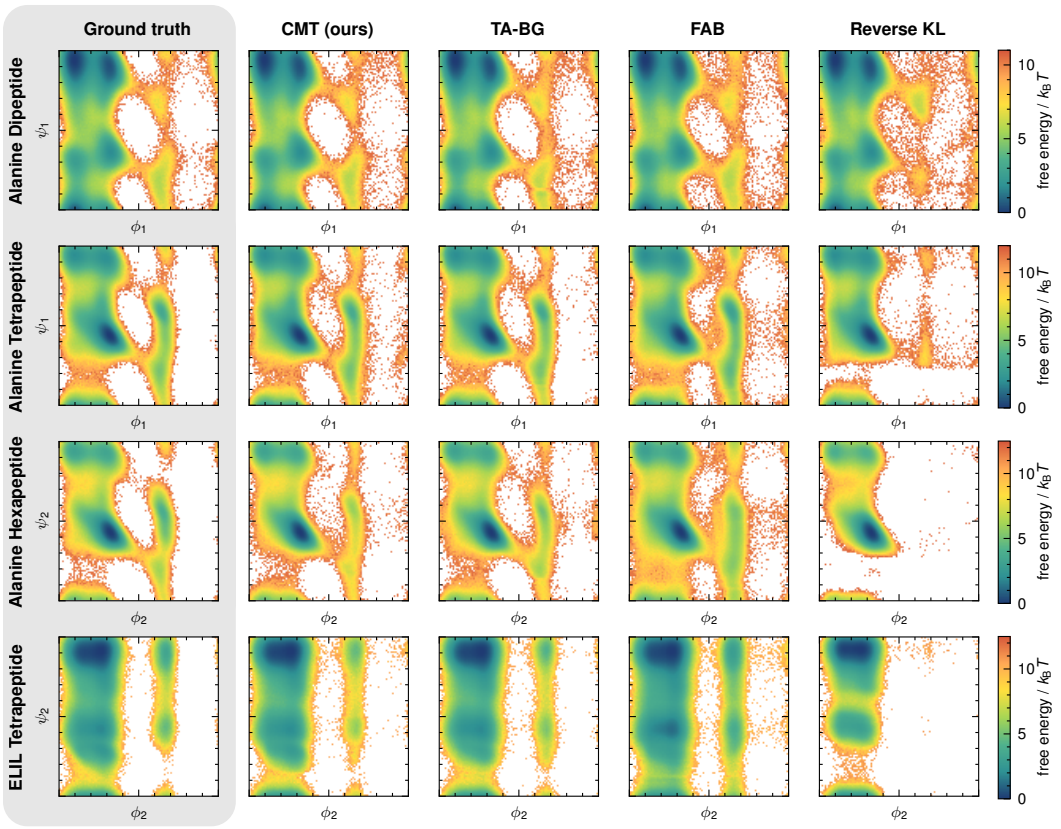


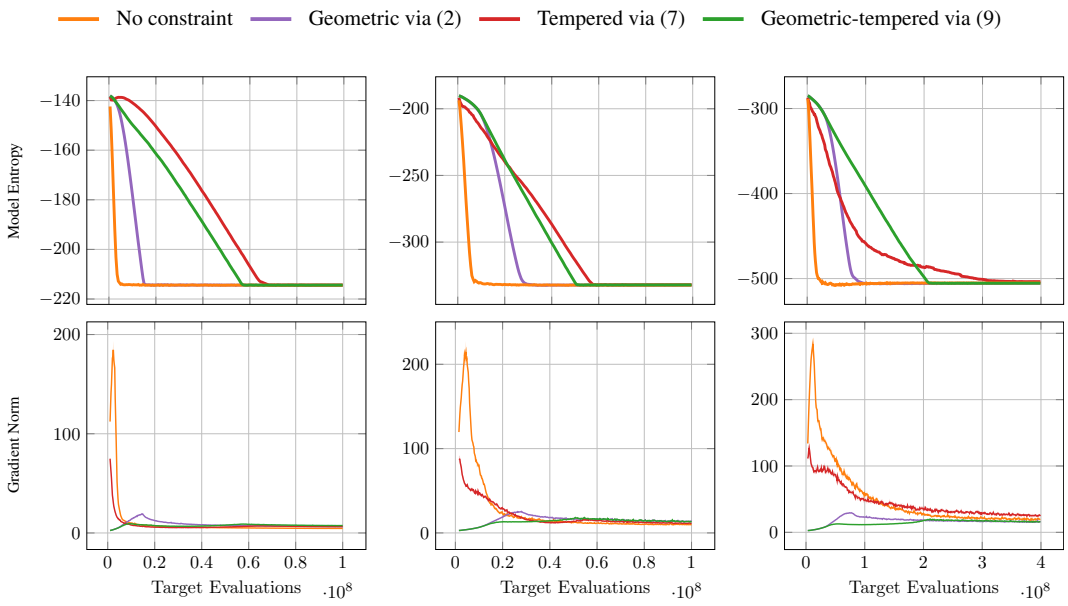
Figure 4: Comparison of Ramachandran plots of backbone dihedral angle pairs obtained with different methods. See App. C.4 for details.

