SCALABLE INFERENCE-TIME ANNEALING FOR CONTINUOUS NORMALIZING FLOWS

Daniel Peñaherrera
CMU-Pitt Computational Biology
Dept. of Computational & Systems Biology
University of Pittsburgh
Pittsburgh, PA 15260
dap181@pitt.edu

Rishal Aggarwal
CMU-Pitt Computational Biology
Dept. of Computational & Systems Biology
University of Pittsburgh
Pittsburgh, PA 15260
rishal.aggarwal@pitt.edu

David Ryan Koes
Dept. of Computational & Systems Biology
University of Pittsburgh
Pittsburgh, PA 15260
dkoes@pitt.edu

ABSTRACT

A long standing challenge in computational chemistry and biophysics is efficiently sampling the Boltzmann distribution of molecules. Advances in generative modeling have been proposed to address the limitations of conventional sampling techniques by eliminating the computational cost of simulation. A promising direction is iteratively finetuning diffusion models along a temperature ladder whereby training data is generated via importance sampling during inference-time annealing. Unfortunately, these methods require computing a divergence over the score field to estimate importance weights, rendering them intractable for larger systems. Here we present scalable inference-time annealing (SITA), which retrains flow-based models to generate samples at progressively lower temperatures using an energy-based model to facilitate fast surrogate likelihoods. We demonstrate start-of-the-art performance on both alanine dipeptide and alanine tripeptide while avoiding costly divergence terms and using nearly 2 orders of magnitude fewer energy evaluations.

1 INTRODUCTION

Sampling the equilibrium ensemble of molecular configurations is a foundational task in statistical physics (Abramson et al., 2024; Zheng & Wang, 2025; Lin et al., 2022), as it provides access to thermodynamic observables like free energies and binding affinities, yet remains notoriously difficult for all but the simplest systems. Molecular ensembles are defined by the Boltzmann distribution whose probability density function is defined by $\pi(x) = Z^{-1} \exp(-\frac{E(x)}{k_B T})$ where $E(\cdot)$ is the energy function of the system x, k_B is the Boltzmann constant, T is the temperature, and Z is the normalizing constant known as the partition function. The difficulty in sampling the Boltzmann distribution arises from the highly non-convex and rugged nature of the energy function.

An emerging class of samplers are deep learning generative models that hold the promise for fast, amortized sampling of the Boltzmann distribution (Jing et al., 2024; Zheng et al., 2024; Schreiner et al., 2023; Wang et al., 2024; Lewis et al., 2025). However, the current paradigm of training these models is circular: it requires equilibrium ensembles for training data, yet generating such ensembles is precisely the intractable problem at hand. Recently, a new paradigm has emerged: train a generative model on high-temperature simulation data, then iteratively refit it on its own samples generated via an annealing mechanism that progressively lowers the temperature. Two benefits follow: (i) high-temperature simulation enhances exploration and mode coverage, and (ii) recursive bootstrapping enables data-efficient recovery of the target distribution.

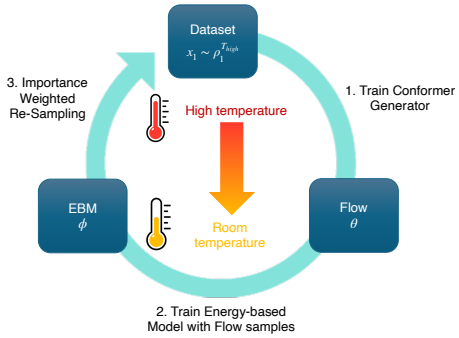

Figure 1: The SITA training loop. A flow model θ trained on high-temperature samples is used to generate proposals for training an energy-based model ϕ. Importance-weighted resampling with the learned surrogate likelihoods produces samples at lower temperatures, which seed the next annealing step and enables large temperature jumps without expensive Jacobian computations.

One such annealing method is PITA (Akhound-Sadegh et al., 2025), which combines diffusion models with inference-time modifications to the reverse-time generative process, enabling generation of samples that are approximately distributed by the lower temperature Boltzmann distribution. To improve training, PITA relies on self-normalized importance sampling (SNIS) at each annealing step based on the Feynman-Kac formula. This SNIS estimator, however, requires evaluating the divergence of the score field along the full integration path of the reverse process. Such computational overhead poses a serious limitation for systems with many degrees of freedom.

Main Contributions. In this work, we present SITA (Figure 1), a scalable inference-time annealing for continuous flows that circumvents path integral-based SNIS estimators via reliance on a new class of surrogate likelihood estimators called *BoltzNCE*. On the benchmark systems of alanine dipeptide and alanine tripeptide, we demonstrate the following:

- **Annealing for flows.** We develop an simple inference-time annealing procedure for continuous flow models that allows large temperature jumps across a pre-defined temperature ladder.
- **Fast surrogate likelihoods.** We show that learned likelihoods can replace expensive Jacobian computations in the reweighting step, recovering Boltzmann statistics with high accuracy.
- **Empirical validation.** SITA achieves state-of-the-art results across several metrics while requiring significantly fewer energy function evaluations than PITA.

