AUTOREGRESSIVE BOLTZMANN GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient sampling of molecular systems at thermodynamic equilibrium is a hallmark challenge in statistical physics. This challenge has driven the development of Boltzmann Generators (BGs), which allow rapid generation of uncorrelated equilibrium samples by combining a generative model with exact likelihoods and an importance sampling correction. However, modern BGs predominantly rely on normalizing flows (NFs), which either suffer from limited expressivity due to strict invertibility constraints (discrete time) or computationally expensive likelihoods (continuous time). In this paper, we propose AUTOREGRESSIVE BOLTZMANN GENERATORS (ARBG), a novel autoregressive modelling framework that overcomes these limitations by departing from the flow-based BG paradigm. ARBG circumvents the topological constraints of flows and enables sequential inference-time interventions, while offering enhanced scalability by leveraging architectures effective in Large Language Models. We empirically demonstrate that ARBG leads to significant improvements over flow-based models across all benchmarks, but particularly in larger peptide systems such as the 10-residue Chignolin. Furthermore, we introduce ROBIN, a 132 million parameter transferable model trained with the ARBG framework which improves over the previous state-of-the-art, reducing the zero-shot energy error, $\mathcal{E}$-$\mathcal{W}_2$, on 8-residue systems by over 60%.

1 INTRODUCTION

A central insight of statistical mechanics is that macroscopic phenomena, such as protein-folding (Noé et al., 2009; Lindorff-Larsen et al., 2011), magnetization of an Ising model (Yang, 1952), and crystal structure formation (Parrinello & Rahman, 1980; Matsumoto et al., 2002) are governed by the ensemble of microscopic states at equilibrium. This equilibrium is known as the *Boltzmann distribution* $\mu_{\text{target}}(x) \propto \exp\left(-\mathcal{E}(x)/k_{\text{B}}T\right)$, where $\mathcal{E}(x)$ is the dimensionless potential energy of a conformation $x \in \mathbb{R}^{n\times 3}$, and k_BT is the thermal energy at temperature T. Accordingly, the computational challenge is to efficiently draw statistically independent samples from this distribution.

A key characteristic of this sampling problem is that states in thermodynamic equilibrium—i.e., modes of the distribution—are often sparse and well-separated by high-energy barriers (Wirnsberger et al., 2020; Rizzi et al., 2021). The dominant approach for exploring this landscape is Molecular Dynamics (MD) (Alder & Wainwright, 1959; Rahman, 1964), which seeks to simulate the equations of motion with finely-discretized time-steps. However, this approach suffers from a severe timescale problem. MD simulations often use timesteps on the order of femtoseconds (10^{-15}s), but mode mixing across these high-energy barriers requires time-resolutions of microseconds (10^{-6}s) to seconds (10^{0}s) (Olsson, 2026). Consequently, the vast majority of MD computation is spent simulating high-frequency vibrations within local minima rather than exploring the global energy landscape, rendering MD too computationally expensive for practical problems (Perez et al., 2025). While many accelerated MD schemes have been explored (Hénin et al., 2022; Syed et al., 2021; Klein et al., 2023a; Kapuśniak et al., 2026), they still have difficulty with this fundamental mixing problem.

Boltzmann Generators (BGs) Noé et al. (2019) have emerged as a powerful framework to circumvent this. BGs learn a generative model $p_\theta(x)$ to propose samples for importance sampling, leveraging the exact model likelihood $p_\theta(x)$ and the target energy $\mathcal{E}(x)$. This allows for the parallel generation of independent and consistent samples without having to traverse between modes. These features make BGs an attractive framework when performing equilibrium sampling of large molecular systems. However, to satisfy the requirement for tractable exact likelihoods, the field has relied almost exclusively on normalizing flows (NFs) in either discrete (Tabak & Vanden-Eijnden, 2010; Tabak & Turner, 2013; Dinh et al., 2017; Rezende & Mohamed, 2015) or continuous time (Chen et al., 2018).

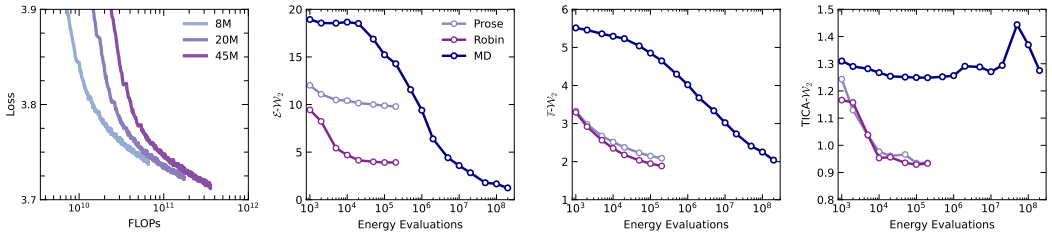

Figure 1: (Left) The loss vs. FLOPs for different model scales on the decapeptide: Chignolin. **(Right)** Depicts the inference-time scaling across three metrics comparing our ROBIN model, the previous state-of-the-art BG Prose, and a short molecular dynamics (MD) chain for the same number of energy function evaluations.

This reliance on flow-based architectures, however, imposes severe theoretical limitations. Fundamentally, NFs in practical instantiations are not only diffeomorphisms but also *homeomorphisms*, as the prior is a single Gaussian (Cornish et al., 2020; Dupont et al., 2019; Runde et al., 2005). Consequently, such generative models preserve the topology of their domain and struggle to model target distributions with distinct topologies to the prior, e.g., disjoint supports, differing number of connected components, or "holes". The equilibrium distribution of molecular conformations precisely exhibits these challenging topologies, as several metastable states are separated by regions of high-energy barriers. To morph the single connected mode of a Gaussian to these effectively separated states, a flow is forced to perform extreme deformations, stretching space across thin bridges and compressing it into modes. This leads to highly non-smooth mappings prone to exploding Lipschitz constants and ill-conditioned Jacobians, resulting in discrete NFs being notoriously unstable, greatly limiting model expressivity.

Continuous normalizing flows (CNFs) (Chen et al., 2018) are free from the same architectural constraints as discrete-time NFs. In addition, modern training strategies for flow-matching (Peluchetti, 2021; Liu, 2022; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023) can greatly stabilize the optimization process, yet the ill-conditioning manifests itself during inference. The learned vector field becomes highly non-smooth, resulting in *stiff* dynamics (Hochbruck & Ostermann, 2010; Hochbruck et al., 2020) leading to a large number of function evaluations at inference. This presents a currently unavoidable tradeoff: (**1**) Discrete-time NFs yield efficient likelihoods but have poor sample quality, while (**2**) CNFs are more expressive, but require expensive ODE solvers for likelihood evaluation, rendering importance sampling approaches computationally prohibitive.

Present work. In this work, we propose AUTOREGRESSIVE BOLTZMANN GENERATORS (ARBG), a novel alternative to flow-based BGs that circumvents these limitations. ARBG employs an autoregressive paradigm to factorize the molecular conformation density into a sequence of conditional densities: $p_\theta(x) = \prod_j p(x_j|x_{<j})$. This formulation offers three distinct advantages. (**1**) ARBG overcomes the expressivity bottlenecks of normalizing flows without incurring the computational cost of continuous flows. (**2**) ARBG avoids the numerical instability of learning high-distortion diffeomorphisms and allows it to model discontinuous jumps and separate modes naturally present in complex multi-modal target densities. (**3**) ARBG benefits from the massive investment and advances in discrete generative modelling that power modern Large Language Models (LLMs), and exhibits similar scaling properties in both model size and inference samples (Figure 1). ARBG leverages decoder-only transformer architectures (Radford et al., 2019), and is therefore compatible with other optimizations such as KV-caching and post-training strategies.

Our main contributions are summarized as follows:

- We introduce AUTOREGRESSIVE BOLTZMANN GENERATORS (ARBG), the first scalable, autoregressive and diffeomorphism-free method for Boltzmann Generation.
- We investigate various proposal formulations, demonstrating that discrete binning not only offers superior training stability and scalability compared to continuous mixture models, but also unlocks inference-time interventions—like early rejection—that are not possible in flow-based architectures.
- We demonstrate that ARBG outperforms all baselines on every single peptide benchmark, with particular focus on the large 10-residue Chignolin with predictable scaling laws (depicted in Figure 1).
- We introduce ROBIN, a 132 million parameter autoregressive generative model that achieves zero-shot generalization to unseen peptides, reducing Energy-W2 error, $\mathcal{E}$-$\mathcal{W}_2$, by $\sim$60% compared to the previous state-of-the-art approach on large peptides: Prose, a discrete-time NF (Tan et al., 2025b).

2 AUTOREGRESSIVE BOLTZMANN GENERATORS

We consider an alternative class of generative models for constructing the proposal distribution $p_\theta(x)$ in a Boltzmann Generator—autoregressive (AR) models. We introduce AUTOREGRESSIVE BOLTZMANN GENERATORS, the first diffeomorphism-free AR model for molecules that operates directly on atoms in their native Cartesian coordinates. In the context of BGs, the central challenge of AR modelling arises from the continuous nature of molecular coordinates, in contrast to the discrete data on which AR models are frequently used. This setting, therefore, presents an opportunity for developing novel AR frameworks for continuous-state systems. We further motivate our model choice by identifying two key properties required of a proposal distribution within a Boltzmann Generator:

(1) Fast and Accurate Likelihoods. A core requirement of any proposal is to facilitate IS-based correction of samples—necessitating access to fast, unbiased, and exact likelihood evaluation.

(2) Scalability. We require expressive generative model families $p_\theta(x)$ that can capture the sparse and rugged energy landscape of high-dimensional molecular systems with predictable scaling.

We highlight that both desiderata are demonstrably satisfied by autoregressive models, but not necessarily flow-based models. In particular, autoregressive models *exactly* factorize the joint density over x as a sequence of conditional distributions that is used to predict the next dimension conditioned on the history $x_{<j} = [x_1, \dots x_{j-1}]$, for $j \in [d]$:

$$\log p_\theta(x) = \sum_{j=1}^{d} \log p_\theta(x_j | x_{<j}). \tag{1}$$

Clearly, Eq. 1 allows for exact likelihood computation in a single pass, avoiding Jacobian determinants or computationally expensive numerical solvers for Neural ODEs. Moreover, AR models have achieved empirical success across a spectrum of domains at scale, including large-scale discrete modelling of text (Comanici et al., 2025) and images (Dosovitskiy, 2020). Indeed, the diversity of data domains tackled by AR models comes without specific constraints, such as invertible architectures, or the need to model diffeomorphisms. The latter fact we argue is particularly important for molecular modelling due to the non-smooth nature of the target Boltzmann distribution, $\mu_{\text{target}}(x)$.

Autoregressive Modelling of Conformations. We consider inputs of the form $x \in \mathbb{R}^{n\times 3}$, which are flattened into a single $d = n \times 3$ dimensional vector by an AR model as defined in Equation (1). From here on, we use subscripts such as x_j to denote the j-th dimension of the input vector x rather than a continuous time index as done for CNFs. As AR models require an ordering to model molecular states, we use a residue-by-residue ordering in which the side-chains immediately follow the backbone atoms.

A key technical challenge of instantiating AR models over continuous spaces is determining the parametrization of the conditional distribution over dimensions $p_\theta(x_j|x_{<j})$. We explore several options by leveraging existing ideas from Mixture Density Networks (MDN) (Bishop, 1994), which output the parameters of the conditional distribution as a mixture (our variants are referred to as Mol-PixelCNN++ and GMM-PixelCNN++). In addition, we also offer a novel parametrization utilizing a uniform binning strategy that is simple yet enjoys closer alignment to LLM training—unveiling predictable scaling behaviours, but now in the context of Boltzmann Generation (detailed below).

Uniform Bin Parameterization. While elegant in theory, Mixture Density Networks such as MoL-PixelCNN++ and GMM-PixelCNN++ are prone to mode collapse of the mixture components π_k (Deng et al., 2022) and are therefore treated as baseline approaches relegated to the appendix. Our approach instead adopts a simpler and empirically superior binning parameterization of $p_\theta(x_j \mid x_{<j})$, in which the model directly predicts bin centres b_l as a Categorical distribution. This enables the direct use of an AR model, analogous to next-token prediction in LLMs, but applied to molecular coordinates. At inference, continuity is recovered by adding uniform noise to the sampled bin, $x_j = b_l + u_l$, where $u_l \sim \text{Unif}(\Delta)$. This uniform binning strategy induces the following piecewise-conditional density:

$$p_\theta(\tilde{x}_j|x_{<j}) = \sum_{l=1}^{L} \pi_\theta(b_l|x_{<j}) \frac{\mathbf{1}\{\tilde{x}_j \in b_l\}}{\Delta}, \tag{2}$$

where $\pi_\theta(b_l|x_{<j}) = \text{Cat}(b_0, \dots, b_L)$ and Δ is the bin width. In Equation (2), when Δ is the same for all bins, the conditional log density has a constant offset $\log \Delta$ which vanishes with the number of bins.