System	Constraint: Trust-Region	Constraint: Entropy	Target Evals ↓	NLL ↓	ESS [%] ↑ ★	Ram KL ↓	Ram KL w. RW ↓
Alanine Dipeptide ($d = 60$)	✗	✗	1×10^8	$\mathbf{-213.667 \pm 0.000}$	(97.71 ± 0.01) %	$(1.50 \pm 0.02) \times 10^{-3}$	$(1.50 \pm 0.01) \times 10^{-3}$
	✓	✗	1×10^8	-213.673 ± 0.000	(97.46 ± 0.02) %	$(1.52 \pm 0.03) \times 10^{-3}$	$(1.47 \pm 0.01) \times 10^{-3}$
	✗	✓	1×10^8	$\mathbf{-213.677 \pm 0.000}$	(97.76 ± 0.04) %	$(1.50 \pm 0.02) \times 10^{-3}$	$\mathbf{(1.38 \pm 0.02) \times 10^{-3}}$
	✓	✓	1×10^8	$\mathbf{-213.677 \pm 0.000}$	(97.69 ± 0.03) %	$\mathbf{(1.49 \pm 0.02) \times 10^{-3}}$	$(1.41 \pm 0.02) \times 10^{-3}$
Alanine Tetra-Peptide ($d = 120$)	✗	✗	1×10^8	-329.787 ± 0.193	(69.60 ± 2.20) %	$(6.92 \pm 3.44) \times 10^{-2}$	$(6.64 \pm 3.25) \times 10^{-2}$
	✓	✗	1×10^8	-330.154 ± 0.001	(66.11 ± 0.02) %	$(2.08 \pm 0.04) \times 10^{-3}$	$(2.05 \pm 0.04) \times 10^{-3}$
	✗	✓	1×10^8	-330.130 ± 0.003	(63.34 ± 0.24) %	$(2.27 \pm 0.04) \times 10^{-3}$	$(1.77 \pm 0.02) \times 10^{-3}$
	✓	✓	1×10^8	$\mathbf{-330.167 \pm 0.002}$	(68.49 ± 0.14) %	$\mathbf{(1.99 \pm 0.05) \times 10^{-3}}$	$\mathbf{(1.65 \pm 0.07) \times 10^{-3}}$
Alanine Hexa-Peptide ($d = 180$)	✗	✗	4×10^8	-499.746 ± 0.183	(25.63 ± 1.05) %	$(2.52 \pm 0.33) \times 10^{-1}$	$(2.55 \pm 0.36) \times 10^{-1}$
	✓	✗	4×10^8	-501.322 ± 0.232	(33.45 ± 2.45) %	$(4.21 \pm 1.34) \times 10^{-2}$	$(4.14 \pm 1.40) \times 10^{-2}$
	✗	✓	4×10^8	-501.333 ± 0.017	(15.03 ± 1.00) %	$(1.27 \pm 0.08) \times 10^{-2}$	$(1.39 \pm 0.11) \times 10^{-2}$
	✓	✓	4×10^8	$\mathbf{-501.761 \pm 0.008}$	(29.20 ± 0.24) %	$\mathbf{(1.25 \pm 0.04) \times 10^{-2}}$	$\mathbf{(1.21 \pm 0.01) \times 10^{-2}}$