2 Background

2.1 Stochastic Interpolants and Flow Matching

A stochastic interpolant (Albergo et al., 2023; Ma et al., 2024) defines a continuous-time process that transports samples from a base distribution ρ_0 to a target distribution $\rho_1 = \pi$. For paired samples $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$, the interpolant takes the form

$$I_t = \alpha_t x_0 + \beta_t x_1, \tag{1}$$

where $\alpha, \beta : [0, 1] \to \mathbb{R}$ are continuously differentiable scalar functions satisfying $\alpha_0 = 1$, $\alpha_1 = 0$, $\beta_0 = 0$, and $\beta_1 = 1$. This formulation encompasses diffusion models (Song et al., 2020; Ho et al., 2020), flow matching (Lipman et al., 2022), and rectified flows (Liu et al., 2022) as special cases under appropriate choices of α and β.

The time-marginal density $\rho_t = \text{Law}(I_t)$ coincides with the density evolved by a probability flow transporting mass from ρ_0 to ρ_1:

$$\dot{X}_t = v_t(X_t), \qquad v_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x], \tag{2}$$

where the velocity field v_t is characterized as the conditional expectation of the interpolant's time derivative.

Given a coupling $\rho(x_0, x_1)$ between the base and target distributions, the velocity field can be learned by training a neural network $\hat{v}$ via the regression loss

$$\mathcal{L}_v(\hat{v}) = \mathbb{E}\left[\|\hat{v}_t(I_t) - \dot{I}_t\|^2\right], \tag{3}$$

where the expectation is taken over (t, x_0, x_1). Samples are then generated by integrating the learned dynamics

$$\dot{x}_t = \hat{v}_t(x_t), \quad x_0 \sim \rho_0. \tag{4}$$

The density of the generated samples is given by the change-of-variables formula

$$\hat{\rho}_1(\hat{x}_1) = \rho_0(x_0) \exp\left(-\int_0^1 \nabla \cdot v_t(x_t)\, dt\right). \tag{5}$$

2.2 Surrogate Likelihood Estimators (BoltzNCE)

SITA employs a surrogate likelihood estimator parameterized as an energy-based model (EBM) to enable efficient reweighting of samples produced by the flow. Training EBMs was historically considered intractable, but recent advances, particularly BoltzNCE (Aggarwal et al., 2025), have made it practical by combining score matching with noise contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010; Oord et al., 2018).

BoltzNCE training proceeds in two stages. First, samples $\hat{x}_1 \sim \hat{\rho}_1$ are drawn from the flow to serve as training data for the EBM. Second, the EBM is fitted to these samples to learn an energy function U over the flow's output distribution $\hat{\rho}_1$.

The training objective is derived by introducing an auxiliary stochastic interpolant from a Gaussian base to $\hat{\rho}_1$. The score matching component takes the form

$$\mathcal{L}_{\text{SM}}(\hat{U}) = \mathbb{E}\left[|\alpha_t \nabla \hat{U}_t(\tilde{I}_t) + x_0|^2\right], \tag{6}$$

where the expectation is over $(t, x_0, \hat{x}_1)$ with $\tilde{I}_t = \alpha_t x_0 + \beta_t \hat{x}_1$. While score matching constrains the gradient of the energy, an additional NCE-based term is needed to anchor the energy values themselves. This is achieved by training the model to discriminate between samples from different time points:

$$\mathcal{L}_{\text{InfoNCE}}(\hat{U}) = -\mathbb{E}\left[\log\left(\frac{\exp(\hat{U}_t(\tilde{I}_t))}{\exp(\hat{U}_{t'}(\tilde{I}_t)) + \exp(\hat{U}_t(\tilde{I}_t))}\right)\right], \tag{7}$$

where t' denotes a contrastive time point and the expectation is over $(t', t, x_0, \hat{x}_1)$. The full BoltzNCE objective combines both terms:

$$\mathcal{L}_{\text{BoltzNCE}}(\hat{U}) = \mathcal{L}_{\text{SM}}(\hat{U}) + \mathcal{L}_{\text{InfoNCE}}(\hat{U}). \tag{8}$$

3 Scalable Inference-Time Annealing

In this section, we outline how through a combination of surrogate likelihood estimation and flow annealing we facilitate a highly scalable algorithm to model the Boltzmann distribution of physical systems. Let $\{T_k\}_{k=0}^K$ denote a decreasing sequence of temperatures and $E(\cdot)$ be the energy function associated with the target Boltzmann distribution $\pi(x_1)$, which we can evaluate exactly. Given a flow with parameters θ pre-trained on high-temperature simulation data and an EBM with parameters ϕ pre-trained on flow generated outputs, SITA's multi-phase annealing bootstrap proceeds as follows:

(i) **Anneal the flow.** Samples $\hat{x}_1 \sim \hat{\rho}_1^{T_{k+1}}$ are generated from the flow by drawing $x_0 \sim \mathcal{N}(0, \frac{T_{k+1}}{T_k}\mathbf{I})$ and integrating the generative ODE.