Autoregressive Twisted Sequential Monte–Carlo. We employ an autoregressive twisted SMC framework to improve sampling efficiency by steering generation away from physically-implausible sub-

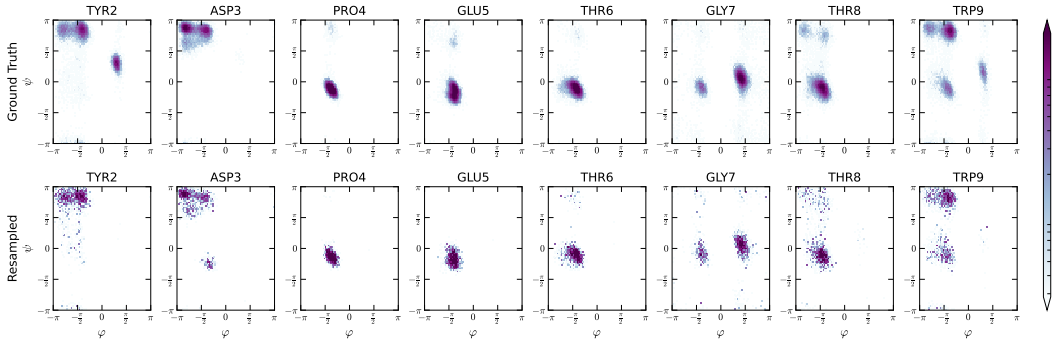

Figure 2: Ramachandran plots for Chignolin (top row: ground truth test set; bottom row: ARBG's predictions).

structures via intermediate twisted distributions defined by partial energy evaluations. By resampling at residue-level granularity, we can use AR model likelihoods to inject physical validity constraints early in the generative process, avoiding wasted computation. Details are provided in Section C.7.

Table 1: Results on tri-alanine (AL3), alanine tetrapeptide (AL4), hexa-alanine (AL6), and Chignolin (GYDPETGTWG). Evaluations are performed with 2×10^5 energy evaluations; all methods except SBG use SNIS. Best values in **bold**, with second-best underlined.

	Tri-alanine (AL3)		Tetrapeptide (AL4)		Hexa-alanine (AL6)		Chignolin (GYDPETGTWG)	
Algorithm	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$	$\mathcal{E}$-$\mathcal{W}_2 \downarrow$	$\mathbb{T}$-$\mathcal{W}_2 \downarrow$
ECNF++	2.206 ± 0.813	0.962 ± 0.253	5.638 ± 0.483	1.002 ± 0.061	10.668 ± 0.285	1.902 ± 0.055	—	—
RegFlow	0.853 ± 0.105	1.577 ± 0.140	3.277 ± 0.546	2.342 ± 0.102	—	—	—	—
SBG	0.598 ± 0.084	0.503 ± 0.029	1.007 ± 0.382	1.039 ± 0.069	1.189 ± 0.357	1.444 ± 0.140	10.819 ± 7.206	3.778 ± 0.440
FALCON-A	1.385 ± 0.182	0.343 ± 0.004	2.929 ± 0.068	1.094 ± 0.034	1.211 ± 0.105	1.163 ± 0.112	—	—
FALCON	0.544 ± 0.013	0.452 ± 0.011	0.686 ± 0.047	0.858 ± 0.077	0.892 ± 0.311	1.256 ± 0.132	—	—
GIVT	1.354 ± 0.058	0.343 ± 0.008	1.033 ± 0.449	1.113 ± 0.100	1.206 ± 0.056	1.527 ± 0.048	45.646 ± 20.989	3.031 ± 0.098
MoL-PixelCNN++	0.506 ± 0.082	1.024 ± 0.686	1.643 ± 0.504	1.415 ± 0.110	1.429 ± 0.186	1.264 ± 0.205	140.717 ± 49.113	3.391 ± 0.093
GMM-PixelCNN++	0.249 ± 0.025	0.364 ± 0.016	1.434 ± 0.783	0.806 ± 0.056	1.164 ± 0.037	1.285 ± 0.058	23.339 ± 6.485	3.007 ± 0.086
ARBG (ours)	$\mathbf{0.202 \pm 0.010}$	$\mathbf{0.312 \pm 0.003}$	$\mathbf{0.449 \pm 0.030}$	$\mathbf{0.592 \pm 0.010}$	$\mathbf{0.328 \pm 0.122}$	$\mathbf{1.094 \pm 0.052}$	$\mathbf{1.723 \pm 0.075}$	$\mathbf{2.632 \pm 0.044}$

3 EXPERIMENTS

In this section, we empirically validate the efficacy of AUTOREGRESSIVE BOLTZMANN GENERATORS across a variety of molecular conformation sampling tasks. Our evaluation focuses on assessing the framework's scalability on single peptides systems ranging from alanine dipeptide to the 10-residue Chignolin in the same experimental data configuration of Tan et al. (2025a). We also evaluate the zero-shot generalization capabilities on unseen sequences using our autoregressive transferable model, ROBIN, on the ManyPeptidesMD introduced in (Tan et al., 2025b).

Baselines. We consider a suite of prior baselines that include the equivariant CNF (Klein et al., 2023c; Klein & Noe, 2024), using the ECNF++ results of Tan et al. (2025a). We also compare against discrete NFs, in RegFlow (Rehman et al., 2025b), and the prior state-of-the-art method for single-system sampling SBG (Tan et al., 2025a). Further, we benchmark performance relative to few-step CNFs, i.e., flow maps (Boffi et al., 2025; Geng et al., 2025): FALCON/FALCON-A, which differ in their training objectives (Rehman et al., 2025a). For ALDP, we also include BoltzNCE, an energy-based model trained via noise-contrastive estimation (Aggarwal et al., 2025). We further train a GIVT (Tschannen et al., 2024), MoL-PixelCNN++, and GMM-PixelCNN++ as described in Section 2, following the same training procedure as ARBG. For the transferable Boltzmann sampling setting, we compare against Timewarp (Klein et al., 2023b), BioEmu (Lewis et al., 2025), UniSim (Yu et al., 2025), TarFlow (Zhai et al., 2025), and the prior SOTA for transferable generation in Prose (Tan et al., 2025b).

Alanine-based Systems. We evaluate the performance of ARBG on conformation sampling tasks for four different alanine-based systems from alanine dipeptide (ALDP) (2 residues) to hexa-alanine (6 residues). We report our results in Table 1, and defer the ALDP results to Table A.3 in §C.4 due to both the simplicity of the dataset and also problems with the dataset construction leading to mode-collapse of models. We find that ARBG comprehensively outperforms on both $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ for all considered systems. We further observe MDN baselines like GMM-PixelCNN++ and GIVT achieve second place, highlighting the overall power of AR models for BG's in comparison to flow-based BGs.

Scaling to Decapeptides. SBG (Tan et al., 2025a) first demonstrated that discrete-time NFs can be scaled to the decapeptide Chignolin—a particularly challenging molecular system due to the existence

Table 2: Quantitative results across peptides of length 4 and 8. All methods evaluated a budget of 10^4 energy evaluations (top) or 2×10^5 (bottom). The best values are **bolded**, with the second-best underlined.

# Residues →	4AA (30 systems)			8AA (30 systems)		
Model ↓	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$
TimeWarp	7.237	2.204	0.993	—	—	—
BioEmu	90.079	2.037	1.479	193.873	4.638	1.601
UniSim	$> 10^4$	2.766	1.733	$> 10^3$	6.156	1.495
ECNF++	10.032	1.121	0.572	—	—	—
TarFlow	1.260	0.924	0.492	11.298	2.733	1.087
Prose	**0.932**	**0.752**	**0.367**	10.038	2.456	0.988
ROBIN (ours)	1.168	0.886	0.471	**4.251**	2.325	**0.943**
ROBIN (ours) SMC	1.079	0.874	0.463	4.263	**2.315**	0.977
2×10^5 evals						
TarFlow	0.929	0.776	0.498	10.826	2.320	1.057
Prose	0.646	**0.607**	**0.349**	9.360	2.019	0.960
ROBIN (ours)	**0.531**	0.649	0.379	**3.615**	**1.902**	**0.882**

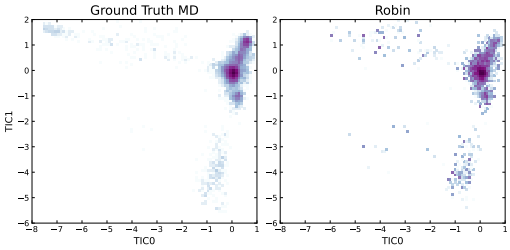

Figure 3: TICA plots demonstrating the strong agreement between the true MD distribution against predictions from ROBIN for a previously-unseen octapeptide obtained from the test set: CGSWHKQR.

of the β-hairpin secondary structure. As shown in Table 1, ARBG significantly outperforms the SBG approach across all global and local evaluation metrics, further demonstrating its scalability. In Figure A.5, we show that the re-weighted energy distribution proposed by our model closely matches that of MD simulation data. In addition, the Ramachandran plots (Ramachandran et al., 1963) in Figure 2 indicate that our model accurately captures nearly all the conformational modes present in the test set.

Transferable Generation. We also introduce ROBIN, a transferable model trained using the ARBG framework with additional conditioning information—detailed in §D.2—to allow zero-shot transfer to unseen peptides in the ManyPeptidesMD dataset (Tan et al., 2025b). We report our results in Table 2, which contain test performance metrics averaged over 30 different sequences of length 4 and 8. Empirically, we observe significantly superior performance over the current state-of-the-art method (Prose) on all molecular systems of size 8 with competitive performance on sequences of size 4.

Inference Scaling. We also investigate the scaling behaviour of inference samples relative to molecular dynamics (MD) and Prose on octapeptides in Figure 1 and Figure A.16. We find that ROBIN performs favourably against Prose and MD *achieving the same performance with an order of magnitude fewer samples vs. Prose and three orders of magnitude vs. MD* in terms of $\mathbb{T}$-$\mathcal{W}_2$. Furthermore, ROBIN also outperforms Prose for the same computational budget (Figure A.16), despite operating on dimensions instead of atoms. We provide further analysis and also investigate the non-monotonic behaviour of the TICA-$\mathcal{W}_2$ for MD in §C.8. Finally, we also provide TICA plots which demonstrate the strong zero-shot performance of ROBIN on an unseen octapeptide compared to ground truth MD in Figure 3.

Performance with Bin Resolution. As we increase the number of bins used in ARBG, the granularity of the coordinates generated by the model increases. In §A, we first discretize the coordinates into a fixed number of bins, then use uniformly sampled noise from each bin to reconstruct these molecules, demonstrating an upper bound on performance for a model with a fixed number of bins. In Figure A.6, for the tri-alanine (AL3) single peptide system, we ablate the number of bins, and empirically validate that an increasing bin count monotonically improves performance on the resampled $\mathcal{E}$-$\mathcal{W}_2$.

Sampling Temperature. We perform inference-time ablations across all models to determine the sampling temperatures that yield optimal performance. As shown in Figure A.6, the energy distribution varies systematically with temperature: at low temperatures, probability mass concentrates on high-likelihood (low-energy) modes, which can suppress or entirely miss modes. At higher temperatures, the distribution flattens, encouraging diversity but over-sampling modes and degrading sample quality. We quantify this trade-off through a temperature sweep, which reveals an optimal temperature of $T = 1.02$ for AL4. We perform equivalent sweeps for all other systems, with optimal temperatures summarized in §C.2, finding that lower temperatures improve performance on larger molecular systems.

4 CONCLUSION

In this work, we introduced AUTOREGRESSIVE BOLTZMANN GENERATORS (ARBG), a novel autoregressive framework for Boltzmann Generators that offers a promising alternative to the dominant paradigm of flow-based approaches. In particular, ARBG offers new tools that circumvent the expressivity and efficiency constraints that hinder discrete flow-based models while being more computationally efficient at likelihood estimation than CNFs. Importantly, ARBG enjoys the same toolkit as LLMs that come with feature-rich optimizations that enable scaling laws for language and token level-steering, which we demonstrate for the first time in the context of Boltzmann Generators.

REFERENCES

Rishal Aggarwal, Jacky Chen, Nicholas M Boffi, and David Ryan Koes. BoltzNCE: Learning likelihoods for boltzmann generation with stochastic interpolants and noise contrastive estimation. In *Neural Information Processing Systems (NeurIPS)*, 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. In *Neural Information Processing Systems (NeurIPS)*, 2025.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *International Conference on Learning Representations (ICLR)*, 2023.

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

Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. Gfn2-xtb-an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of chemical theory and computation*, 15 3:1652–1671, 2018.

Lukas Billera, Anton Oresten, Aron Stålmarck, Kenta Sato, Mateusz Kaduk, and Ben Murrell. The continuous language of protein structure. *bioRxiv*, 2024. doi: 10.1101/2024.05.11.593685.

Christopher M Bishop. Mixture density networks. 1994.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Neural Information Processing Systems (NeurIPS)*, 2025.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Neural Information Processing Systems (NIPS)*, 2018.

Austin H. Cheng, Chong Sun, and Alán Aspuru-Guzik. Scalable autoregressive 3d molecule generation, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pp. 2133–2143. PMLR, 2020.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Haowen Deng, Mai Bui, Nassir Navab, Leonidas Guibas, Slobodan Ilic, and Tolga Birdal. Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation. *International Journal of Computer Vision*, 130(7):1627–1654, 2022.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *International Conference on Learning Representations (ICLR)*, 2017.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Felix Draxler, Peter Sorrenson, Lea Zimmermann, Armand Rousselot, and Ullrich Köthe. Free-form flows: Make any architecture a normalizing flow. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019.