Table 3: Performance of CMT with the trust-region and entropy constraints selectively enabled or disabled. For all metrics except ESS, the best-performing variant is highlighted in bold. ★ The reverse ESS is only partially comparable, as training without the entropy constraint led to partial mode collapse on alanine tetrapeptide and alanine hexapeptide, rendering direct comparisons unreliable. To avoid confusion, no best-performing method is highlighted in bold for ESS.

addition to the alanine hexapeptide results shown in the main paper, we also report results for alanine dipeptide and alanine tetrapeptide. The absence of both constraints effectively corresponds to importance-weighted forward KL training. Considering the NLL, which serves as a forward metric, it becomes clear that both constraints are necessary to achieve optimal performance. Variants of the method without the entropy constraint exhibited at least partial mode collapse, rendering the ESS largely incomparable. Partial mode collapse is further reflected in the RAM KL and RAM KL w. RW. metrics, which take substantially higher values when the entropy constraint is omitted.

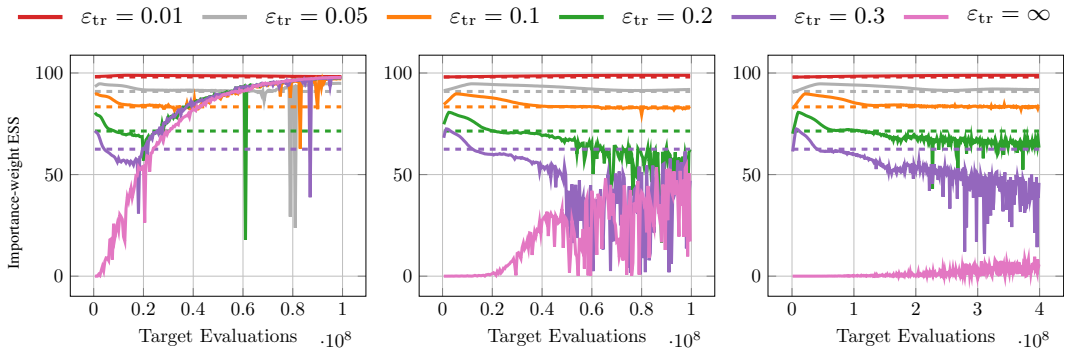
Fig. 5 depicts the evolution of model entropy and the gradient norm (prior to clipping) during training across different systems. Training with only the entropy constraint yields an approximately



(a) Alanine Dipeptide (b) Alanine Tetrapeptide (c) Alanine Hexapeptide

Figure 5: Effect of trust-region and entropy constraint on the model entropy (top row) and the gradient norm (bottom row) across different molecular systems.

linear decay of entropy for both alanine dipeptide and alanine hexapeptide. In the case of alanine hexapeptide, however, the entropy constraint is noticeably violated, likely due to the system's higher dimensionality and the pronounced discrepancy between the initial model distribution q_0 and the first intermediate distribution q_1. Larger system sizes also tend to increase the gradient norm, most prominently in alanine hexapeptide. The combination of the trust-region and entropy constraints produces the most stable gradient norms, while the approximately linear entropy decay indicates that the entropy constraint is effectively enforced, thereby enabling its practical application even in the case of alanine hexapeptide. By contrast, the trust-region constraint alone leads to a more rapid entropy collapse, which reduces exploration and ultimately limits the algorithm's final performance in practice.



(a) Alanine Dipeptide (b) Alanine Tetrapeptide (c) Alanine Hexapeptide

Figure 6: Importance-weight variance between successive intermediate distributions, shown in terms of effective sample size (ESS), for different trust-region bounds and system sizes. Each trust-region bound ε_{tr} defines an approximate lower bound on the ESS, indicated by dashed lines.

