PREFIX-CONDITIONED BOLTZMANN GENERATORS: TOWARDS GENERAL MOLECULAR SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable estimation of equilibrium molecular properties requires sampling the Boltzmann distribution, a central bottleneck in molecular simulation. Boltzmann Generators (BGs) use generative models with tractable likelihoods to enable importance reweighting to the Boltzmann distribution. Recent work has scaled BG training to larger and more diverse datasets, but existing approaches often specialize in particular molecular classes and single molecules. We introduce PREFIX-CONDITIONED BOLTZMANN GENERATORS (PBGS), a family of autoregressive Boltzmann generators conditioned on text and coordinate prefixes, without a specialized atom-wise molecular encoder. Across a series of models trained on *Many Peptides*, we identify scaling relationships linking sampling quality to model size, training tokens, training compute, atom count, and coordinate discretization resolution. Guided by these relationships, we scale PBGS up to 640M parameters, achieving state-of-the-art performance on unseen peptide systems containing up to 157 atoms. Models trained across diverse molecular classes and environmental conditions achieve comparable performance on our main benchmark at lower training cost than models trained only on peptides. These results suggest beneficial transfer across molecular classes. Beyond peptides, we demonstrate the versatility of PBGS through small-molecule sampling, solvation free energy estimation, and conditional microsolvation of systems containing over 280 atoms. Together, these results support PBGS as a promising approach to general molecular sampling.

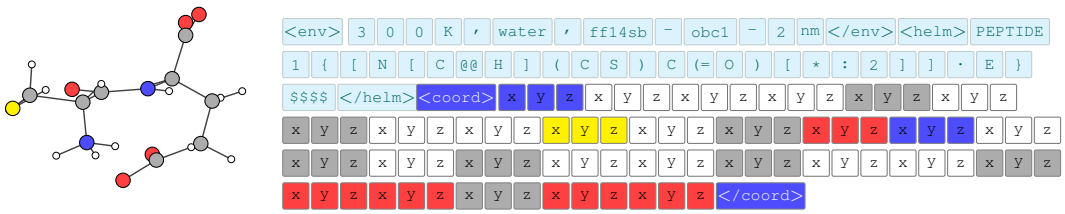


Figure 1: **Prefix-Conditioned Boltzmann Generators** represent molecular environments, chemical graphs, and atomic coordinates as discrete tokens, enabling flexible sampling across molecular systems and environments through prefix conditioning. Sampling quality exhibits empirical scaling relationships, while training across diverse molecular classes suggests positive transfer.

1 INTRODUCTION

In statistical mechanics, the properties of a material at equilibrium, for example a therapeutic small molecule, are defined by a probability density over its atomic coordinates induced by random collisions between the molecule and its constant-temperature environment, the Boltzmann distribution. Sampling these distributions across relevant states and environments supports the estimation of properties such as binding affinity and solubility. However, obtaining independent equilibrium samples remains a long-standing computational challenge. The dominant approach, molecular dynamics (MD), integrates atomic equations of motion and, with suitable temperature control, can sample the target equilibrium distribution. Yet metastability can make transitions between conformational states rare, producing strongly correlated trajectories and limiting effective sampling (Perez et al., 2025; Olsson, 2026).

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

Boltzmann generators (BGs) (Noé et al., 2019) offer an alternative by generating independent molecular configurations from a learned proposal distribution with tractable likelihoods, enabling importance reweighting to the target Boltzmann distribution. Recent work has extended this approach to transferable models that sample multiple molecular systems (Klein & Noe, 2024). Unlike protein structure prediction (Jumper et al., 2021), equilibrium sampling requires capturing the relative statistical weights of different configurations, and scaling transferable BGs to larger and more diverse systems remains challenging. Early BGs relied primarily on normalizing flows (Tabak & Turner, 2013; Dinh et al., 2017; Rezende & Mohamed, 2015; Chen et al., 2018), which model atomic coordinates in continuous space (Klein et al., 2023; Köhler et al., 2020; Tan et al., 2025a;b). More recently, however, Rehman et al. (2026) demonstrated state-of-the-art peptide sampling using autoregressive models of discretized atomic coordinates.