(ii) **Finetune the EBM.** Using flow generated samples, the EBM is finetuned to better approximate the likelihoods of $\hat{x}_1 \sim \hat{\rho}_1^{T_{k+1}}$.

(iii) **Importance sampling.** We use the newly fitted EBM to compute importance weights $\tilde{w}(\hat{x}_1) = \exp(-\frac{1}{K_B T_{k+1}} E(\hat{x}_1) - \hat{U}_\phi(\hat{x}_1))$ for each sample generated by the flow. Subsequently, samples are re-weighted to provide a new dataset at temperature T_{k+1} to retrain the flow.

(iv) **Finetune the flow.** Using the set of importance weighted samples, the flow is finetuned to better approximate the annealed target distribution. Upon completion, the whole process begins again until the final temperature T_K is reached.

We emphasize that the same two models are maintained throughout the entire annealing bootstrap process. Their respective optimizers are simply re-initialize at the start of each new finetuning step. Pseudo-code is provided in the Appendix (Algorithm 1).

3.1 Temperature Steering of the Probability Flow ODE

We observe that velocity field models can induce temperature changes in the generated distribution without explicit temperature conditioning during training. A velocity field $\hat{v}$ trained on simulation data at temperature T_{high} transports samples from the base distribution $\rho_0(x_0) = \mathcal{N}(0, \mathbf{I})$ to produce $\hat{x}_1 \sim \hat{\rho}_1^{T_{\text{high}}}$, where

$$\hat{\rho}_1^{T_{\text{high}}}(\hat{x}_1) = \rho_0(x_0) \exp\left(- \int_0^1 \nabla \cdot \hat{v}_t(x_t)\, dt \right). \tag{9}$$

Crucially, the flow learns to map ρ_0 to an approximation of the high-temperature target without explicitly modeling the temperature dependence, this information is encoded implicitly in the flow's output distribution.

This implicit encoding enables a simple mechanism for temperature transfer. By raising the flow density to the power $\kappa = T_{\text{high}}/T_{\text{low}}$, we obtain the proportionality relation

$$\hat{\rho}_1^{T_{\text{low}}}(\hat{x}_1) \propto \left[\hat{\rho}_1^{T_{\text{high}}}(\hat{x}_1) \right]^\kappa = \left[\rho_0(x_0) \exp\left(- \int_0^1 \nabla \cdot \hat{v}_t(x_t)\, dt \right) \right]^\kappa. \tag{10}$$

Consequently, sampling from the low-temperature distribution $\hat{\rho}_1^{T_{\text{low}}}$ only requires a rescaling of the base distribution variance by κ^{-1} at inference time:

$$[\rho_0(x)]^\kappa = \exp\left(-\frac{T_{\text{high}} \|x\|^2}{2T_{\text{low}}} \right) \propto \mathcal{N}(0, \kappa^{-1}\mathbf{I}). \tag{11}$$

This rescaling allows SITA to achieve large temperature jumps without modifying the velocity field model's architecture or enforcing volume preservation under the instantaneous change of variables. Further details on temperature steerability are provided in Appendix A.

4 Experiments

We evaluate SITA against PITA on two molecular benchmark systems: alanine dipeptide and alanine tripeptide. Additionally, we include two separate baselines from Akhound-Sadegh et al. (2025) namely a diffusion model (MD-Diff) and normalizing flow (MD-NF) trained on MD data collected at the target temperature of 300 Kelvin. For our experiment, we pre-train SITA's flow model on the same 1200K MD simulation data as PITA. All evaluation metrics we calculate on the same test-split used in PITA, containing 10,000 randomized frames from a 300K MD simulation. All metrics were evaluated across 3 seeds by generating 10,000 samples from SITA's flow post-bootstrap. To quantify mode coverage and precision, we compute Wasserstein distances over Ramachandran coordinates ($\mathbb{T}$-$\mathbb{W}_2$) and energy distributions ($\mathcal{E}$-$\mathbb{W}_1$, $\mathcal{E}$-$\mathbb{W}_2$). Additionally, we measure KL divergence between ground-truth and generated Ramachandran distributions (RAM-KL).