Ablation study on the trust-region bound. Fig. 6 illustrates the importance-weight variance of CMT across different trust-region bounds and system sizes, highlighting the approximate direct relationship between the trust-region bound and the variance of importance weights between consecutive intermediate distributions. Importance-weight variance is expressed in terms of effective

sample size (ESS). In the absence of a trust-region constraint ($\varepsilon_{\mathrm{tr}} = \infty$), the ESS decreases with increasing system size. By contrast, finite trust-region bounds yield a substantially larger and more stable ESS, with the approximate lower bound on the ESS becoming increasingly well realized as the trust-region bound $\varepsilon_{\mathrm{tr}}$ decreases. Notably, this approximate lower bound is independent of the problem's dimensionality, a property that is empirically supported.

C Experimental setup

C.1 Architecture

Our normalizing flow architecture closely follows the one used in previous works [51, 63, 64]. We represent the conformations of the studied molecular systems using internal coordinates based on bond lengths, angles, and dihedral angles.

We use 8 pairs of neural spline coupling layers based on monotonic rational-quadratic splines [31]. The splines map from $[0, 1]$ to $[0, 1]$ using 8 bins. We use a random mask to select transformed and conditioned dimensions in the first coupling of each pair, and the corresponding inverted mask for the second coupling. The dihedral angle dimensions are modeled with circular splines [62] to respect their topology, with a random (fixed) periodic shift applied after each coupling layer. The parameter networks that calculate the spline parameters in each coupling are fully connected neural networks with hidden dimensions $[256, 256, 256, 256, 256]$ and ReLU activation functions. To capture their periodicity, dihedral angles ψ_i are encoded as $(\cos\psi_i, \sin\psi_i)$ when passing them to the parameter network.

As the base distribution of the normalizing flow, we use a uniform distribution in $[0, 1]$ for the dihedral angles and a Gaussian truncated to $[0, 1]$ with mean $\mu = 0.5$ and standard deviation $\sigma = 0.1$ for the bond lengths and angles.

We follow [64] to map the internal coordinates to the range $[0, 1]$ of the spline transformations: Dihedral angles are divided by 2π. Bond lengths and angles are shifted and scaled as $\eta_i' = (\eta_i - \eta_{i;\mathrm{min}})/\sigma + 0.5$, where $\eta_{i;\mathrm{min}}$ is obtained from a minimum energy structure after energy minimization. σ was set to $0.07\,\mathrm{nm}$ for bond lengths and 0.5730 for angle dimensions.

The studied molecular systems have two chiral forms (mirror images), L- and R-chirality, while in nature, one almost only finds the L-chirality. To constrain the generated molecular configurations to the L-chirality, we constrain the spline output ranges of the relevant dihedral angles (see [64] for details). Similarly, some atoms and groups (such as the hydrogen atoms in CH_3 groups) are permutation invariant in the force field energy parametrization, but have a preference in the ground truth molecular dynamics data due to very large barriers. Similarly to the chirality constraints, we constrain the splines such that only the permutation found in the ground truth data can be generated [64].

C.2 Target densities

The goal of all our experiments is to sample molecular systems at $300\,\mathrm{K}$. An overview of the studied molecular systems, including their force field parametrization, is given in Table 4. We explicitly note that the largest studied system, ELIL, does not contain capping groups, in contrast to the other three systems.

The energy evaluations during training were performed with the OpenMM 8.0.0 [33] CPU platform, using 18 workers in parallel.

Following previous work [51, 64], we use a regularized energy function to avoid large van der Waals energies due to atom clashes:

$$E_{\mathrm{reg.}}(E) = \begin{cases} E, & \text{if } E \le E_{\mathrm{high}}, \\ \log(E - E_{\mathrm{high}} + 1) + E_{\mathrm{high}}, & \text{if } E_{\mathrm{high}} < E \le E_{\mathrm{max}}, \\ \log(E_{\mathrm{max}} - E_{\mathrm{high}} + 1) + E_{\mathrm{high}}, & \text{if } E > E_{\mathrm{max}}. \end{cases} \tag{22}$$

We set $E_{\mathrm{high}} = 1 \times 10^8$ and $E_{\mathrm{max}} = 1 \times 10^{20}$ [51].

Ground truth datasets

We use ground truth test datasets obtained from extensive molecular dynamics simulations to calculate the metrics reported in Table 1.