These developments motivate three questions: **(1)** How does sampling quality of autoregressive BGs scale with model size, training data, and training compute? **(2)** What conditioning representation supports training and sampling across diverse molecular systems and environments? **(3)** How do sampling quality and computational efficiency change as the number of atoms increases?

We introduce PREFIX-CONDITIONED BOLTZMANN GENERATORS (PBGS), a family of autoregressive BGs that represent molecular environments, chemical graphs, and atomic coordinates as discrete tokens. This representation enables conditioning through text and coordinate prefixes within a shared interface (Figure 1), without the separate atom-wise conditioning encoder used by Rehman et al. (2026). The autoregressive factorization preserves tractable proposal likelihoods, supporting importance reweighting.

Across models trained on *Many Peptides*, we characterize empirical scaling relationships linking sampling quality to model size, training compute, atom count, and coordinate discretization resolution. Guided by these relationships, we scale PBGS to 640M parameters, achieving state-of-the-art performance on unseen peptide systems containing up to 157 atoms. We also train PBG-Joint across molecular classes and environmental conditions. Despite using substantially fewer parameters and lower training compute, PBG-Joint matches or exceeds the peptide performance of our strongest specialist models (Table 2), suggesting beneficial transfer across training contexts.

Beyond improving sampling quality, we accelerate autoregressive generation using speculative decoding (Leviathan et al., 2023), achieving speedups of up to $2.14\times$ over standard autoregressive decoding, with particular benefits for larger systems. We demonstrate the flexibility of prefix conditioning through peptide and small-molecule sampling conditioned on amino acid sequences and SMILES, respectively, as well as conditional microsolvation around fixed solutes in systems containing more than 280 atoms. Finally, we apply the learned proposals to solvation free-energy estimation, extending their use from conformational sampling to thermodynamic property estimation.

Our main contributions are:

- We introduce PREFIX-CONDITIONED BOLTZMANN GENERATORS (PBGS), autoregressive molecular samplers that use a shared discrete-token representation for text and coordinate prefix conditioning while retaining tractable proposal likelihoods.
- We characterize empirical scaling relationships for sampling quality and use them to guide scaling to 640M parameters, achieving state-of-the-art performance on *Many Peptides*. We further accelerate generation using speculative decoding.
- We demonstrate joint training across molecular classes and environments, with evidence suggesting positive transfer, and apply PBGS to solvation free-energy estimation and conditional microsolvation of systems containing more than 280 atoms.

2 BACKGROUND

Equilibrium Sampling of Molecular Conformations Consider an atomistic system of interest with the descriptor c, n atoms, and a conformation $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \Omega \subseteq \mathbb{R}^{n\times 3}$, where $\Omega \subseteq \mathbb{R}^{n\times 3}$ is the configuration space. At equilibrium and fixed temperature, the probability density of a conformation $\mathbf{x}$ is given by the Boltzmann distribution

$$p(\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp[-\beta U(\mathbf{x})], \qquad Z_{\mathbf{x}} = \int_{\Omega} \exp[-\beta U(\mathbf{x})] \, \mathrm{d}\mathbf{x}, \tag{1}$$

where $U(\mathbf{x})$ is the potential energy, $\beta = (k_{\mathrm{B}}T)^{-1}$ is the inverse thermal energy, and $Z_{\mathbf{x}}$ is the configurational partition function, an integral over the configuration space. Given a test function $\phi(\mathbf{x})$, i.e. a physicochemical property of interest, we can estimate its value by forming the expectation. The required samples may be generated by simulating molecular dynamics, or by drawing independent samples from a learned proposal.

Boltzmann Generators. Boltzmann Generators are generative models which learn a parameterized proposal q_θ that approximates the Boltzmann distribution p and admit a tractable exact likelihood (Noé et al., 2019). Training typically minimizes $D_{\mathrm{KL}}(p\|q_\theta)$ on previously sampled equilibrium conformers, but can also be performed using potential energy supervision alone via minimization of one or more $D_{\mathrm{KL}}(q_\theta\|p)$ objectives (von Klitzing et al., 2026; Neal, 2001; Noé et al., 2019). A transferable Boltzmann Generator shares one set of parameters θ across many systems and realizes each proposal as a conditional $q_\theta(\mathbf{x} \mid c)$ (Klein & Noe, 2024). Once trained, independent samples $\mathbf{x}_i \sim q_\theta$ can then be drawn in parallel, and self-normalized importance sampling (SNIS) enables reweighting of samples to target the intended Boltzmann distribution.