Peter Eastman, Raimondas Galvelis, Raúl P. Peláez, Charlles R. A. Abreu, Stephen E. Farr, Emilio Gallicchio, Anton Gorenko, Michael M. Henry, Frank Hu, Jing Huang, Andreas Krämer, Julien Michel, Joshua A. Mitchell, Vijay S. Pande, João PGLM Rodrigues, Jaime Rodriguez-Guerra, Andrew C. Simmonett, Sukrit Singh, Jason Swails, Philip Turner, Yuanqing Wang, Ivy Zhang, John D. Chodera, Gianni De Fabritiis, and Thomas E. Markland. Openmm 8: Molecular dynamics simulation with machine learning potentials. *The Journal of Physical Chemistry B*, 128(1):109–116, 2024. doi: 10.1021/acs.jpcb.3c06662.

Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 7566–7578. Curran Associates, Inc., 2019.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Johann Flemming Gloy and Simon Olsson. Hollowflow: Efficient sample likelihood evaluation using hollow message passing. In *Neural Information Processing Systems*, 2025.

David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Jérôme Hénin, Tony Lelièvre, Michael R Shirts, Omar Valsson, and Lucie Delemotte. Enhanced sampling methods for molecular dynamics simulations. *arXiv preprint arXiv:2202.04164*, 2022.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017. doi: 10.1088/1361-648X/aa680e.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 2010.

Marlis Hochbruck, Jan Leibold, and Alexander Ostermann. On the convergence of lawson methods for semilinear stiff problems. *Numerische Mathematik*, 2020.

Zhifeng Jing, Chengwen Liu, Sara Y Cheng, Rui Qi, Brandon D Walker, Jean-Philip Piquemal, and Pengyu Ren. Polarizable force fields for biomolecular simulations: Recent advances and applications. *Annual Review of biophysics*, 48(1):371–394, 2019.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6.

Kacper Kapuśniak, Cristian Gabellini, Michael Bronstein, Prudencio Tossou, and Francesco Di Giovanni. Mars-fm: Generative modeling of molecular dynamics via markov state models. In *International Conference on Learning Representations*, 2026.

Leon Klein and Frank Noe. Transferable boltzmann generators. In *Neural Information Processing Systems (NeurIPS)*, 2024.

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

Leon Klein, Andrew Y. K. Foong, Tor Erlend Fjelde, Bruno Kacper Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics. 2023b.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Neural Information Processing Systems (NeurIPS)*, 2023c.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. *International Conference on Machine Learning (ICML)*, 2020.

Dieterich Lawson, Allan Raventós, Andrew Warrington, and Scott Linderman. Sixo: Smoothing inference with twisted objectives, 2022.

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

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Neural Information Processing Systems*, 2024.

Xiaoming Li, Hubert Normandin-Taillon, Chun Wang, and Xiao Huang. Xrmdn: An extended recurrent mixture density network for short-term probabilistic rider demand forecasting with high volatility. *arXiv preprint arXiv:2310.09847*, 2023.

Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How Fast-Folding Proteins Fold. *Science*, 2011. doi: 10.1126/science.1208351.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*, 2023.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Representation Learning*, 2019.

Masakazu Matsumoto, Shinji Saito, and Iwao Ohmine. Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing. *Nature*, 2002.

Laurence Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. SE(3) Equivariant Augmented Coupling Flows. *Advances in Neural Information Processing Systems*, 2023.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *International Conference on Machine Learning*, 2024.

Mhd Hussein Murtada, Z. Faidon Brotzakis, and Michele Vendruscolo. Md-llm-1: A large language model for molecular dynamics. *arXiv*, 2025.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 2019.

Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R. Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 2009. doi: 10.1073/pnas.0905466106.

Simon Olsson. Generative molecular dynamics. *Current Opinion in Structural Biology*, 96:103213, 2026. ISSN 0959-440X. doi: https://doi.org/10.1016/j.sbi.2025.103213. URL https://www.sciencedirect.com/science/article/pii/S0959440X25002313.

Michele Parrinello and Aneesur Rahman. Crystal structure and pair potentials: A molecular-dynamics study. *Physical review letters*, 45(14):1196, 1980.

Stefano Peluchetti. Non-denoising forward-time diffusions, 2021.

Xin Peng and Ang Gao. Flow perturbation to accelerate boltzmann sampling. *Nature Communications*, 16(1):6604, 2025.

Danny Perez, Aidan Thompson, Stan Moore, Tomas Oppelstrup, Ilya Sharapov, Kylee Santos, Amirali Sharifian, Delyan Z Kalchev, Robert Schreiber, Scott Pakin, et al. Breaking the mold: Overcoming the time constraints of molecular dynamics on general-purpose hardware. *The Journal of Chemical Physics*, 162(7), 2025.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *preprint*, 2019.

Aneesur Rahman. Correlations in the motion of atoms in liquid argon. *Physical review*, 136(2A): A405, 1964.

G N Ramachandran, C Ramakrishnan, and V Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 1963.

Seyedeh Fatemeh Razavi, Reshad Hosseini, and Tina Behzad. Frmdn: Flow-based recurrent mixture density network. *arXiv preprint arXiv:2008.02144*, 2020.

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

Danyal Rehman, Oscar Davis, Jiarui Lu, Jian Tang, Michael Bronstein, Yoshua Bengio, Alexander Tong, and Avishek Joey Bose. Efficient regression-based training of normalizing flows for boltzmann generators. *arXiv*, 2025b.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, 2015.

Andrea Rizzi, Paolo Carloni, and Michele Parrinello. Targeted free energy perturbation revisited: Accurate free energies from mapped reference potentials. *The journal of physical chemistry letters*, 2021.

Volker Runde, KA Ribet, and S Axler. *A taste of topology*. Springer, 2005.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Henrik Schopmans and Pascal Friederich. Temperature-Annealed Boltzmann Generators, 2025.

Noam Shazeer. Glu variants improve transformer, 2020.

Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629 – 4640, 2008. doi: 10.1175/2008MWR2529.1.

Saifuddin Syed, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Non-reversible parallel tempering: A scalable highly parallel mcmc scheme. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):321–350, 2021.

Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 2013.

Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 2010.

Charlie B. Tan, Avishek Joey Bose, Chen Lin, Leon Klein, Michael M. Bronstein, and Alexander Tong. Scalable Equilibrium Sampling with Sequential Boltzmann Generators. In *International Conference on Learning Representations (ICLR)*, 2025a.

Charlie B. Tan, Majdi Hassan, Leon Klein, Saifuddin Syed, Dominique Beaini, Michael M. Bronstein, Alexander Tong, and Kirill Neklyudov. Amortized sampling with transferable normalizing flows. In *Neural Information Processing Systems (NeurIPS)*, 2025b.

NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, Hongyu Zhou, Kenkun Liu, Ailin Huang, Bin Wang, Changxin Miao, Deshan Sun, En Yu, Fukun Yin, Gang Yu, Hao Nie, Haoran Lv, Hanpeng Hu, Jia Wang, Jian Zhou, Jianjian Sun, Kaijun Tan, Kang An, Kangheng Lin, Liang Zhao, Mei Chen, Peng Xing, Rui Wang, Shiyu Liu, Shutao Xia, Tianhao You, Wei Ji, Xianfang Zeng, Xin Han, Xuelin Zhang, Yana Wei, Yanming Xu, Yimin Jiang, Yingming Wang, Yu Zhou, Yucheng Han, Ziyang Meng, Binxing Jiao, Daxin Jiang, Xiangyu Zhang, and Yibo Zhu. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale, 2025.

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *ECCV*, 2024.

Christopher von Klitzing, Denis Blessing, Henrik Schopmans, Pascal Friederich, and Gerhard Neumann. Learning boltzmann generators via constrained mass transport, 2025.

Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racanière, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *J. Chem. Phys.*, 2020.

Chen Ning Yang. The spontaneous magnetization of a two-dimensional ising model. *Physical Review*, 85(5):808, 1952.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535. Association for Computational Linguistics, 2021.

Jejoong Yoo and Aleksei Aksimentiev. Improved parameterization of amine–carboxylate and amine–phosphate interactions for molecular dynamics simulations using the charmm and amber force fields. *Journal of chemical theory and computation*, 12(1):430–443, 2016.

Ziyang Yu, Wenbing Huang, and Yang Liu. Unisim: A unified simulator for time-coarsened dynamics of biomolecules, 2025.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing Flows are Capable Generative Models, 2024.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *International Conference on Learning Representations (ICLR)*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, 2019.

APPENDIX

A DATA PREPROCESSING AND ANALYSIS

We analyze the coordinate distribution of each peptide by placing the training data into a fixed number of bins (set to `num_bins` $= 1024$) in Figure A.1.

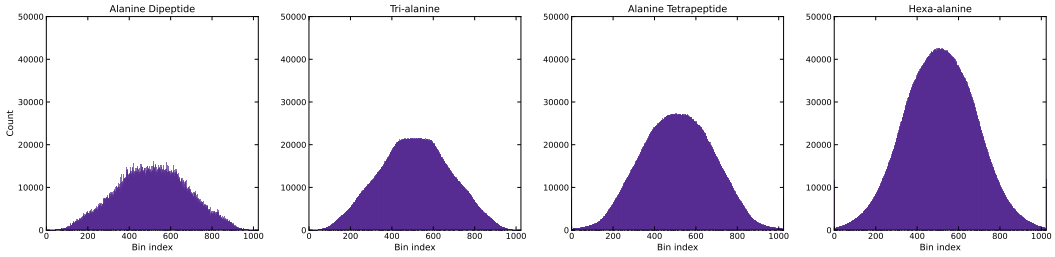

Figure A.1: The distribution of coordinates across bins for fixed bin count with `num_bins` = 1024.

In addition to analyzing the distribution of samples across discretized bins, we evaluated a lower bound on $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ by de-quantizing via uniform noise injection (see Figure A.2). Specifically, training data were discretized into bins and subsequently mapped back to continuous coordinates by reconstructing via uniformly sampled noise. This procedure was repeated over a range of bin sizes to characterize the bin-width–dependent lower bound on attainable performance across all alanine-based peptides. We further conducted ablations to assess the effect of bin size on $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$. As shown in Figure A.3, performance exhibits a near-monotonic dependence on bin resolution, validating our original analysis. Based on these results, we used 4096 bins for alanine dipeptide and tri-alanine, and 8192 bins for larger molecules and ROBIN.

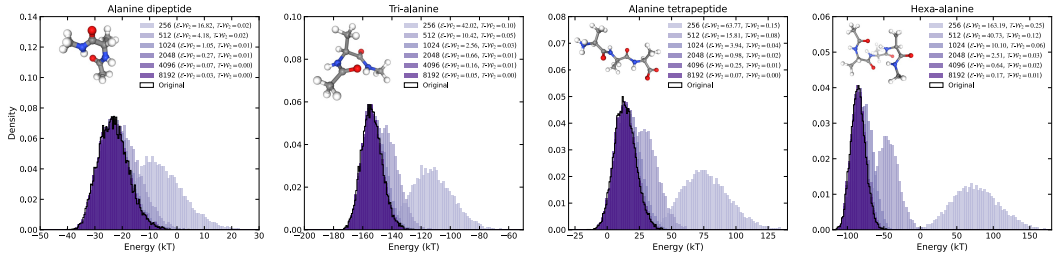

Figure A.2: Energy distributions on the training data as a function of bin discretization. The corresponding $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ values are reported at each level, demonstrating an upper bound on metric learnability.

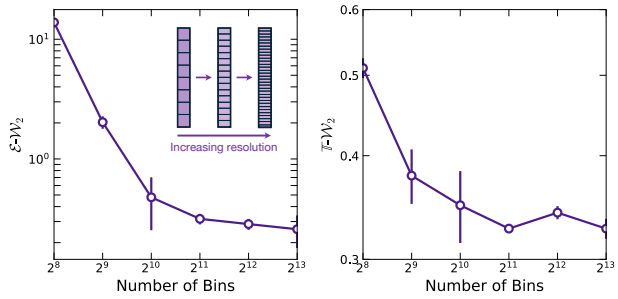

Figure A.3: Model bin count vs. resampled $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ for tri-alanine.

B METRICS

Below, we introduce the metrics used to evaluate model performance and describe their computation. The proposed metrics capture both local and global behaviour. Energy-based metrics assess the accuracy of local interactions, as small geometric perturbations can induce large energy variations. Complementary global metrics—including torus- and TICA-based measures—evaluate mode coverage and the ability of models to capture multi-modal structure. We omit the effective sample size (ESS), as its interpretation is invalidated by the use of SMC.