NAME	SEQUENCE	NO. ATOMS	FORCE FIELD	CONSTRAINTS
ALANINE DIPEPTIDE	ACE-ALA-NME	22	AMBER FF96 WITH OBC1 IMPLICIT SOLVATION	NONE
ALANINE TETRAPEPTIDE	ACE-3·ALA-NME	42	AMBER99SB-ILDN WITH AMBER99 OBC IMPLICIT SOLVATION	HYDROGEN BOND LENGTHS
ALANINE HEXAPEPTIDE	ACE-5·ALA-NME	62	AMBER99SB-ILDN WITH AMBER99 OBC IMPLICIT SOLVATION	HYDROGEN BOND LENGTHS
ELIL	GLU-LEU-ILE-LEU	75	AMBER99SB-ILDN WITH AMBER99 OBC IMPLICIT SOLVATION	HYDROGEN BOND LENGTHS

Table 4: Overview of the molecular systems and corresponding force field parametrization.

1. For alanine dipeptide, we use the ground truth test dataset previously published by [51] [68]. Furthermore, we use the datasets published by [64] as additional validation and training datasets (for the forward KL experiments).
2. For alanine tetrapeptide and alanine hexapeptide, we use the test, validation, and training datasets published by [64].
3. To generate ground truth data for the tetrapeptide ELIL, we followed the simulation protocol by [64]. We performed two replica-exchange molecular dynamics simulations with replicas at $300\,\mathrm{K}$, $332\,\mathrm{K}$, $368\,\mathrm{K}$, $408\,\mathrm{K}$, $451\,\mathrm{K}$, and $500\,\mathrm{K}$, each using a time step of $2\,\mathrm{fs}$. Each simulation used $200\,\mathrm{ns}$ equilibration without exchanges, $200\,\mathrm{ns}$ equilibration with exchanges, and $1\,\mu\mathrm{s}$ production simulation time. One of the simulations was used for the ground truth test dataset, the other simulation for the training and validation datasets.

For each system, the test dataset contains 1×10^7 samples, and the training and validation datasets contain 1×10^6 samples.

C.3 Dual optimization in practice

The concavity of the dual functions permits the use of any suitable nonlinear optimization algorithm. For one-dimensional dual optimization, we employ the bounded Brent method [19], implemented via `scipy.optimize.minimize_scalar` [77], which is the library's default 1D algorithm due to its robustness and efficiency. A minimal working example on how a Lagrangian multiplier is estimated is given in Code Example 1. For 2D duals, we use `scipy.optimize.minimize` with the L-BFGS-B algorithm [84], one of SciPy's default quasi-Newton algorithms. There, we additionally passed the dual gradient function, which we obtained through automatic differentiation. Due to the constraints $\lambda, \eta \geq 0$, and to avoid numerical overflow, we bound both optimizers to stay within the interval $[0, 10^{10}]$. The method `scipy.optimize.minimize` requires an initial guess, which we set to 1×10^{-20}, a value chosen to be close to the lower bound.

C.4 Metrics

In this section, we present several commonly used metrics for both theoretical analysis and experimental evaluation. For more details, we refer to [15].

NEGATIVE LOG-LIKELIHOOD (NLL)

The negative-log-likelihood (NLL) is a forward metric computed between ground truth samples and the learned model distribution $\hat{q}$. It is directly related to the forward KL divergence and the evidence upper bound (EUBO) via

$$\begin{aligned} D_{\mathrm{KL}}(p|q) &= \underbrace{\mathbb{E}_{p(x)}\left[\log \frac{\tilde{p}(x)}{q(x)}\right]}_{\text{EUBO}} - \underbrace{\log \mathcal{Z}}_{\text{const. w.r.t. } q} \\ &= \underbrace{-\mathbb{E}_{p(x)}\left[\log q(x)\right]}_{\text{NLL}} - \underbrace{H(p)}_{\text{const. w.r.t. } q} . \end{aligned}$$