$$\tilde{w}(\mathbf{x}) = \frac{e^{-\beta U(\mathbf{x})}}{q_\theta(\mathbf{x})}, \qquad \log \tilde{w}(\mathbf{x}) = -\beta U(\mathbf{x}) - \log q_\theta(\mathbf{x}), \qquad w_i = \frac{\tilde{w}(\mathbf{x}_i)}{\sum_{j=1}^{N} \tilde{w}(\mathbf{x}_j)}. \tag{2}$$

For a test function ϕ, the SNIS estimator is then $\mathbb{E}_p[\phi] \approx \sum_{i=1}^{N} w_i\, \phi(\mathbf{x}_i)$. The variance of this estimator is controlled by the overlap of q_θ and p.

A common approach to train Boltzmann Generators are Normalizing Flows (Tabak & Vanden-Eijnden, 2010; Dinh et al., 2017; Tan et al., 2025a). Flow-based proposals evaluate exact likelihoods through the change-of-variables formula, requiring invertible transformations with tractable Jacobian determinants. However, these requirements constrain architectural choices and can make distributions with well-separated modes difficult to represent (Cornish et al., 2020).

Autoregressive Boltzmann Generators. Recently Rehman et al. (2026) introduced Autoregressive Boltzmann Generators (ARBG), an alternative approach to Boltzmann Generators based on the autoregressive prediction of discretized atom coordinates. We consider ARBGs to be part of a family of models we will refer to as **Discretized Boltzmann Generators**, composed of five largely independent components: (1) A surjective, and non-injective *Quantizer* κ_Δ, which maps a continuous conformation $\mathbf{x}$ to a voxel code $\mathbf{k} \in \mathcal{A}_b^{n\times 3}$; (2) a bijective *Tokenizer*, which serializes that code into a sequence of lenght m categorical symbols $\mathbf{z} = \tau(\mathbf{k})$; (3) a deterministic *Decoding Order* $\sigma \in S_m$, which determines the order in which those m symbols are to be generated; (4) a *Discrete Generative Model* $q_{\theta,\Delta}$ that learns the proposal distribution over token sequences; and (5) a *Dequantizer* kernel $\mathcal{R}_\Delta$, which assigns a voxel code $\mathbf{k}$ to a conditional distribution over continuous coordinates $\mathbf{x}$. We discuss each component in further detail in Appendix B.

Briefly, ARBGs rely on a quantizer to partition a cube of side L, in which molecules are sampled, into b equal-width cells of size $\Delta = L/b$ along each axis. The scalar map κ_Δ sends each axis coordinate to the index of the unique cell that contains it, resulting in a voxel code for the entire conformation $\mathbf{k} = (k_{id}) \in \mathcal{A}_b^{n\times 3}$. Physically, each voxel code defines a macrostate of the discretized distribution p_Δ which represents the actual learning target of the generative model. Intuitively, as $\Delta \to 0$, the voxel probability per unit volume approaches the continuous Boltzmann density: $p_\Delta(\kappa_\Delta(\mathbf{x}))/\Delta^{3n} \to p(\mathbf{x})$.

The learned proposal is a categorical distribution $q_{\theta,\Delta}(\mathbf{z} \mid c, \sigma)$ on the token sequence $\mathbf{z} = \tau(\mathbf{k})$, conditioned on the system descriptor c and a fixed decoding order σ. In the case of ARBGs this decoding order is derived directly from a predefined atom order π by emitting each atom's 3 discretized spatial coordinates consecutively, yielding $m = 3n$ coordinate tokens. Flattening along σ gives

$$q_{\theta,\Delta}(\mathbf{z} \mid c, \sigma) = \prod_{j=1}^{m} q_{\theta,\Delta}\big(z_{\sigma(j)} \mid z_{\sigma(<j)}, c\big). \tag{3}$$

Once a discretized proposal is sampled and mapped into a voxel code $\mathbf{k}$, a reconstruction kernel $\mathcal{R}_\Delta$ (e.g. a *Uniform Reconstruction Kernel*) returns a continuous conformation $\mathbf{x}$ within the encoded volume of space to evaluate test functions of interest $\phi(\mathbf{x})$ and to estimate importance weights using the potential energy $U(\mathbf{x})$.