B.1 MAIN GEOMETRIC METRICS

2-Wasserstein Energy Distance ($\mathcal{E}$-$\mathcal{W}_2$). To quantify the agreement between generated and reference energy distributions, we compute the squared 2-Wasserstein distance between the energies of generated samples and those obtained from MD. Let $p, q \in \mathcal{P}(\mathbb{R})$ denote the probability distributions over energy values for the generated and reference samples, respectively, and let $\Pi(p, q)$ denote the set of admissible couplings between them. The Wasserstein energy distance is then defined as:

$$\mathcal{E}\text{-}\mathcal{W}_2(p,q)^2 \triangleq \min_{\pi \in \Pi(p,q)} \int_{\mathbb{R}\times\mathbb{R}} |x-y|^2 \, \mathrm{d}\pi(x,y). \tag{3}$$

This metric measures how closely the generated energy landscape matches the reference distribution. Because molecular energies are highly sensitive to local structural change such as bond lengths/angles, $\mathcal{E}$-$\mathcal{W}_2$ is particularly effective at detecting physically relevant discrepancies. Lower values correspond to better agreement with the target Boltzmann distribution.

Torus 2-Wasserstein Distance ($\mathbb{T}$-$\mathcal{W}_2$). To assess structural similarity in torsional space, we compute a 2-Wasserstein distance defined on the torus. For a molecule with $L \in \mathbb{N}$ residues, each conformation is represented by its vector of dihedral angles:

$$\text{Dihedrals}(x) = (\phi_1, \psi_1, \ldots, \phi_{L-1}, \psi_{L-1}) \in [0, 2\pi)^{2(L-1)}. \tag{4}$$

To account for the periodicity of angular variables, the squared cost between two conformations x and y is defined as:

$$c_{\mathbb{T}}(x,y)^2 = \sum_{i=1}^{2(L-1)} \left[\left(\text{Dihedrals}(x)_i - \text{Dihedrals}(y)_i + \pi \right) \bmod 2\pi - \pi \right]^2. \tag{5}$$

The corresponding torus Wasserstein distance between two distributions $p, q \in \mathcal{P}([0, 2\pi)^{2(L-1)})$ is then defined as:

$$\mathcal{T}\text{-}\mathcal{W}_2(p,q)^2 \triangleq \min_{\pi \in \Pi(p,q)} \int c_{\mathbb{T}}(x,y)^2 \, \mathrm{d}\pi(x,y). \tag{6}$$

This metric captures global conformational differences in torsional space while respecting angular periodicity. Unlike energy-based distances, $\mathbb{T}$-$\mathcal{W}_2$ is sensitive to missing or misrepresented conformational modes, providing a complementary assessment of structural diversity and coverage in generative Boltzmann models. One point of note is that although this claim generally holds, in cases where there are few samples from a given mode that are lost, this does not substantially impact the $\mathbb{T}$-$\mathcal{W}_2$, meaning that we can see a reduced value even in the presence of mode loss—one clear example of this phenomenon is presented in Section C.4, where we demonstrate mode collapse despite decreasing $\mathbb{T}$-$\mathcal{W}_2$.

TICA 2-Wasserstein Distance (TICA-$\mathcal{W}_2$). To compare the long-timescale dynamical structure of trajectories, we evaluate discrepancies in a reduced space defined by time-lagged independent component analysis (TICA). TICA identifies collective coordinates that maximize autocorrelation, isolating the slow modes governing conformational dynamics.

Given a mean-centered time series $\{\tilde{x}_t\}_{t=1}^T \subset \mathbb{R}^n$ and a lag time τ, we estimate the empirical covariance matrices:

$$\hat{C}_{00} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \tilde{x}_t \tilde{x}_t^\top, \qquad \hat{C}_{0\tau} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \tilde{x}_t \tilde{x}_{t+\tau}^\top. \tag{7}$$

The dominant slow modes are obtained by solving the generalized eigenvalue problem

$$\hat{C}_{0\tau} w = \lambda \hat{C}_{00} w, \tag{8}$$

where each eigenvector w defines a linear projection with maximal normalized autocorrelation at lag τ. In practice, we retain the first two TICA components $\{w_1, w_2\}$, which capture the slowest dynamical processes.

Using these projections, we define an ℓ_2 cost between configurations $x, y \in \mathbb{R}^n$ as their Euclidean distance in TICA space:

$$c_{\text{TICA}}(x, y)^2 = \sum_{j=1}^{2} \left(w_j^\top x - w_j^\top y \right)^2 . \tag{9}$$

The corresponding TICA Wasserstein distance between generated and reference distributions $p, q \in \mathcal{P}(\mathbb{R}^n)$ is then:

$$\text{TICA-}\mathcal{W}_2(p, q)^2 \triangleq \min_{\pi \in \Pi(p,q)} \int c_{\text{TICA}}(x, y)^2 \, \mathrm{d}\pi(x, y). \tag{10}$$

This metric directly assesses agreement in the slow dynamical subspace learned from the reference trajectory. By construction, TICA-$\mathcal{W}_2$ is sensitive to mismatches in metastable state populations and transition pathways, making it well-suited for evaluating models intended to reproduce long-timescale molecular kinetics.

B.2 On the use of Geometric over Likelihood-based metrics in High-dimensions

In this work, we prioritize Wasserstein-based metrics (TICA, Torus, Energy) over likelihood-based metrics. While ESS is a standard diagnostic for the efficiency of SNIS estimators, it is widely recognized as a potentially misleading proxy for sample quality in high-dimensional spaces. In high-dimensional spaces, importance sampling is susceptible to the "curse of dimensionality", referred to as *weight collapse* in the particle filtering literature (Snyder et al., 2008). As the dimensionality increases, the overlap between the typical sets of the proposal and target distributions vanishes exponentially. Consequently, the variance of the importance weights becomes dominated by rare samples that land in the small region of overlap. This results in an estimator variance that explodes, rendering ESS an unreliable metric for performance in systems with hundreds of degrees of freedom (e.g., Decapeptides), as the metric becomes sensitive to global scaling factors rather than local mode coverage.

The Bias Toward Mode Collapse. The core limitation of ESS in the context of Boltzmann Generation is its tendency to reward "mode-seeking" behaviour over "mass-covering" behaviour. ESS is derived from the variance of the importance weights $w(x) = \mu_{\text{target}}(x)/p_\theta(x)$. Specifically,

$$\text{ESS}(x) = \sum_j \mu_{\text{target}}(x^j)/p_\theta(x^j)$$

where $x^j \sim p_\theta(x^j)$. A generative model p_θ can maximize ESS by collapsing its probability mass into a single, highly stable metastable state (a single mode of μ_{target}). In this scenario, the ratio $\mu_{\text{target}}(x)/p_\theta(x)$ remains stable within that specific region, yielding a high ESS; however, this comes at the cost of failing to sample other metastable states (mode dropping). In Figure A.4, we compare the Ramachandran plots between the ground truth MD data, FALCON, and ARBG for one of the torsion angles present in tri-alanine. It can clearly be seen that the mode between 0 and $\frac{\pi}{2}$ exists in the training data, while being lost in FALCON—a model that obtains a higher ESS than ARBG.

Conversely, a model that attempts to cover the full diversity of the Boltzmann distribution ("mass-covering") is much more likely to assign non-zero probability to high-energy regions where $\mu_{\text{target}}(x) \approx 0$. This results in high variance of the importance weights and a low ESS, despite the model being superior in terms of exploring the global conformational space.

On the Interaction with Numerical Error in Practice. In practice, with models that have some numerical error, the variance of importance weights—and therefore ESS—is often dominated by numerical errors. Where a small fraction of samples will compute as having an unusually high likelihood. This causes the creation of a large importance weight. This is why, in practice, all models utilize a form of clipping to ensure reasonable importance weight values and to prevent collapse. In this work, we use a clipping value of 0.002 where the samples with the largest 0.002

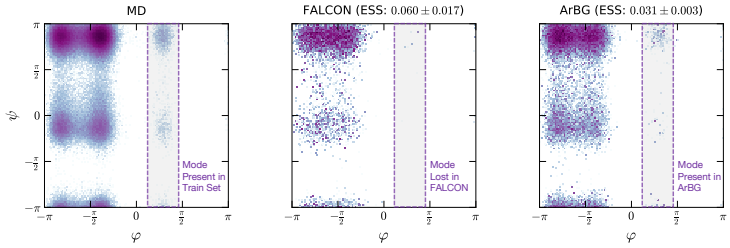

Figure A.4: Left: Ramachandran plot from the ground truth MD data for tri-alanine; **Center**: FALCON's torsion angle predictions; **Right**: Torsion angle predictions from ARBG. FALCON clearly loses one of the conformational modes at inference.

fraction of importance weights are clipped following prior work (Klein et al., 2023c; Midgley et al., 2023; Tan et al., 2025b; Rehman et al., 2025a). In practice, ESS is highly sensitive to numerical precision and is often dominated by outliers, particularly in high-dimensional settings. By contrast, geometry-based metrics are substantially more robust to such numerical effects and provide a more reliable characterization of global model behaviour.

C ADDITIONAL RESULTS

Below, we demonstrate learned proposal and re-sampled energy distributions by ARBG across all alanine-based systems and Chignolin. We observe near perfect alignment between the re-weighted energy distribution and the ground truth MD distribution across all analyzed systems.

C.1 ENERGY DISTRIBUTION OF SINGLE PEPTIDE SYSTEMS

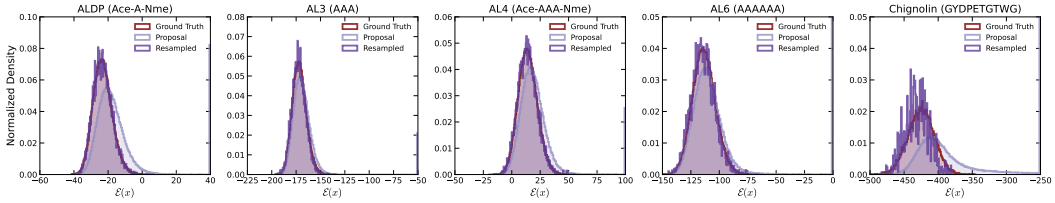

Figure A.5: Energy histogram of the proposal and re-sampled distribution compared to ground truth MD data for all single peptide systems considered (alanine dipeptide all the way to the decapeptide Chignolin).

C.2 TEMPERATURE TUNING

Sampling temperature controls the entropy of the model distribution by scaling logits at inference time. The optimal temperatures for all systems are reported in Table A.1. For smaller systems, temperatures near 1.0 are sufficient, with slight gains observed for values marginally above 1.0. In contrast, larger systems benefit from lower temperatures, which likely mitigate underfitting.

Table A.1: Optimal sampling temperatures identified via inference-time temperature sweeps across systems.

System	Optimal Temperature
Alanine Dipeptide	1.03
Tri-alanine	0.99
Alanine Tetrapeptide	1.02
Hexa-alanine	0.98
Chignolin	0.88
ROBIN (Transferable)	0.95

Temperature on Alanine Tetrapeptide. In Figure A.6, we study the effects of temperature tuning on the proposal and re-weighted distributions on the alanine tetrapeptide system. First, as noted in the

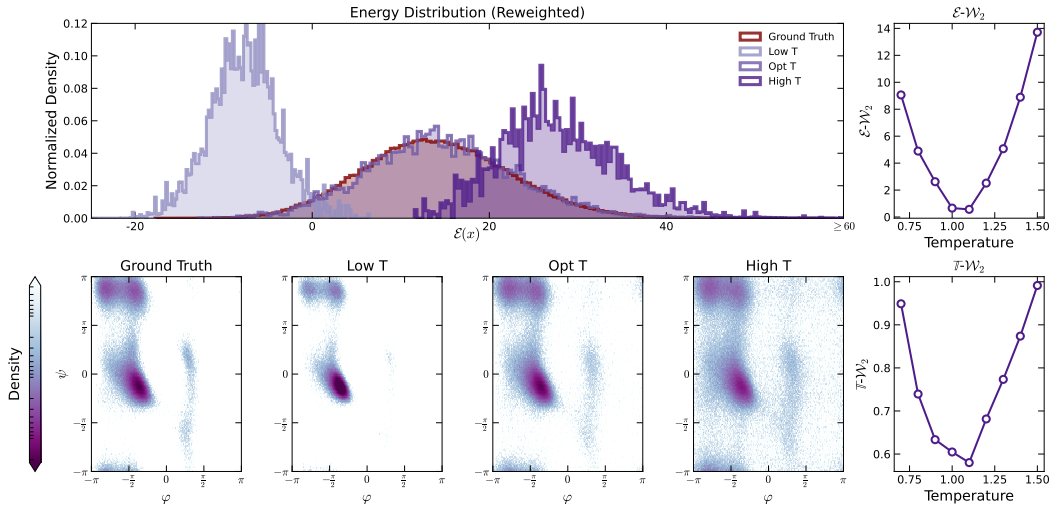

Figure A.6: Ablations on model temperature. For the energy distribution, we demonstrate that lower temperatures sample lower energy modes more frequently, while the converse holds for higher temperatures. We also show how the modes become more prominent at high temperatures and are lost at lower temperatures. Finally, we show how an optimal temperature exists for optimizing $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$.

main text, we find that the optimal temperature is slightly higher than 1.0. This is interesting and in line with prior works that find a slightly more diffuse proposal may be slightly better for Boltzmann Generation metrics, as it allows better coverage of the space. We also demonstrate the over-sampling of modes that occurs with higher temperature proposals in the Ramachandran plots provided.