Code Example 1: Minimal working example of the dual optimization for objective (2).

```python
import numpy as np
import torch
from scipy.optimize import minimize_scalar

def estimate_log_Z(
    model_log_prob: torch.Tensor,
    target_log_prob: torch.Tensor,
    tr_mul: float,
) -> torch.Tensor:
    """Estimate log-partition function of next intermediate density"""
    log_N = torch.tensor(target_log_prob.shape[0]).log()
    log_iw = (target_log_prob - model_log_prob) / (1 + tr_mul)
    log_Z = torch.logsumexp(log_iw, dim=0) - log_N
    return log_Z


def find_best_kl_multiplier(
    model_log_prob: torch.Tensor,
    target_log_prob: torch.Tensor,
    eps_tr: float,
    max_multiplier: float = 1e10,
) -> float:
    """Finds the best Lagrangian multiplier by maximizing the dual"""
    # define dual function (dependent on Lagrangian multiplier)
    def dual(tr_mul: float):
        log_Z = estimate_log_Z(
            model_log_prob=model_log_prob,
            target_log_prob=target_log_prob,
            tr_mul=tr_mul,
        )
        dual_value = -(1 + tr_mul) * log_Z - tr_mul * eps_tr
        return dual_value.item()

    neg_dual = lambda mul: -dual(mul) # concave -> convex

    res = minimize_scalar(
        neg_dual,
        bounds=(0.0, max_multiplier),
        method="Bounded"
    )
    best_tr_mul = float(res.x)
    return best_tr_mul
```

The metrics NLL, EUBO and forward KL therefore only deviate from each other by an additive constant.

Effective sample size (ESS)

The effective sample size (ESS) is defined as

$$\mathrm{ESS}(a, b) = \frac{1}{1 + \mathrm{Var}_{a(x)}\left[\frac{b(x)}{a(x)}\right]}, \quad a, b \in \mathcal{P}(\mathbb{R}^d).$$

Closely following the notation of [15], the reverse ESS

$$\mathrm{ESS}(q, p) = \frac{\mathcal{Z}_r}{\mathbb{E}_{q(x)}\left[\left(\frac{\tilde{p}(x)}{q(x)}\right)^2\right]}, \quad \text{with} \quad \mathcal{Z}_r = \mathbb{E}_{q(x)}\left[\frac{\tilde{p}(x)}{q(x)}\right]$$

can be directly estimated via Monte Carlo using samples from the model q and the unnormalized target $\tilde{p}$.

Following [51, 64], for reverse ESS, we clipped the top 0.01% importance-weights, setting them to the smallest value among them for numerical reasons. Furthermore, ESS is computed using the regularized energy function, defined in Eq. (22).

Although forward ESS could be computed using samples from the target distribution, [64] found it to be extremely sensitive to the chosen clipping threshold and prone to instability. Consequently, only the reverse ESS was used, even though it may not fully capture phenomena such as mode collapse.

RAMACHANDRAN PLOTS

A Ramachandran plot visualizes the 2D log-density of the joint distribution of a pair of dihedral angles in a peptide's backbone. For more details, we refer to [64]. These plots are used to visualize a peptide's main degrees of freedom and are likely to show mode collapse if it occurs. A Ramachandran plot is effectively a histogram of the occurrence of dihedral angles and is computed solely from model or ground-truth samples.

For alanine tetrapeptide, alanine hexapeptide, and ELIL tetrapeptide, which contain multiple backbone dihedral angle pairs, we always show the pair exhibiting the most pronounced deviation from the ground truth, which is the same across methods. Among the four runs made per method in Fig. 4, we selected the one with the lowest Ram KL value. For Fig. 3, we always selected the run with the highest Ram KL value to illustrate that variations with fewer constraints are more likely to exhibit mode collapse.

RAMACHANDRAN KL DIVERGENCE (RAM KL AND RAM KL W.RW.)

To obtain quantitative estimates of the quality of such Ramachandran plots, we used two metrics, following the approach of [51, 64]. We computed the forward KL divergence between the Ramachandran plots from ground truth and model samples (RAM KL). For this, we used 100×100 bins and 1×10^7 samples. Additionally, we also calculated a reweighted version of the metric (Ram KL w.Rw.) where the model samples were first reweighted to the target distribution before generation of Ramachandran plots.

For the larger systems, where more than one Ramachandran plot exists, we reported the average Ramachandran KL.

C.5 Hyperparameters

Hyperparameters play a crucial role in the performance of all models. Common hyperparameters include the choice of optimizer, learning rate, batch size, gradient steps, and weight decay. Below, we provide a description of the hyperparameters for each method, emphasizing any method-specific choices.