Architecture. SITA's flow model is a Geometric Vector Perceptron graph neural network, introduced by Jing et al. (2021b), equipped with $E(3)$-invariance in its scalar features and E(3)-equivariance in its vector features. To parameterize the EBM, we use the Graphormer architecture Ying et al. (2021) due to its successful application in Aggarwal et al. (2025). Both flow and score matching objectives used for training make use of trigonometric interpolants introduced in Albergo et al. (2023), where $\alpha_t = \cos(\frac{\pi}{2}t)$ and $\beta_t = \sin(\frac{\pi}{2}t)$.

4.1 Main results

Alanine Dipeptide (ADP). SITA is applied to the task of sampling conformations of alanine dipeptide at the target temperature of 300K, having been initially pre-trained on an MD dataset simulated at 1200K. We define an annealing schedule over temperatures 755.95K, 555.52K, 408.24K, 300.00K. We note this includes one extra step of 408.24K compared to PITA applied to alanine dipeptide. Yet, despite this we report substantially fewer energy evaluations.

Table 1: ESS across temperatures for ADP.

	755.95K	555.52K	408.24K	300.00K
ESS ↑	0.131	0.218	0.192	0.268

At each annealing step, SITA's annealed flow generates 200,000 samples to fine-tune the EBM. A new set of 100,000 samples is then produced by importance weighted re-sampling of the original 200,000 sample using a the EBM to calculate the SNIS estimate. We report the effective sample size (ESS) for each annealing step in Table 3. SITA achieves the best performance on Rama-KL and the energy Wasserstein-2 metric while using roughly $60\times$ fewer energy evaluations, as can be seen from Table **??**. While SITA outperforms PITA on the energy Wasserstein-1 metric, both fail to outperform the MD-NF baseline trained directed on equilibrium samples generate by MD simulation at 300K. Visual comparison confirms that SITA captures all major conformational basins and matches the reference energy distribution (Figure 2). SITA remains competitive with PITA on the torsion metric ($\mathbb{T}$-$\mathcal{W}_2$), where accurately recovering mode populations remains challenging.

It should be noted that in terms of wall-clock time for training, PITA takes roughly 9.4 hours to complete its full training on alanine dipeptide. SITA, in contrasts, requires 44.7 hours for the bootstrap alone. This can largely be attributed to SITA's alternating training scheme, differences in model architectures, and differences in GPU hardware performance. Lastly, inference time for generating 30,000 samples takes 4.7 minutes for PITA and 8.7 minutes for SITA.

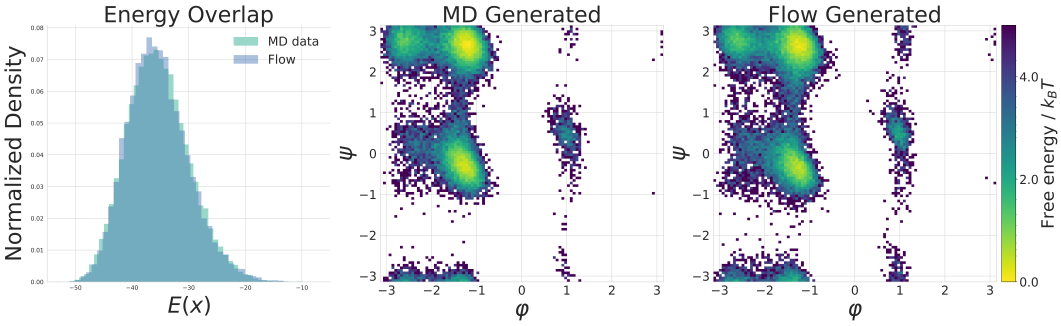

Figure 2: Alanine dipeptide comparison on 30,000 samples from both SITA and MD simulation. Flow-generated samples closely match the MD reference in both energy distribution and Ramachandran free energy landscape, capturing all major conformational basins.

	Rama-KL ↓	Energy-$\mathcal{W}_1$ ↓	Energy-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓	#Energy Evals ↓
SITA	$\mathbf{0.517 \pm 0.013}$	0.865 ± 0.080	$\mathbf{0.939 \pm 0.079}$	0.326 ± 0.004	$\mathbf{8 \times 10^5}$
PITA	4.773 ± 0.460	1.530 ± 0.068	1.615 ± 0.053	$\mathbf{0.270 \pm 0.023}$	5×10^7
MD-Diff	1.308 ± 0.072	3.627 ± 0.023	3.704 ± 0.026	0.310 ± 0.001	5×10^7
MD-NF	13.533 ± 0.024	$\mathbf{0.551 \pm 0.062}$	1.198 ± 0.069	0.403 ± 0.045	5×10^7

Table 2: Alanine Dipeptide at 300K. All metrics calculated over 10,000 samples across 3 seeds.

Table 3: ESS across temperatures for ATP.