Table A.2: Inference speed in samples per second for best performing models in the transferable setting. ROBIN is around 50% faster than Prose per model evaluation, but is slower in terms of samples per second due to operating over dimensions instead of atom coordinates and therefore requires $3\times$ the model evaluations.

	2AA	4AA	8AA
TarFlow	737	329	126
PROSE	338	158	66
ROBIN	260	87	29

C.3 INFERENCE TIME

In Table A.2, we report inference throughput (samples per second) for transferable models with 2, 4, and 8 residues, averaged over 30 systems. While ROBIN is slower than Prose, its substantially higher sample quality yields superior performance under a fixed sampling budget (see Figure 1).

C.4 ALANINE DIPEPTIDE

Below, we summarize the results of all ARBG variants in conjunction with other baselines on ALDP. ARBG outperforms all competing models—including both discrete-time normalizing flows as well as CNFs, on $\mathcal{E}$-$\mathcal{W}_2$, with competitive performance on $\mathbb{T}$-$\mathcal{W}_2$.

Mode Collapse. When training models on the alanine dipeptide dataset from Klein et al. (2023c), we observe that we can continue improving our performance across both global and local metrics if we train our models for longer; however, in this process, part of the performance improvement comes from losing the mode, which artificially inflates ESS. Torus, which is designed to be a global metric, also suffers given that there are an insufficient number of points in that mode to radically impact the degradation of performance, yielding a nearly monotonic trend in performance improvement as training time increases. For larger systems, like tri-alanine and above, this behaviour is not observed.

In Figure A.7, we provide a clear demonstration of the lost mode on ALDP. The Ramachandran plots are shown for two different instances in the training process—Epoch 110 and Epoch 370. We show

Table A.3: Results on alanine dipeptide. Best results are **bolded**, with second-best underlined.

	Alanine dipeptide (ALDP)	
Algorithm ↓	$\mathcal{E}$-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓
BoltzNCE	0.27 ± 0.02	0.57 ± 0.00
SE(3)-EACF	108.202	2.867
ECNF	0.419	0.311
RegFlow	0.501 ± 0.011	0.951 ± 0.054
ECNF++	0.914 ± 0.122	0.189 ± 0.019
SBG	0.741 ± 0.189	0.431 ± 0.141
FALCON-A	0.512 ± 0.038	0.180 ± 0.005
FALCON	0.225 ± 0.104	0.402 ± 0.021
GIVT	0.256 ± 0.033	$\mathbf{0.175 \pm 0.171}$
MoL-PixelCNN++	1.447 ± 0.277	0.528 ± 0.028
GMM-PixelCNN++	0.763 ± 0.118	0.354 ± 0.098
ARBG	$\mathbf{0.209 \pm 0.041}$	0.402 ± 0.008

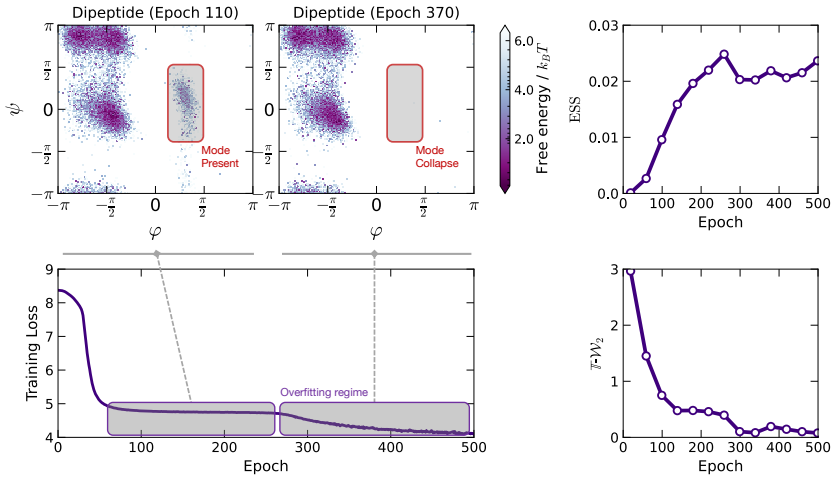

Figure A.7: We demonstrate that training models for too long leads to overfitting on the training data, which despite improving resampled metrics, yields undesirable behaviour.

that earlier in training, the mode exists, but as training continues, it dissappears. This can also be observed on the training loss curve as annotated. We also provide the ESS and Torus results during training to demonstrate that the loss of the mode improves performance on metrics.

We believe this stems from the method used to generate the dataset in Klein et al. (2023c). Specifically, the dataset was generated in the following way:

1. MD simulation using *Amberff99SBildn* force-field at 300K for 1 ms using `openMM` Eastman et al. (2024) with a timestep of 1 femto-second.
2. Relaxation of 10^5 uniformly randomly selected states from the MD data for 100 femto-seconds each using the *GFN2-xTB* forcefield Bannwarth et al. (2018) and the ASE library Hjorth Larsen et al. (2017) with a friction constant of 0.5 a.u.
3. To make the density nearly equal between negative and positive φ dihedral angles, importance sampling is performed using weights from a von Mises distribution f_{vM}. Specifically, weights for each sample are computed as:

$$\omega(\varphi) = 150 f_{vM}(\varphi | \mu = 1, \kappa = 10) + 1 \tag{11}$$

with 10^5 training samples drawn from the weighted distribution.

Specifically, this final reweighting step makes it possible for powerful models to overfit on the positive φ mode. The reweighting step causes there to be multiple instances of exactly the same data sample

in the training set. For powerful models, seeing the same datapoint multiple times (even with data augmentation) causes overfitting. We observe that the likelihood of these exact training samples explodes, causing the distribution after importance sampling to remove the positive φ mode.

Recommendation. For newer and more powerful Boltzmann Generator models, we recommend using training sets without importance sampling, as these are much more difficult to overfit on specific training samples. It is important to be mindful of overfitting-type behaviour on these small datasets with relatively powerful models.

C.5 Ramachandran Plots for Other Single Peptide Systems

Here, we demonstrate the competitive performance of ARBG across single peptide systems by showing the Ramachandran plots for all systems considered. In all cases considered, ARBG captures nearly every mode present in the test data, clearly illustrating the quality of the learned likelihoods and their synergy with SNIS.

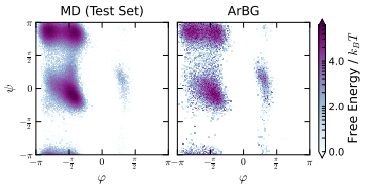

Figure A.8: Left: Test data for alanine dipeptide; **Right**: ARBG's angular predictions for alanine dipeptide.

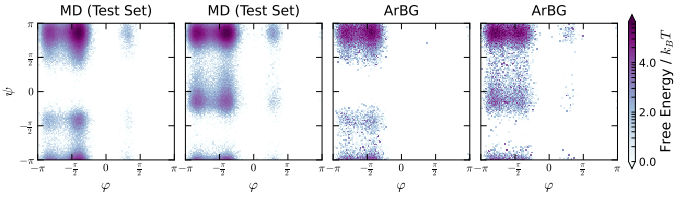

Figure A.9: Left: Test data for tri-alanine; **Right**: ARBG's angular predictions for tri-alanine.

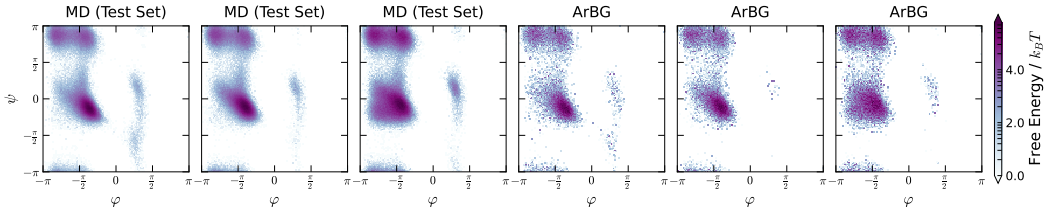

Figure A.10: Left: Test data for alanine tetrapeptide; **Right**: ARBG's angular predictions for alanine tetrapeptide.

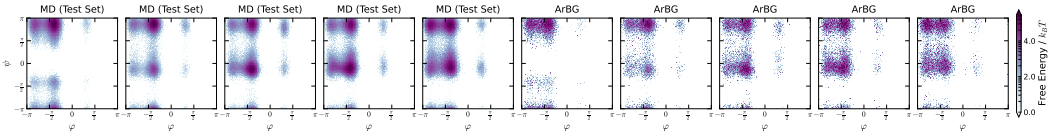

Figure A.11: Left: Test data for hexa-alanine; **Right**: ARBG's angular predictions for hexa-alanine.

C.6 De-quantization Strategies

For ARBG to generate the continuous Cartesian coordinates of molecular systems, we explored three different sampling strategies that we detail below:

1. Sampling uniformly from the discrete bin selected by the model. This is our reasonable default choice and defines a piecewise-constant continuous density in $\mathbb{R}^d$.
2. Using the center value of the bin. This strategy reduces a bit of variability from generation and may make bond lengths slightly more uniform. With enough bins, this is not necessary. Empirically, we observed small benefits for this de-quantization strategy over uniformly sampling from the bin.
3. Using the biased training data distribution for the chosen molecule to determine empirical offsets that we apply to the chosen bin at inference time. Especially for larger bin sizes, we may be able to fit more interesting distributions concerning the training dataset within the bin. Here, we use a Dirac distribution over the empirical mean of the training dataset within the bin. We find this gives a small boost in performance, especially when the training data is centered.

C.7 Transferable Generation

Autoregressive Twisted SMC Efficiency Benefits. We evaluate the efficiency gains that can be obtained using our Autoregressive Twisted SMC algorithm. The main idea is that it is easy to detect samples that will not result in valid low energy samples early on during inference. Using twist functions, which define state-dependent transformations (or change of measure) that are applied to particle weights to bias sampling toward rare or important events and reduce variance, we are able to essentially stop inference early for any sample that exhibits a high partial energy.

In Figure A.12, we investigate the partial energy distributions on the sequence SQQKVAFE test set peptide using ROBIN. To demonstrate this, we perform SMC inference without resampling for 10,000 generations. We record the partial energy of each sample at each residue checkpoint. We find that residue 2 has around 2% of samples that have poor energy samples while residue 7 has $> 7\%$ of samples that have high energies and likely represent steric clashes or other high energy features. These samples represent "wasted" compute, in that they will not contribute to the final distribution of samples. Therefore, additional efficiency can be gained by filtering these out early. On this peptide, using the $\min(\mathcal{E}(x)) + 100$ filter on energy at the earliest time a sample is registered as high energy, we find a savings of roughly 3% over a method without intermediate resampling.

While this is a relatively minor saving at this scale, we expect that for more complex, larger molecular systems where the proposal has more failure modes, the advantage of autoregressive SMC would substantially increase cost savings or sampling efficiency, depending on the exact method and utilization of this concept. Further, this is a unique feature of autoregressive models, whereby this sequential rejection of unfavourable candidates can be performed to limit energy evaluations.

TICA Plots for Unseen Octapeptides. To demonstrate the quality of ROBIN and its zero-shot performance one larger molecular systems, we include TICA plots for seven different unseen octapeptides in Figure A.13. In this process, we show the predictive capacity of ROBIN as it nearly perfectly captures all the modes of these unseen systems, speaking to the quality of the learned model.

Peptide-level Performance and Learnability. We evaluate the learned model on each peptide in the test set and report the resampled $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ in Figures A.14 and A.15 for ROBIN, Prose, and TarFlow. Overall, performance is broadly comparable across models on a per-sequence basis; however, several peptides remain challenging for all methods. For example, all models perform poorly on the $\mathcal{E}$-$\mathcal{W}_2$ for KRRGFFLE. Further analysis indicates that sequences with substantial charge contributions are particularly difficult to learn, especially those containing R and Y residues. Although all amino acids incur steep energy penalties outside favourable conformations, certain side chains are exceptionally sensitive to small geometric perturbations. Arginine's planar, highly charged guanidinium group exhibits strongly orientation-dependent electrostatics, while tyrosine's aromatic ring engages in highly directional non-bonded interactions (Yoo & Aksimentiev, 2016; Jing et al., 2019). Consequently, minor deviations in side-chain geometry can produce large energy fluctuations, complicating the learning of equilibrium conformational distributions for these residues.

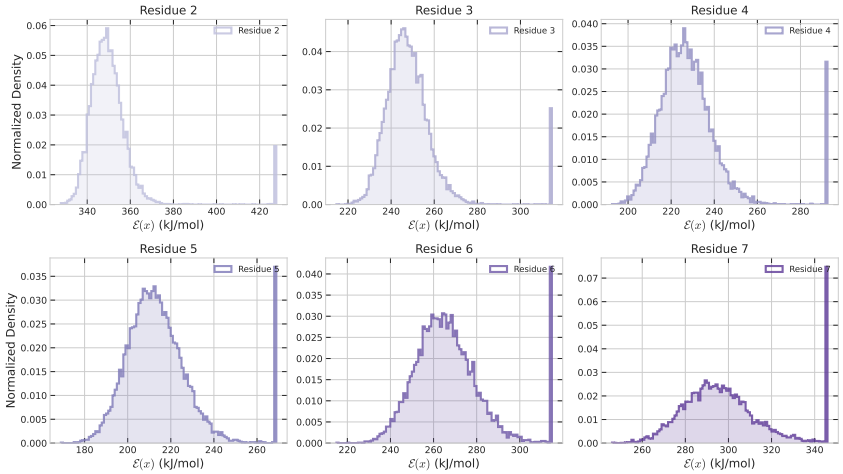

Figure A.12: Histogram of the energy values for 10,000 samples on SQQKVAFE for each residue, where we perform resampling for SMC. We clip the maximum energy to $\min(\mathcal{E}(x)) + 100$. The spikes represent all values greater than or equal to that histogram value. We can see that by residue 7, $> 7\%$ of samples have extremely large energies, which likely represent clashes or erroneous bond lengths.