All experiments employed the Adam optimizer [42]. Our implementation builds on the Python packages *bgflow* [56], *nflows* [32], and *PyTorch* [60]. The number of parameters in the normalizing flow architecture for each system is summarized in Tab. 5.

Table 5: Number of flow parameters for each system. The number of parameters is completely determined by a molecular system's size, as the architecture is the same across all systems.

	ALANINE DIPEPTIDE	ALANINE TETRAPEPTIDE	ALANINE HEXAPEPTIDE	ELIL
NUMBER OF PARAMETERS	7 421 512	9 452 376	12 124 616	13 727 952

CMT

We refer to Tab. 6 for the general and method-specific hyperparameters of CMT.

TA-BG

Tab. 7 summarizes the hyperparameters for the pre-training of TA-BG [64] using the reverse KL divergence.

After pre-training, the temperature is annealed with a geometrically decaying temperature sequence and the hyperparameters summarized in Tab. 8. The TA-BG experiments on alanine dipeptide and

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	batch size	1000	1000	2000	2000
	Learning rate	4×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}
	LR scheduler	cosine	cosine	cosine	cosine
	Gradient Descent Steps	400 000	400 000	800 000	1 600 000
	Weight-Decay	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	LR Linear Warmup Steps	1000	1000	1000	1000
	Max Grad Norm	100.0	100.0	100.0	100.0
Method-specific	Trust-region bound	0.3	0.3	0.3	0.3
	Entropy bound	0.8	1.8	1.4	0.7
	Buffer size	500 000	500 000	1 000 000	1 000 000
	Gradient Descent Steps Per Annealing Step	2000	2000	2000	2000

Table 6: Hyperparameter settings for CMT (general and method-specific) for all systems.

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	Target Temperature	1200 K	1200 K	1200 K	3000 K
	Batch size	256	256	512	512
	Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
	LR scheduler	cosine	cosine	cosine	cosine
	Gradient Descent Steps	100 000	100 000	250 000	250 000
	Weight-Decay	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	LR Linear Warmup Steps	1000	1000	1000	1000
	Max Grad Norm	100.0	100.0	100.0	100.0
	NO. Highest Energy Values Removed	10	10	20	20

Table 7: Hyperparameter settings for TA-BG pre-training for all systems.

alanine tetrapeptide used the geometric temperature annealing sequence

$$1200\,\mathrm{K} \to 1028.69\,\mathrm{K} \to 881.84\,\mathrm{K} \to 755.95\,\mathrm{K} \to 648.04\,\mathrm{K} \to 555.52\,\mathrm{K} \to 476.22\,\mathrm{K} \to 408.24\,\mathrm{K} \to 349.96\,\mathrm{K} \to 300.00\,\mathrm{K} \to 300.00\,\mathrm{K}.$$

Including an additional finetuning step per temperature, TA-BG employs the temperature sequence

$$1200\,\mathrm{K} \to 1028.69\,\mathrm{K} \to 1028.69\,\mathrm{K} \to 881.84\,\mathrm{K} \to 881.84\,\mathrm{K} \to 755.95\,\mathrm{K} \to 755.95\,\mathrm{K} \to 648.04\,\mathrm{K} \to 648.04\,\mathrm{K} \to 555.52\,\mathrm{K} \to 555.52\,\mathrm{K} \to 476.22\,\mathrm{K} \to 476.22\,\mathrm{K} \to 408.24\,\mathrm{K} \to 408.24\,\mathrm{K} \to 349.96\,\mathrm{K} \to 349.96\,\mathrm{K} \to 300.00\,\mathrm{K} \to 300.00\,\mathrm{K}$$

on alanine hexapeptide. On ELIL, reverse KL pre-training suffers from mode-collapse at $1200\,\mathrm{K}$. Therefore, the temperature annealing starts at $3000\,\mathrm{K}$, resulting in the temperature sequence