Efficient training methods and widespread adoption make autoregressive models a promising basis for scaling Boltzmann generators towards larger and more diverse molecular systems. This requires large, chemically diverse training corpora, but current ARBG conditioning schemes to construct the conditioning information c limit this flexibility.

In Robin, the transferable ARBG in Rehman et al. (2026), an encoder η_θ maps hand-designed features $F(g, \pi)$ of a chemical graph $G = (V, E)$ with $n = |V|$ atoms to atom embeddings, $C_g = \eta_\theta(F(G)) \in \mathbb{R}^{n \times d}$, whereby the ordering of atoms π itself is given by a graph based heuristic $\chi_{intra}(G)$ (Case et al., 2023). These embeddings are added to the corresponding coordinate-token representations and used as cross-attention keys and values before the causal decoder.

This current design imposes three restrictions. First, one embedding per atom makes conditioning length grow with system size, even when a compact descriptor is available, such as a protein sequence or SMILES (Weininger, 1988). Second, the complete atom set and its ordering must be specified before sampling, preventing, for example, generation of a variable number of explicit solvent molecules. Third, extension beyond pre-specified chemistries requires changes to the hand-designed feature schema and its encoder.

3 Prefix-Conditioned Boltzmann Generators

Conditioning and prompt construction. We encode the molecular system and its environment as a tokenized text prefix c. Extending the graph conditioning used in ARBGs, this prefix specifies the molecular graph, solvent environment, potential energy model, and temperature. Environment and molecular specifications are separated by special delimiter tokens, with a final separator marking the start of the coordinate sequence (Table 1).

We represent molecular graphs using the Hierarchical Editing Language for Macromolecules (HELM) (Zhang et al., 2012). HELM supports linear, branched, and cyclic biopolymers, including chemical modifications. Connectivity annotations specify links between monomers, while inline SMILES (Weininger, 1988) with labeled attachment points describe residues outside the standard monomer vocabulary. Small molecules are represented using SMILES within HELM chemical components. This provides a shared textual interface for biopolymers, modified residues, and small molecules, without a separate atom-wise conditioning encoder.

Table 1: **Example System Prompts.** Shown are representative molecules from public and newly generated datasets and their system prompts. `[BOS]` and `[SEP]` abbreviate the special delimiter tokens between environment, HELM, and coordinate blocks.

System	Source	Prompt
CGSWHKQR	Many Peptides	`[BOS] 310K,water,ff14sb-obc1-2nm [SEP] PEPTIDE1{C.G.S.W.H.K.Q.R}$$$$ [SEP]`
DEMT	Vacuum Peptides	`[BOS] 300K,vacuum,ff14sb [SEP] PEPTIDE1{D.E.M.T}$$$$ [SEP]`
HYGW + 128 waters	Microsolvated Peptides	`[BOS] 300K,water,ff14sb-tip3p-rf2nm [SEP] PEPTIDE1{H.Y.G.W}$$$$ [SEP]`
cyclo-AYTW	Many Cycles	`[BOS] 300K,water,ff14sb-obc1-1nm [SEP] PEPTIDE1{A.Y.T.W}$PEPTIDE1,PEPTIDE1,1:R1-4:R2$$$ [SEP]`
Molecule 00621	MDQM9-nc	`[BOS] 300K,vacuum,gaff2-am1bcc [SEP] CHEM1{[C[C@@H](CO)CC#N]}$$$$ [SEP]`
2,2,4-Trimethylpentane	FreeSolv (vacuum)	`[BOS] 300K,vacuum,gaff2-am1bcc [SEP] CHEM1{[CC(C)CC(C)(C)C]}$$$$ [SEP]`
Bromotrifluoro-methane	FreeSolv (implicit water)	`[BOS] 300K,water,gaff2-am1bcc-obc1-2nm [SEP] CHEM1{[FC(F)(F)Br]}$$$$ [SEP]`
Phenyl formate + 128 waters	FreeSolv (microsolvated)	`[BOS] 300K,water,gaff2-am1bcc-tip3p-rf2nm [SEP] CHEM1{[O=COc1ccccc1]}$$$$ [SEP]`