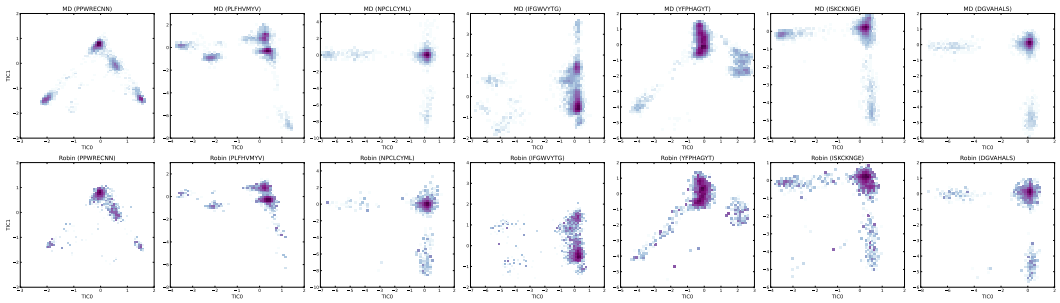

Figure A.13: TICA modes using ROBIN on seven unseen octapeptides (from left to right): PPWRECNN, PLFHVMYV, NPCLCYML, IFGWVYTG, YFPHAGYT, ISKCKNGE, DGVAHALS.

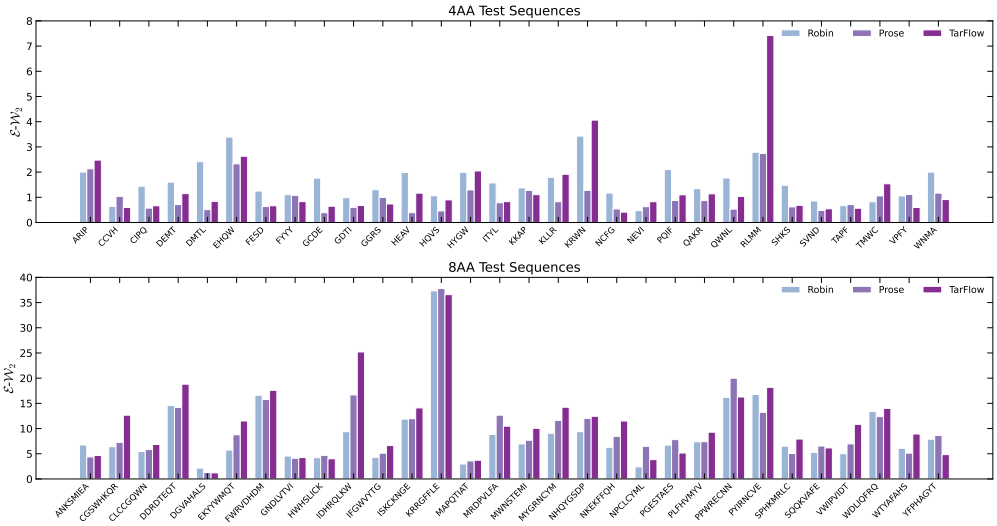

Figure A.14: The resampled $\mathcal{E}$-$\mathcal{W}_2$ across peptides when comparing ROBIN, Prose, and SBG. Models were evaluated using 10^4 samples, with tetrapeptides in the top plot and octapeptides in the bottom plot.

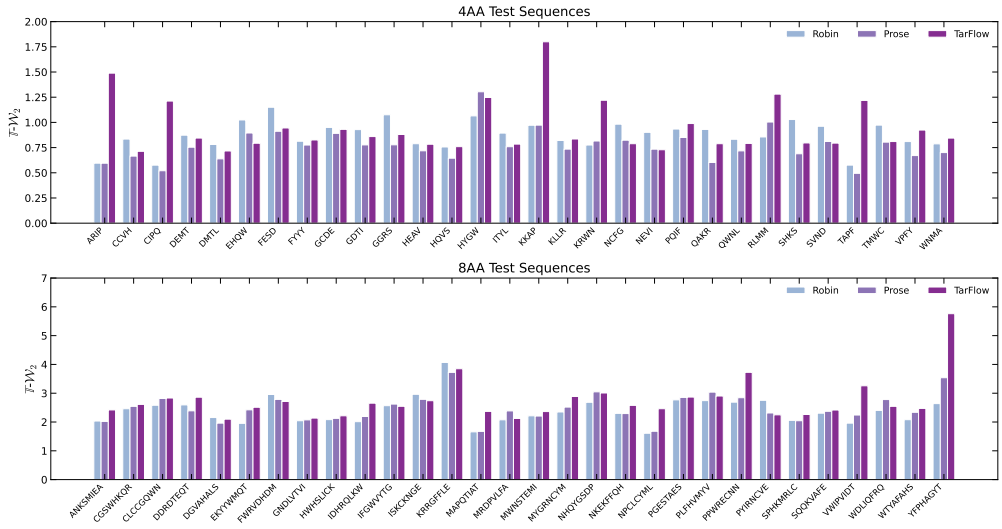

Figure A.15: The resampled $\mathbb{T}$-$\mathcal{W}_2$ across peptides when comparing ROBIN, Prose, and SBG. Models were evaluated using 10^4 samples, with tetrapeptides in the top plot and octapeptides in the bottom plot.

C.8 INFERENCE SCALING

In Figure 1, we demonstrated the scaling performance of ROBIN against Prose and MD on unseen octopeptide systems in terms of the number of function and energy evaluations. In this section, we continue an investigation into the inference scaling behaviour of ROBIN on octopeptides.

GPU hour performance. In Figure A.16, we investigate not only an equal number of energy evaluations, but also the scaling performance in terms of GPU hours. While ROBIN is slower than Prose or MD per model or energy function evaluation, its performance for the same number of GPU hours is better, especially on the $\mathbb{T}$-$\mathcal{W}_2$.

Here, and in Figure 1, we notice an unexpected trend in the TICA-$\mathcal{W}_2$ plots. Specifically, the TICA-$\mathcal{W}_2$ is relatively flat for MD, and then unexpectedly spikes at around 10^7-10^8 energy evaluations. We investigate this further by breaking the performance out sequence by sequence in Figures A.18 to A.20, where we show all 30 test set eight residue sequences. We notice a few interesting things, as stated in the following:

1. As noted in Figures A.14 and A.15, the variability between sequences is quite high. Often, the performance is non-monotonic. For some sequences, one model is better than another, particularly at low energy evaluations; however, as the number of energy evaluations grows, ROBIN generally outperforms Prose.
2. TICA-$\mathcal{W}_2$ for MD often has extremely non-monotonic elements often after 10^7 energy evaluations due to mode jumps. Specifically, it takes around 10^7 MD steps for chains to jump to the next mode. This often drastically changes TICA-$\mathcal{W}_2$ as the relative weights between modes evolve, especially when the new mode has less free energy than the starting mode. We demonstrate this clearly in Figure A.17, where we see the MD over-sample a mode that is incorrectly weighted, leading to significant spikes in the TICA-$\mathcal{W}_2$.

We note that these plots are compared to the test set trajectories which have been run for 50 times as long as the longest MD chain. Given that we see the first mode mixing events at around 10^7 energy evaluations, this implies that the test chains may not be fully mixed.

An interesting difference between MD and Boltzmann Generator traces is that the latter are often more monotonic than the MD trajectories. This is reasonable as Boltzmann Generators (both Prose and ROBIN) sample independently from their proposal where MD is autocorrelated. This means that the performance is often significantly better for BG type models in the few-step regime for unseen peptides. ROBIN outperforms all others on 8 residue systems on average.

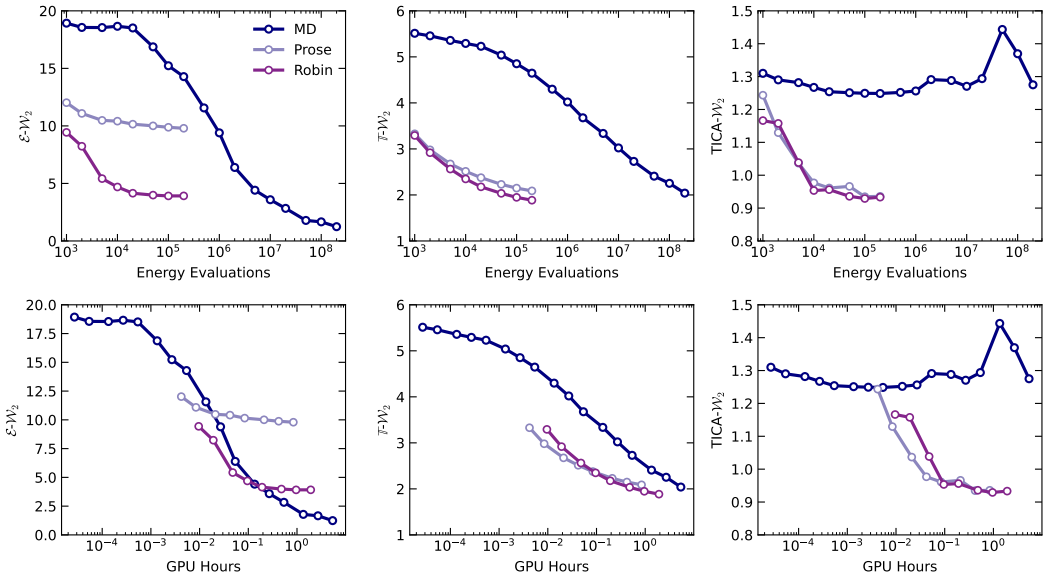

Figure A.16: $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, and TICA-$\mathcal{W}_2$ against the number of energy evaluations and GPU hours for MD, Prose, and ROBIN, demonstrating the fewer energy evaluations needed for the Boltzmann Generators to outperform MD.

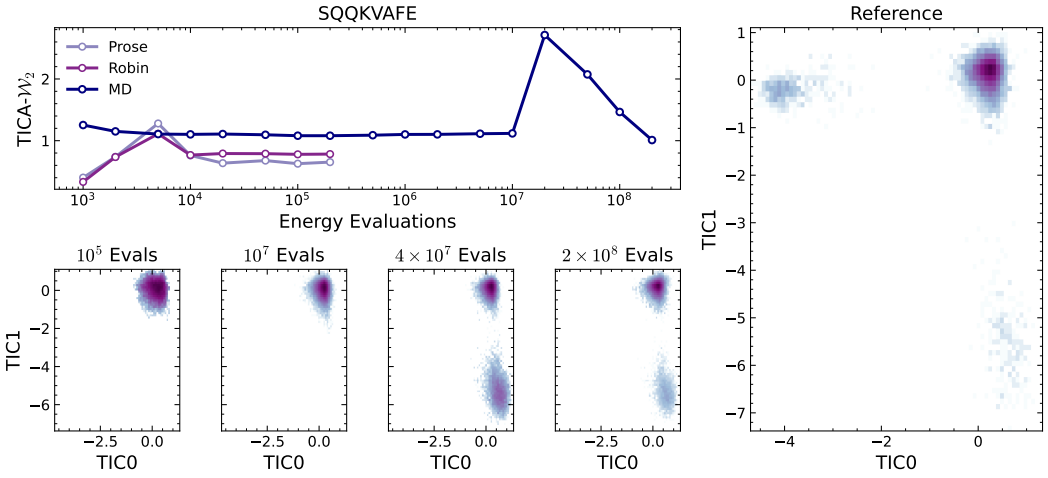

Figure A.17: How the TICA-$\mathcal{W}_2$ varies as a function of energy evaluations across MD, Prose, and ROBIN. In addition, in the bottom row we see how the MD simulation slowly discovers modes as more energy evaluations are performed; in this process, it often searches and samples in incorrect regions, amplifying the TICA-$\mathcal{W}_2$, and then recovering with additional samples when it recognizes that these are energetically unfavourable regions.