$$3000.00\,\mathrm{K} \to 2573.09\,\mathrm{K} \to 2573.09\,\mathrm{K} \to 2573.09\,\mathrm{K} \to 2206.93\,\mathrm{K} \to 2206.93\,\mathrm{K} \to 2206.93\,\mathrm{K} \to 1892.88\,\mathrm{K} \to 1892.88\,\mathrm{K} \to 1892.88\,\mathrm{K} \to 1623.52\,\mathrm{K} \to 1623.52\,\mathrm{K} \to 1623.52\,\mathrm{K} \to 1392.49\,\mathrm{K} \to 1392.49\,\mathrm{K} \to 1392.49\,\mathrm{K} \to 1194.33\,\mathrm{K} \to 1194.33\,\mathrm{K} \to 1194.33\,\mathrm{K} \to 1024.37\,\mathrm{K} \to 1024.37\,\mathrm{K} \to 1024.37\,\mathrm{K} \to 878.60\,\mathrm{K} \to 878.60\,\mathrm{K} \to 878.60\,\mathrm{K} \to 753.57\,\mathrm{K} \to 753.57\,\mathrm{K} \to 753.57\,\mathrm{K} \to 646.34\,\mathrm{K} \to 646.34\,\mathrm{K} \to 646.34\,\mathrm{K} \to 554.36\,\mathrm{K} \to 554.36\,\mathrm{K} \to 554.36\,\mathrm{K} \to 475.48\,\mathrm{K} \to 475.48\,\mathrm{K} \to 475.48\,\mathrm{K} \to 407.81\,\mathrm{K} \to 407.81\,\mathrm{K} \to 407.81\,\mathrm{K} \to 349.78\,\mathrm{K} \to 349.78\,\mathrm{K} \to 349.78\,\mathrm{K} \to 300.00\,\mathrm{K} \to 300.00\,\mathrm{K}.$$

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	Batch size	2048	4096	2048	2048
	Learning rate	5×10^{-6}	1×10^{-5}	5×10^{-6}	5×10^{-6}
	LR scheduler	cosine (per temperature step)	-	-	-
	Gradient Descent Steps	300 000	200 000	360 000	600 000
Method-specific	Buffer size	7 440 000	7 440 000	15 111 111	22 400 000
	Buffer Resampled To	2 000 000	2 000 000	2 000 000	10 000 000
	Gradient Descent Steps per annealing step	30 000	20 000	20 000	20 000

Table 8: Hyperparameter settings for TA-BG (general and method-specific) for all systems.

FAB

The used hyperparameters for FAB [51] can be found in Tab. 9. Furthermore, we used a step size of 0.05 for the Hamiltonian Monte Carlo [30] transitions. For details on the method and its hyperparameters, we refer to [51].

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	Batch size	1024	1024	1024	2048
	Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	2×10^{-4}
	LR scheduler	cosine	cosine	cosine	cosine
	Gradient Descent Steps	50 000	50 000	50 000	25 000
	Weight-Decay	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	LR Linear Warmup Steps	1000	1000	1000	1000
	Max Grad Norm	1000.0	1000.0	1000.0	1000.0
Method-specific	No. Intermed. Dist.	8	8	8	16
	No. Inner HMC Steps	4	4	8	8

Table 9: Hyperparameter settings of FAB (general and method-specific) for all systems.

Forward and Reverse KL

This section reports the used hyperparameters for training with the forward KL divergence on MD data (Tab. 10) and the hyperparameters for training with the reverse KL divergence (Tab. 11). A description on how the MD data was obtained can be found in App. C.2.

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	Batch size	1024	1024	1024	1024
	Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}
	LR scheduler	cosine	cosine	cosine	cosine
	Gradient Descent Steps	100 000	100 000	120 000	140 000

Table 10: Hyperparameter settings of forward KL training using MD data for all systems.

	Hyperparameters	Alanine Dipeptide	Alanine Tetrapeptide	Alanine Hexapeptide	ELIL
General	Batch size	1024	1024	1024	1024
	Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
	LR scheduler	cosine	cosine	cosine	cosine
	Gradient Descent Steps	250 000	250 000	250 000	250 000
	Weight-Decay	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	LR Linear Warmup Steps	1000	1000	1000	1000
	Max Grad Norm	100.0	100.0	100.0	100.0
	NO. Highest Energy Values Removed	40	40	40	40

Table 11: Hyperparameter settings of reverse KL training for all systems.