	755.95K	555.52K	408.24K	300.00K
ESS ↑	0.045	0.067	0.074	0.065

Alanine Tripeptide (ATP). We further evaluate SITA on the larger alanine tripeptide system, using the same annealing schedule and bootstrapping procedure as in the alanine dipeptide experiment. Note that in this application, the annealing schedules between SITA and PITA are now the same. Here, we observe that SITA outperforms all methods on nearly all metrics (Table 4) with the exception of the torsion metric ($\mathbb{T}$-$\mathcal{W}_2$). Remarkably, SITA is able to achieve this superior performance without requiring any relaxation of generated samples. In contrast, PITA must run a short MD refinement of its generated samples at the target temperature to remain competitive. Runtime for the SITA's annealing bootstrap on alanine tripeptide does however exceed that of the alanine dipeptide bootstrap, clocking in at a total of 3 days 22 hours and 22 minutes.

	Rama-KL	Energy-$\mathcal{W}_1$ ↓	Energy-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$	#Energy Evals
SITA	$\mathbf{0.361 \pm 0.025}$	$\mathbf{1.933 \pm 0.298}$	$\mathbf{2.054 \pm 0.268}$	0.798 ± 0.268	$\mathbf{8 \times 10^5}$
PITA	1.209 ± 0.144	2.567 ± 0.108	2.592 ± 0.107	0.521 ± 0.006	8×10^7
PITA (w/o relaxation)	8.535 ± 0.254	86.270 ± 0.294	87.695 ± 0.294	0.651 ± 0.013	5×10^7
MD-Diff	9.662 ± 0.085	7.416 ± 0.130	7.599 ± 0.137	$\mathbf{0.424 \pm 0.011}$	8×10^7

Table 4: Alanine Tripeptide at 300K. All metrics calculated over 10,000 samples across 3 seeds.

4.2 TICA Evaluations

In this section we discuss the use of time-independent component analysis (TICA) Pérez-Hernández et al. (2013); Schwantes & Pande (2013) for assessing generative model performance. In the application of generative modeling to MD data, it is common to quantify the discrepancy between TICA projections of model generated samples and MD simulation generated samples. In fact, the original PITA paper reports Wasserstein-1 and Wasserstein-2 metrics on such TICA projections. However based on PITA's publicly available code, we found that these metrics relied on projecting the first 10,000 frames of the MD trajectory of each molecular system studied in order to match the number of samples generated by the diffusion model at test-time. We caution against this method of down-sampling due to the fact that MD simulation data do not represent I.I.D. samples but are in fact a trajectory with time-dependent correlations. This means that simply selecting the first few frames of a trajectory will result in the potential dropping of modes and / or failing to capture the correct relative weights of the modes. Instead, we advocate for simply randomly sampling uniformly along frames of the MD trajectory.

In Table 5, we report TICA-$\mathcal{W}_1$ and TICA-$\mathcal{W}_2$ metrics using both down-sampling techniques and find significant improvement when uniformly sampling frames. Due to the non-availability of the original MD trajectories used by PITA, we generated our own trajectories for both alanine dipeptide and alanine tripeptide using the same simulation configuration outline in PITA. The qualitative agreement between the MD-generated and SITA-generated TICA projections is further illustrated in Figure 3, which shows density scatter plots of the first two TICA components for both molecular systems.

	ADP TICA				ATP TICA			
	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1^*$	$\mathcal{W}_2^*$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1^*$	$\mathcal{W}_2^*$
SITA	0.155 ± 0.009	0.629 ± 0.212	0.089 ± 0.012	0.414 ± 0.035	0.184 ± 0.007	0.723 ± 0.012	0.073 ± 0.009	0.315 ± 0.039
PITA	—	—	—	—	0.272 ± 0.017	0.952 ± 0.055	—	—
PITA (w/o relax.)	0.118 ± 0.006	0.379 ± 0.028	—	—	0.405 ± 0.014	0.999 ± 0.043	—	—
MD-Diff	0.113 ± 0.001	0.579 ± 0.004	—	—	0.059 ± 0.006	0.426 ± 0.010	—	—
MD-NF	0.138 ± 0.003	0.586 ± 0.003	—	—	—	—	—	—

Table 5: TICA-$\mathcal{W}_1$ and TICA-$\mathcal{W}_2$ metrics for alanine dipeptide (ADP) and alanine tripeptide (ATP). Columns marked with * denote metrics computed using uniform frame sampling along the MD trajectory, while unmarked columns use the first-frame down-sampling method from the original PITA paper. All metrics are calculated over 10,000 samples across 3 seeds.