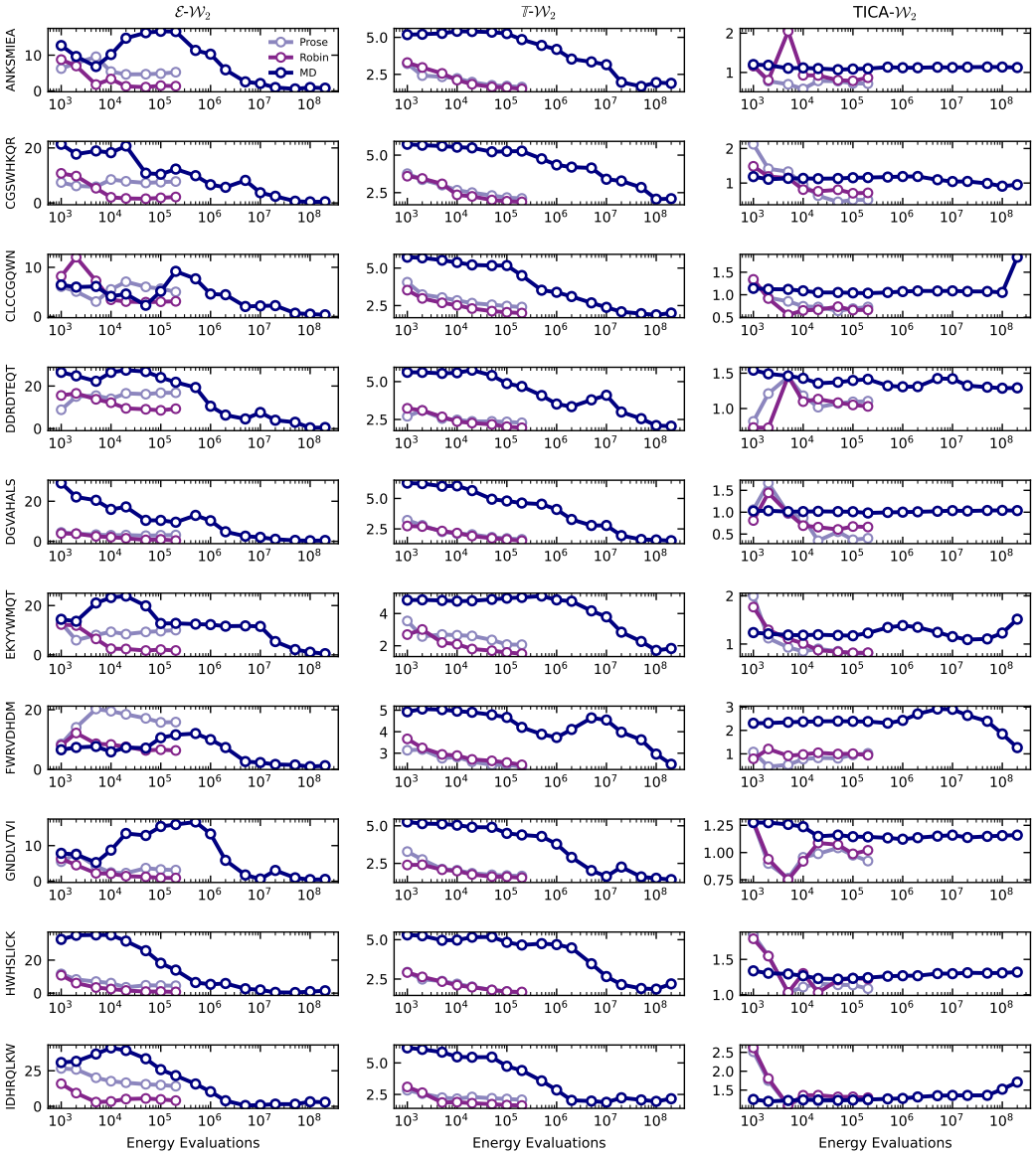

Figure A.18: $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, TICA-$\mathcal{W}_2$ per peptide vs. Energy Evaluations.

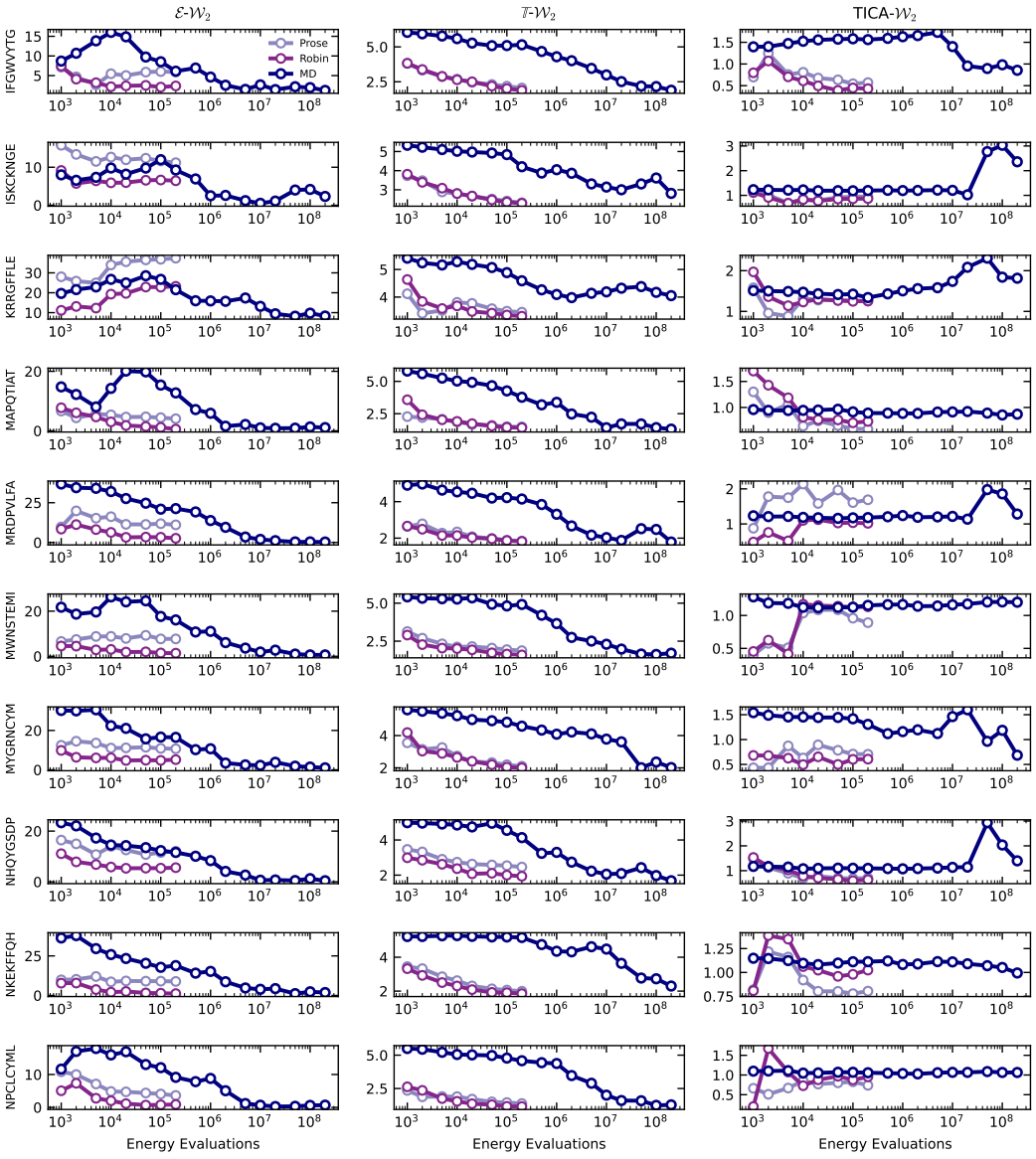

Figure A.19: $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, TICA-$\mathcal{W}_2$ per peptide vs. Energy Evaluations.

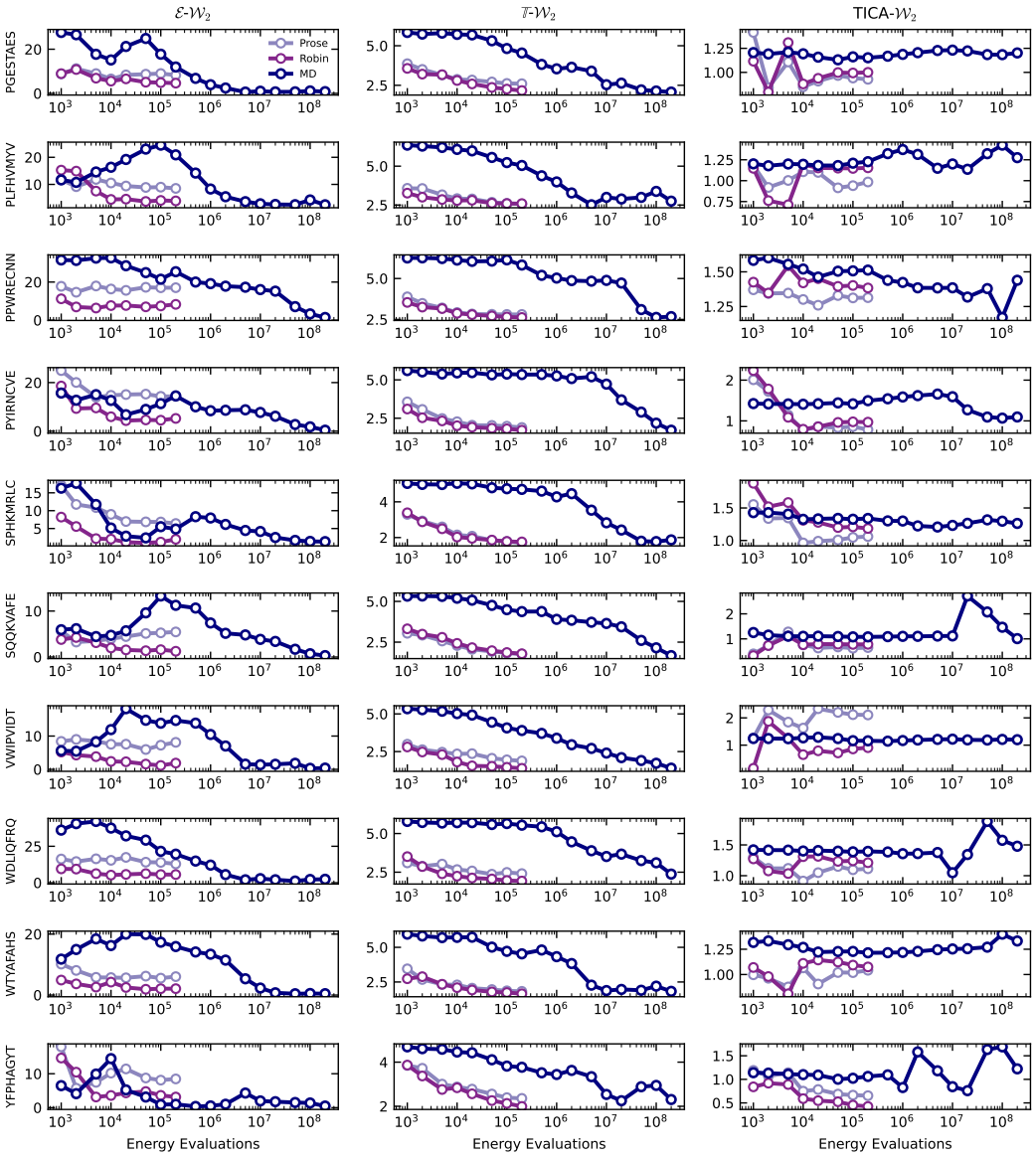

Figure A.20: $\mathcal{E}$-$\mathcal{W}_2$, $\mathbb{T}$-$\mathcal{W}_2$, TICA-$\mathcal{W}_2$ per peptide vs. Energy Evaluations.

D EXPERIMENTAL CONFIGURATIONS

D.1 ARCHITECTURE

The architecture choice employed follows a standard but performant recipe of Transformer-based building blocks. In Figure A.21, we capture the model specifications within a Transformer Block, which models the conditional distribution $p_\theta(x_j|x_{<j})$ and is composed of causal self-attention, RMSNorm (Zhang & Sennrich, 2019), and SwiGLU activations (Shazeer, 2020). The two main differences between the set of single peptide experiments and the transferable setting were: (**1**) the scale of the model; and (**2**) the conditioning, which we cover in detail below. Unique to the molecular setting, we include an additional source of conditioning through embeddings of the atom type and residue types that are injected into the main transformer block. We discuss details below.