Variable length coordinate generation. After the final `[SEP]` token, the model predicts discretized Cartesian coordinates autoregressively, with three coordinate tokens per atom. Termination differs between nonsolvated and explicitly solvated systems. For nonsolvated systems, training sequences end with an `[EOS]` token. If $z_{1:m}$ is a complete sequence of $m = 3n$ coordinate tokens, its probability is

$$q_\theta(z_{1:m}, \texttt{[EOS]} \mid c) = \left[\prod_{t=1}^{m} q_\theta(z_t \mid c, z_{<t})\right] q_\theta(\texttt{[EOS]} \mid c, z_{1:m}).$$

For explicitly solvated systems, however, training sequences contain no `[EOS]` token. At inference, the number of atoms to generate is specified externally, and sampling stops after the corresponding number of coordinate tokens is reached, allowing, for example, for sampling of a variable number of solvent molecules.

Supramolecular atom ordering. Autoregressive Boltzmann generators generate coordinate tokens according to an atom order π. We obtain the order within each molecule from an existing graph-based heuristic χ_{intra} (RDKit). To generalize the canonicalization of atom orders to supramolecular systems, such as explicitly solvated small molecules, $G = G_0 \sqcup G_1 \sqcup \cdots \sqcup G_K$, comprising a reference solute G_0 and K identical solvent molecules, we rely on the indistinguishability of solvent molecules and define $\chi_{\text{inter}}(G, \mathbf{x}) = \rho$ as the permutation that orders, for example, waters by increasing distance from their oxygen atom to the nearest solute atom: $d_{\rho(1)} \leq \cdots \leq d_{\rho(K)}$ with $d_j = \min_{i \in V(G_0)} \|\mathbf{x}_{o_j} - \mathbf{x}_i\|_2$, where o_j denotes the oxygen atom of water j. The complete atom order places the solute first, followed by the ordered solvent molecules: $\pi = \chi(G, \mathbf{x}) = \chi_{\text{intra}}(G_0) \,\|\, \chi_{\text{intra}}(G_{\rho(1)}) \,\|\, \cdots \,\|\, \chi_{\text{intra}}(G_{\rho(K)})$. This coordinate-dependent rule selects an ordering of indistinguishable solvent molecules in the training conformations. We found this greatly reduced collisions during sampling as compared to random ordering during preliminary experimentation with discrete diffusion (Shi et al., 2024; Sahoo et al., 2024). At inference, we generate waters autoregressively without enforcing increasing solute–water distance. This leads to a framework where during training we canonicalize atom and molecular order, and softly encourage global rotation and translation equivariance through data augmentation.

4 Experiments

4.1 Increasing Sampling Quality via Model Scaling

Baselines. We reproduce the transferable Autoregressive Boltzmann Generator ROBIN (Rehman et al., 2026) as the closest method to our work. In addition, we also compare with published results using autoregressive normalizing flows (Prose (Tan et al., 2025a) and TarFlow (Zhai et al., 2025)).

Metrics. Following prior work, we evaluate sampling quality using the stand-alone relative Effective Sample Size (rESS) (Kish, 1957), as well as three Wasserstein metrics describing similarities between sampled and reference conformations: (a) The 2-Wasserstein distance of the dimensionless reduced potential energy ($\mathcal{E}$-W_2), where $\mathcal{E}(\mathbf{x}) \equiv \beta U(\mathbf{x})$, (b) the periodicity-aware 2-Wasserstein distance ($\mathbb{T}$-$\mathcal{W}_2$) of the backbone torsion angles, and (c) the 2-Wasserstein distance along the first two components (TICA-$\mathcal{W}_2$) of a time-lagged independent component analysis (TICA). In contrast to recent work (Rehman et al., 2026), we re-emphasize the rESS as a metric, given its ability to provide a necessary but not sufficient indicator of sampling quality in contexts in which no reference samples are available. To increase the computational throughput and generality towards non-peptide molecules, we also introduce a sliced 2-Wasserstein distance of *all* torsion angles of a given molecule ($\mathbb{T}$g-$\mathcal{SW}_2^*$) (Appendix F). For scaling model fitting we evaluate the previously described metrics without the common 0.2% clipping of importance weights. We mark these unclipped metrics with a superscript, yielding rESS*, $\mathcal{E}$-W_2^* and $\mathbb{T}$-$\mathcal{W}_2^*$.