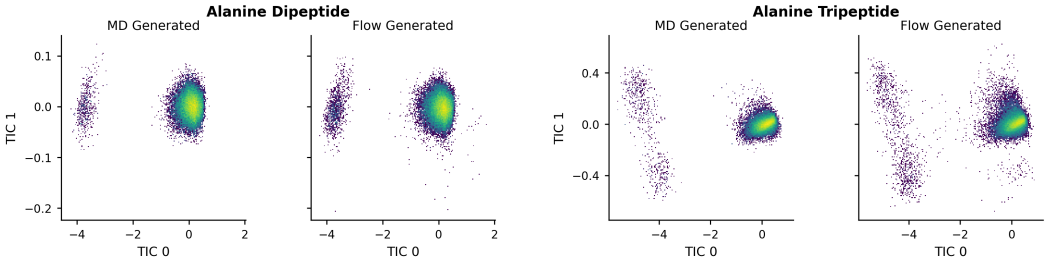

Figure 3: TICA projection density scatter plots comparing MD-generated and SITA flow-generated samples for alanine dipeptide (left) and alanine tripeptide (right). Each panel plots TICA projections of 30,000 samples, with color intensity reflecting local sample density. The close correspondence between MD and flow-generated distributions in both systems provides qualitative confirmation of the quantitative improvements reported in Table 5.

5 Conclusion

We introduce SITA, a scalable framework that trains continuous flow models to recover target Boltzmann distributions via temperature annealing. SITA achieves computational efficiency by circumventing vector field divergence computations and minimizing energy function evaluations. Our method sets a new state-of-the-art on both alanine dipeptide and alanine tripeptide systems. More crucially, SITA provides a tractable route to modeling molecular ensembles with many degrees of freedom—a regime where existing methods struggle. Future directions include transferable architectures and application to larger molecular systems. All code is available at `https://github.com/countrsignal/sita/`.

References

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

Rishal Aggarwal, Jacky Chen, Nicholas M Boffi, and David Ryan Koes. BoltzNCE: Learning likelihoods for boltzmann generation with stochastic interpolants and noise contrastive estimation. *arXiv preprint arXiv:2507.00846*, 2025.

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

Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and boltzmann generators. *Phys. Rev. Res.*, 4:L042005, Oct 2022. doi: 10.1103/PhysRevResearch.4.L042005. URL `https://link.aps.org/doi/10.1103/PhysRevResearch.4.L042005`.

Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3d de novo molecule generation. *arXiv:2404.19739 [q-bio.BM]*, 2024. URL `https://arxiv.org/abs/2404.19739`.

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`.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

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

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021a.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons, 2021b. URL https://arxiv.org/abs/2009.01411.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

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

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of Chemical Physics*, 139(1), July 2013. ISSN 1089-7690. doi: 10.1063/1.4811489. URL http://dx.doi.org/10.1063/1.4811489.

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.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. *Advances in Neural Information Processing Systems*, 36:36449–36462, 2023.

Christian R. Schwantes and Vijay S. Pande. Improvements in markov state model construction reveal many non-native interactions in the folding of ntl9. *Journal of Chemical Theory and Computation*, 9(4):2000–2009, 2013. doi: 10.1021/ct300878a. URL https://doi.org/10.1021/ct300878a. PMID: 23750122.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yan Wang, Lihao Wang, Yuning Shen, Yiqun Wang, Huizhuo Yuan, Yue Wu, and Quanquan Gu. Protein conformation generation via force-guided se (3) diffusion models. *arXiv preprint arXiv:2403.14088*, 2024.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph, 2023. URL https://arxiv.org/abs/2105.02605.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Haiyang Zheng and Jin Wang. Alphafold3 in drug discovery: A comprehensive assessment of capabilities, limitations, and applications. *bioRxiv*, pp. 2025–04, 2025.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, et al. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567, 2024.

A Temperature Steerability in Flows

A.1 Normalizing Flows

In their original formulation, normalizing flows learn a differentiable bijection $x = \phi_\theta(z)$ that maps samples from a simple base distribution $\rho_Z(z)$, typically a multivariate Gaussian, to some target distribution $\pi(x)$. By the change-of-variables formula, likelihoods under the normalizing flow can be evaluated as

$$\rho_\theta(x) = \rho_Z(\phi_\theta^{-1}(x)) \left| \frac{\partial \phi_\theta^{-1}}{\partial x}(x) \right|, \tag{12}$$

where $z = \phi_\theta^{-1}(x)$ and $\left|\partial \phi_\theta^{-1}/\partial x\right|$ is the Jacobian determinant of the flow map ϕ_θ, which is comprised of a discrete sequence of invertible neural network layers with parameters θ. In the continuous time perspective, continuous normalizing flows (CNF) replaces this stack of maps with a time-indexed family of maps $X_t(x) = \phi_t(x_0)$ with $t \in [0, 1]$ that acts as the pushforward of the base distribution ρ_0 at time $t = 0$ to some time-varying density ρ_t that terminates at ρ_1 at time $t = 1$. Conveniently, the flow map ϕ_t is characterized by the ordinary differential equation (ODE)

$$\dot{X}_t(x) = \frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)), \qquad X_{t=0}(x) = x \tag{13}$$

where v_t is the *velocity field* governing the transport of individual samples. From the density perspective, the pushforward of ρ_0 under ϕ_t yields a time-varying density ρ_t that satisfies the continuity equation

$$\partial_t \rho_t + \nabla \cdot (v_t \rho_t) = 0 \qquad \text{with boundaries } \rho_{t=0} = \rho_0 \text{ and } \rho_{t=1} = \rho_1. \tag{14}$$