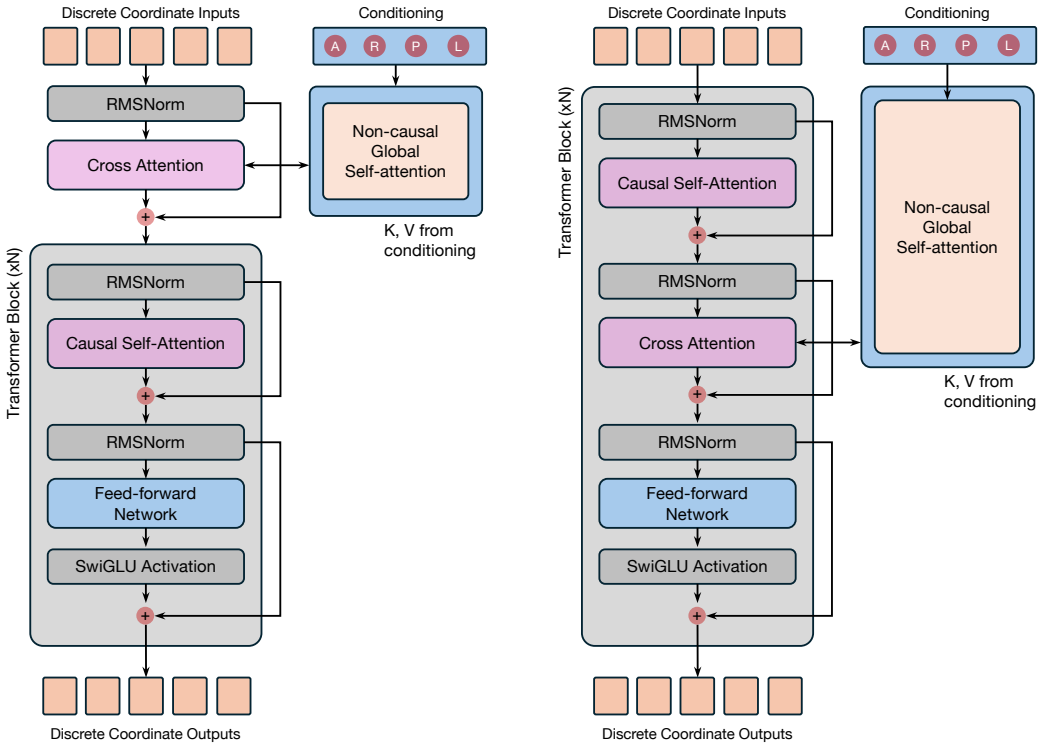

Figure A.21: The transformer-based architecture variants that were considered. **Left**: The decoder-only like architecture that takes the conditioning information in at the initial generation step; **Right**: The encoder-decoder like architecture that repeatedly has a cross attention block that interacts with every transformer layer.

D.2 CONDITIONING

In the transferable setting, we explore various conditioning strategies. In line with Tan et al. (2025b), the conditioning information considers: (**1**) atom type, A; (**2**) residue type, R; (**3**) residue position, P; and (**4**) sequence length, L. To feed this conditioning information into the transformer block, we consider two approaches that capture the ethos of encoder-decoder and decoder-only architectures commonly employed in modern state-of-the-art LLMs.

More specifically, in the decoder-only variant (see left schematic of Figure A.21), we pass the conditioning information through a separate transformer model with non-causal masking (token by token predictions should have access to global conditioning information). The representations obtained from the transformer are subsequently injected directly into the first layer of the larger transformer. For language, most conditional information is passed in at the beginning of the model as context, with additional passing at each layer excluded.

For encoder-decoder models—commonly adopted for settings like machine translation—where global conditioning information is useful for model predictions, two separate transformers are used for both

encoding and decoding, with the interaction between them governed by a learned cross attention module. In a similar fashion to the decoder-only variant, we use a non-causal transformer to obtain representations for the conditioning information, and then compute cross attention by varying the query vectors of our larger model for fixed key and value vectors from our non-causal transformer. This approach allows us to more explicitly distinguish between information types (coordinate-type information being processed by the main transformer vs. residue-level information being processed by the non-causal transformer).

D.3 Model Sizes

ARBG and ROBIN. For all single peptide and transferable generation experiments, we concluded upon the model configurations reported in Table A.5. For ease of contrast, we include the configurations used for competing models: SBG and Prose in Table A.4. In addition, although the model configurations appear identical between all alanine datasets for ARBG and ROBIN, the difference in parameter count can be attributed to the number of bins used. As stated in Table A.5, for alanine dipeptide and tri-alanine, we use 4096 bins, while for alanine tetrapeptide and larger, we use 8192 bins.

Table A.4: SBG and Prose configurations across molecular systems (Tan et al., 2025a;b).

System	Layers / Block	Blocks	Channels	Parameters (M)
SBG (Alanine Dipeptide)	4	4	256	13
SBG (Tri-alanine)	6	6	256	29
SBG (Alanine Tetrapeptide)	6	6	384	64
SBG (Hexa-alanine)	6	6	384	64
SBG (Chignolin)	8	8	384	114
Prose (Transferable)	8	8	384	285

MoL/GMM-PixelCNN++ and GIVT. For fair comparison, with ARBG we inherit the same de-quantization strategy and architectural blocks as ARBG when constructing the MoL/GMM-PixelCNN++ baselines. For all single peptide experiments, we set the bin count to $|B| = 2048$. For GIVT, as it is a fully continuous model in the vein of a true Mixture Density Network, there is no de-quantization needed. In all three baselines, the model includes an additional output projection head that outputs the parameters of the mixture distribution and has the shape:

$$\text{outputproj} = 3K + 2, \tag{12}$$

where K is the number of mixtures, and for each mixture we output the means, scales, and logits over the mixture components. Lastly, we use two additional parameters as dependency coefficients to model the linear dependency coefficients for better modelling of correlated coordinates. In each case, the models are trained by computing the negative log likelihood under the mixture distribution. All remaining training settings are identical to the main model ARBG for single peptide systems.

D.4 Training

Optimizer, Learning Rate, and Scheduler. Following the recent success of the Muon optimizer in accelerating LLM training (Jordan et al.), we adopt it in our experiments. Muon is a momentum-based optimizer that applies Newton–Schulz orthogonalization to gradient updates. For weight matrices, it maintains an orthogonalized momentum buffer computed using five Newton–Schulz iterations, with Nesterov momentum ($\mu = 0.95$) and a learning rate of 0.02. For one-dimensional parameters, such

Table A.5: ARBG and ROBIN configurations for single system experiments and transferable sampling.

System	Heads	Head Dim.	Layers	Channels	Expansion	Parameters (M)	Bins
ARBG (Alanine Dipeptide)	8	32	8	256	4	7.4	4096
ARBG (Tri-alanine)	8	32	8	256	4	7.4	4096
ARBG (Alanine Tetrapeptide)	8	32	8	256	4	10.5	4096
ARBG (Hexa-alanine)	8	32	8	256	4	10.5	8192
ARBG (Chignolin)	8	64	12	512	4	39.9	8192
ROBIN (Transferable)	8	64	16	768	4	132.0	8192

as biases and normalization layers, the optimizer falls back to AdamW (Loshchilov & Hutter, 2019) with a learning rate of 0.002 and $(\beta_1, \beta_2) = (0.9, 0.999)$. We apply decoupled weight decay of 0.01 to all parameters and combine the optimizer with a cosine learning rate schedule with a warm-up phase covering 5% of the training iterations.

Flash Attention. To improve training efficiency and reduce memory consumption, we employ FlashAttention for all models (Dao, 2024). FlashAttention computes the attention operation using fused kernels that significantly reduce the number of memory reads/writes by avoiding the explicit materialization of attention matrices. This substantially reduces memory overhead and improves throughput, enabling faster training and better hardware utilization without altering the underlying attention computation or model behaviour.

Lower precision training and inference. We tested training in both bf16 and float32. We found that for smaller models, bf16 performed equally well to float32 training. However, for larger models, and primarily ROBIN, we observed that bf16 training sometimes resulted in training instability. We therefore chose to train ROBIN using float32 precision. We also found that for inference, float32 inference was more consistent than bf16, which anecdotally had some numerical irregularities.

Hardware. Training and inference was performed across multiple heterogeneous clusters containing a variety of NVIDIA GPUs. We primarily utilized L40S and RTX6000Pro GPUs for inference and training of single-system ARBG models and H100/H200 GPUs for training of ROBIN.

D.4.1 SINGLE-SYSTEM

For the Chignolin scaling plot in Figure 1, we removed the scheduler, fixed the learning rate to 3×10^{-4}, and set the batch size to 256 across all trained models to ensure a fair comparison.

D.4.2 ROBIN TRAINING

We train ROBIN using the same number of training steps as Prose, with a comparable batch size of 448 (7/8 of the Prose batch size 512). We use a learning rate 5×10^{-4}, with a cosine annealing schedule without weight decay.

D.5 INFERENCE

D.5.1 AUTOREGRESSIVE TWISTED SEQUENTIAL MONTE CARLO

In this section, we detail our usage of SMC and the details on how it is applied in our peptide setting. In the ideal setting with a terminal reward $\mu_{\text{target}}(x)$, we would directly have access to the optimal intermediate density i.e.

$$\eta_j^\star \propto p_\theta(x_{\leq j})\psi_j^\star(x_{\leq j}). \tag{13}$$

with optimal twist functions:

$$\psi_j^\star(x_{\leq j}) \propto \sum_{x_{>j}} p_\theta(x_{>j}|x_{\leq j})\mu_{\text{target}}(x) \tag{14}$$

However, these optimal twist functions are, in general, difficult to obtain. While many works attempt to learn them using a variety of objective,s including soft Q-learning (Mudgal et al., 2024), noise contrastive estimation (Lawson et al., 2022), and classification (Yang & Klein, 2021), we already have a reasonable twist function using pre-defined energy functions. We approximate the twist function by the relative likelihood of a sample under the target energy function and our model. To encourage samples that are lower energy during our autoregressive generation.

We use the intermediate signals of a peptide-based energy function (either Amber ff99SBildn or Amber 14, depending on the system) (Tan et al., 2025a). However, these peptide-based energy functions only function correctly on complete peptides. In the case of a partial peptide (for instance, the subset of atoms PPW in the peptide PPWRECNN), these atoms are not able to be processed by the energy function because they do not form a complete peptide, which has a C-terminus oxygen cap, often denoted as OXT, the terminal oxygen.

We are therefore not able to efficiently evaluate the partial energy of a subsequence like PPW. Instead, we generate one more atom, the nitrogen atom of the next residue, which can tell us a reasonable direction for the oxygen atom to go. Our procedure is then as follows:

1. Generate dimensions until we have a full residue plus the nitrogen atom of the next residue.
2. Find the direction of the nitrogen atom relative to the carbon atom its attached to.
3. Replace this nitrogen atom with an oxygen atom in the same direction off of the carbon atom, but at the optimal distance for carbon-oxygen bonds at 0.125 nanometers.
4. Evaluate the energy of this subset of the peptide using the relevant Amber energy function.

This procedure allows us to calculate the intermediate twist functions, which then allows for resampling to improve the distribution during autoregressive sampling.

We perform SMC over the entire batch of samples for the best performance. This is much larger than can fit on a single GPU. Therefore, we operate in batches. We perform batched generation with KV-caching per residue generated for minimal slow down. This has multiple advantages over other generation procedures. First, we only need to hold a single GPU batch worth of Keys and Values at once. Second, we only need to regenerate the cache every residue. This means for a large batch of B samples, we need to regenerate the cache at most $(L-1)B$ times, where L is the length of the peptide. This minimizes the additional overhead of SMC to a few model evaluations (less than a 10% overhead).

D.5.2 KV-CACHING

KV-caching is a technique used during inference that stores the attention keys and values from previous tokens during decoding to prevent recomputing them at every step (Pope et al., 2023). By caching these tensors, the model avoids redundant attention computations over the entire prefix, reducing per-token complexity and lowering latency at inference time. Consequently, we adopt it to reduce inference time cost when generating samples from the equilibrium distribution.

D.6 TABLE AND FIGURE SPECIFIC DETAILS

Table 1. ECNF++, RegFlow, SBG, FALCON-A, and FALCON results are taken from Tan et al. (2025a) and Rehman et al. (2025a). SBG here is SBG with Sequential Monte–Carlo sampling instead of SNIS, as this performed slightly better. All other methods utilize SNIS. ECNF++ dashes represent models that were not run on that system due to scaling concerns.

E RELATED WORK

Boltzmann Generators. The use of deep generative models in equilibrium sampling was popularized with the introduction of Boltzmann Generators (BGs) (Noé et al., 2019). Most subsequent work has focused on refining the NF architecture used in BGs to improve expressivity (Zhai et al., 2024; Draxler et al., 2024), stability (Schopmans & Friederich, 2025; von Klitzing et al., 2025), and generalization (Klein & Noe, 2024; Peng & Gao, 2025). CNFs offer superior expressivity and easy handling of symmetries (Köhler et al., 2020; Klein et al., 2023c), but incur a high computational cost for likelihood evaluation during inference, which is only partially ameliorated with approximate few-step models (Rehman et al., 2025a), architectural constraints (Gloy & Olsson, 2025), or learned energy functions (Aggarwal et al., 2025; Akhound-Sadegh et al., 2025).

Autoregressive Models on Continuous Spaces. Several works have applied transformer-based models in generating molecular structures, including latent space (Murtada et al., 2025), flow-based (Cheng et al., 2025; Team et al., 2025). While it is possible to design an autoregressive SE(3)-invariant architecture (Gebauer et al., 2019), these have not scaled as well as transformer-based methods. Mixture Density Networks (Bishop, 1994) have been explored for decades as a way to parametrize outputs over continuous spaces while providing exact densities (Razavi et al., 2020). These have been employed for generating sketches (Ha & Eck, 2017), images (Salimans et al., 2017), world models (Ha & Schmidhuber, 2018), and demand forecasting (Li et al., 2023). AR MDNs have also recently been reintroduced (Tschannen et al., 2024; Li et al., 2024; Billera et al., 2024); however, all these methods focus on proposal quality, rather than interrogating the use of likelihoods for BGs. Moreover, MDNs have historically been prone to dropping modes in their mixtures (Deng et al., 2022), leading to suboptimal performance (as also observed for molecules in Table 1).