Denoting $\hat{\rho}_1$ as the CNF's output distribution, likelihoods for generated samples $\hat{x}_1$ are given by the instantaneous change-of-variables formula,

$$\hat{\rho}_1(\hat{x}_1) = \rho_0(x_0) \exp\left(- \int_0^1 \nabla \cdot v_t(x_t) dt \right). \tag{15}$$

Generative modeling thus reduces to estimating a velocity field v_t such that (14) is satisfied.

A.2 Explicit Temperature Steering

As shown in Dibak et al. (2022), normalizing flows can be extended to model thermodynamic ensembles across a range of temperatures. When trained on high-temperature simulation data, a flow can generate samples at lower temperatures by *explicitly* accounting for temperature as a parameter of both the flow map and base distribution. Concretely, a change of temperature from $T_{\text{high}} \to T_{\text{low}}$ induces the following scaling relation in the output density:

$$\rho_Z^{T_{\text{low}}}(z) \left| \frac{\partial \phi_{T_{\text{low}}}^{-1}}{\partial x}(x) \right| \propto \left[\rho_Z^{T_{\text{high}}}(z) \left| \frac{\partial \phi_{T_{\text{high}}}^{-1}}{\partial x}(x) \right| \right]^{\kappa} \tag{16}$$

where $\kappa = T_{\text{high}}/T_{\text{low}}$. A flow satisfying this relation is said to be *temperature steerable*. Under volume preservation, the relation decouples: proportionality in the base distribution implies proportionality in the Jacobian determinants

$$\rho_Z^{T_{\text{low}}}(z) \propto \left[\rho_Z^{T_{\text{high}}}(z) \right]^{\kappa} \implies \left| \frac{\partial \phi_{T_{\text{low}}}^{-1}}{\partial x}(x) \right| \propto \left[\left| \frac{\partial \phi_{T_{\text{high}}}^{-1}}{\partial x}(x) \right| \right]^{\kappa}, \tag{17}$$

a condition satisfied by adopting a Gaussian base with temperature-scaled variance, $z \sim \mathcal{N}(0, T_{\text{low}}\mathbf{I})$. SITA *implicitly* accounts for temperature through these proportionality relations under the instantaneous change of variables.

B Architecture Details

B.1 Input Representation

We encode molecular conformers as fully connected graphs whose node attributes reflect the atom types specified by alanine dipeptide's topology. Both the flow and EBM operate directly in Cartesian space, taking raw atomic coordinates as input.

B.2 Flow Architecture: Geometric Vector Perceptrons

The flow component of SITA is built on an E(3)-equivariant graph neural network employing geometric vector perceptrons (Satorras et al., 2021) on molecular generation tasks (Dunn & Koes, 2024).

Following Dunn & Koes (2024), we incorporate architectural modifications shown to improve performance. Message passing follows the formulation of Jing et al. (2021a):

$$(m_{i\to j}^{(s)},\, m_{i\to j}^{(v)}) = \psi_M\Big([h_i^{(l)} : d_{ij}^{(l)}],\; v_i : \left[\frac{x_i^{(l)} - x_j^{(l)}}{d_{ij}^{(l)}}\right]\Big) \tag{18}$$

where $m_{i\to j}^{(v)}$ and $m_{i\to j}^{(s)}$ denote scalar and vector messages from node i to j, h_i collects scalar node feature, d_{ij} encodes pairwise distances via a radial basis, and x gives atomic positions. Full details on node and position updates appear in Appendix C of Dunn & Koes (2024).

B.3 EBM Architecture: Graphormer

For SITA's energy-based model, we employ the graphormer (Ying et al., 2021), a transformer variant that has achieved strong results on molecular property prediction. The key departure from standard transformers lies in a structure-aware attention bias derived from graph topology. In the 3D setting, this bias emerges from the pairwise Euclidean distance matrix processed through an MLP.

At a high level, graphormers interleave GNN-style operations within transformer blocks (Yang et al., 2023). Each attention head computes:

$$\text{head} = \text{softmax}\Big(\frac{QK^{\mathsf{T}}}{\sqrt{d}} + B\Big)\, V \tag{19}$$

In the original formulation, B is a learned bias matrix. Our implementation instead computes B by passing the Euclidean distance matrix passed through an MLP, directly injecting geometric information into the attention mechanism.

C Training Setup

C.1 Computational Resources

All training experiments were ran on a single NVIDIA L40 GPU with 46 GB GDDR6 memory.

C.2 Pre-training

For both the flow and EBM, each were pre-trained for a total of 500 epochs with a batch size 512. Each model was trained using separate Adam optimizers with learning rates of $1e-3$ for the flow and $5e-4$ for the EBM. Both models utilized separate reduce-on-plateau learning rate schedulers that monitored for loss convergence at a patience of 30 epochs and a reduction factor of 0.5. An exponential moving average with a decay of 0.999 was used for each. The flow was training on 500,000 conformations of alanine dipeptide simulated at 300K. In turn, the EBM was trained one 200,000 alanine dipeptide conformers generated by the flow model.

C.3 Annealing Bootstrap

Training setup for the annealing bootstrap is nearly identical to that of the pre-training phase with the only exception that both model use a learning rate of $5e-4$.

D Metrics

Effective Sample Size Reported ESS is calculate by

$$\text{ESS}(\{w_i\}_i^N) = \frac{1}{\sum_i w_i^2} \tag{20}$$

where weights w_i are normalized.

Evaluation via optimal transport. We measure sample quality through the lens of optimal transport. Let $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$ denote samples from the generated and reference distributions, respectively. The 2-Wasserstein distance seeks the minimum-cost coupling π between these point clouds:

$$W_2(\mu,\nu) = \left(\min_{\pi \in \Pi(\mu,\nu)} \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij}\, c(x_i, y_j)^2 \right)^{1/2} \tag{21}$$

where $\Pi(\mu,\nu)$ contains all joint distributions with the correct marginals. Optimal couplings are obtained via the POT library (Flamary et al., 2021). Below we describe two cost functions tailored to alanine dipeptide.

Energetic Cost. Potential energy provides a global summary sensitive to both local bonded geometry and long-range interactions. We assess agreement in energy space by setting

$$c_E(x,y)^2 = \big(E(x) - E(y)\big)^2 \tag{22}$$

yielding the metric E-$\mathcal{W}_2$.

Torsional Cost. Ramachandran angles (ϕ, ψ) offer a compact descriptor of backbone geometry. For alanine dipeptide, the backbone geometry is fully characterized by two dihedral angles, ϕ and ψ. Since these angles live on a torus, we adopt a periodic cost:

$$c_{\mathcal{T}}(x,y)^2 = \Big[\big(\phi_x - \phi_y + \pi\big) \bmod 2\pi - \pi\Big]^2 + \Big[\big(\psi_x - \psi_y + \pi\big) \bmod 2\pi - \pi\Big]^2 \tag{23}$$

Ramachandran KL divergence. We discretize the (ϕ, ψ) torus into a uniform grid and estimate densities via 2D histograms over $[-\pi, \pi]^2$. The KL divergence between reference and generated distributions is then approximated as

$$D_{\mathrm{KL}}(p\|q) \approx \sum_{i,j} p_{ij} \log \frac{p_{ij} + \epsilon}{q_{ij} + \epsilon} \cdot \Delta^2 \tag{24}$$

where p_{ij} and q_{ij} are the normalized histogram densities in bin (i, j), ϵ is a small constant for numerical stability, and $\Delta = 2\pi / N_{\mathrm{bins}}$ is the bin width.

E PSEUDOCODE

Algorithm 1 Scalable Inference-Time Annealing (SITA)

Require: Temperature ladder $\{T_k\}_{k=0}^{K}$ (decreasing), energy function $E(\cdot)$
Require: Pre-trained flow $\hat{v}_\theta$, pre-trained EBM $\hat{U}_\phi$
Ensure: Flow model $\hat{v}_\theta$ at target temperature T_K

for $k = 0$ to $K - 1$ **do**
  // Anneal the flow
  Draw $x_0 \sim \mathcal{N}(0, \frac{T_{k+1}}{T_k}\mathbf{I})$
  Generate $\hat{x}_1 \sim \hat{\rho}_1^{T_{k+1}}$ by integrating generative ODE

  // Finetune the EBM
  Reinitialize optimizer for ϕ
  Update ϕ using $\mathcal{L}_{\text{BoltzNCE}}(\hat{U}_\phi)$ (see 8) for a total of N_{EBM} epochs

  // Importance-weighted resampling
  Compute log weights $\log \tilde{w}(\hat{x}_1) \leftarrow -\frac{1}{k_B T_{k+1}} E(\hat{x}_1) - \hat{U}_\phi(\hat{x}_1)$

  // Clip at 99th percentile
  $\log \tilde{w}(\hat{x}_1) \leftarrow \texttt{quantile\_clip}(\log \tilde{w}(\hat{x}_1), 0.99)$
  Resample $\{\hat{x}_1\}$ according to self-normalized weights $w(\hat{x}_1)$

  // Finetune the flow
  Reinitialize optimizer for θ
  Update θ on resampled data using $\mathcal{L}_v(\hat{v}_\theta)$ (see 3) for a total of N_{Flow} epochs
end for
return $\hat{v}_\theta$