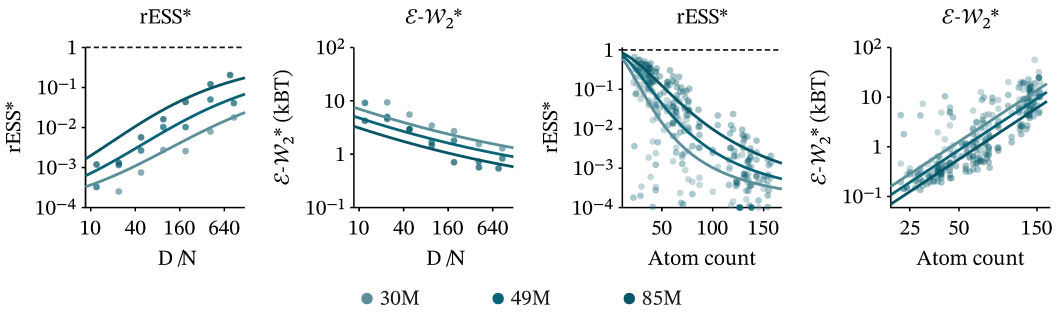


Figure 2: **Sampling Quality Scales with Compute and Decreases with System Size.** Left: Median rESS* and $\mathcal{E}$-W_2^* on *Many Peptides* test systems versus training budget in loss tokens per non-embedding parameter (D/N). Right: per-system rESS* and $\mathcal{E}$-W_2^* versus atom count for representative models at $D/N = 192$.

Sampling Quality Scales with Compute and Decreases with System Size. In order to assess the ability for Prefix-Conditioned Boltzmann Generators to scale towards higher sampling quality and large atomistic system sizes, we investigated whether sampling performance could be anticipated

with a scaling model covering model size, compute, system atom count and discretization resolution. To this end, we trained a series of 130 autoregressive PBGs with 8M-85M non-embedding parameters (N) on the *Many Peptides* and assessed training compute budgets from $D/N = 12$ to $D/N = 1760$ loss tokens per non-embedding parameter, where D counts loss tokens, using a Warmup-Stable-Decay schedule (Hägele et al., 2024) approach.

Table 2: **Evaluation results on *Many Peptides*.** Shows means across the test systems listed in the headers, with 10^4 samples each. The first section of values are literature derived, following sections are measured. Best values in **bold**, with second-best underlined. D/N next to a model name gives the number of loss tokens per non-embedding parameter for that model. T denotes the logit temperature.

	2AA (30 systems)				4AA (30 systems)				8AA (29 systems)¶			
Model	rESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	T-$\mathcal{W}_2$ ↓	TICA-$\mathcal{W}_2$ ↓	rESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	T-$\mathcal{W}_2$ ↓	TICA-$\mathcal{W}_2$ ↓	rESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	T-$\mathcal{W}_2$ ↓	TICA-$\mathcal{W}_2$ ↓
TarFlow	0.134	0.452	0.193	—	0.045	1.260	0.924	0.492	0.008	11.272	2.644	1.059
Prose	0.191	0.371	0.210	—	0.071	0.932	0.752	**0.367**	0.011	9.817	2.451	1.026
ROBIN SMC	—	—	—	—	—	1.079	0.874	0.463	—	4.238	2.300	0.996
ROBIN ($T = 0.9$)	—	—	—	—	0.031	1.168	0.886	0.471	0.013	4.295	2.314	0.957
ROBIN ($T = 1.0$)	0.035	0.702	0.255	0.062	0.031	1.077	0.808	0.447	0.013	5.933	2.253	0.973
PBG 30M ($D/N = 1760$)	0.452	0.283	0.120	0.043	0.178	0.774	0.621	0.408	0.020	6.788	2.172	1.024
PBG 49M ($D/N = 416$)	0.418	0.272	0.128	0.043	0.162	0.764	0.647	0.430	0.019	6.705	2.167	1.044
PBG 85M ($D/N = 768$)	0.647	0.274	0.105	0.041	0.344	0.530	0.569	0.399	0.047	4.142	1.955	0.987
PBG 160M ($D/N = 352$)	0.550	0.286	0.115	0.043	0.281	0.529	0.572	0.403	0.043	3.716	1.941	0.998
PBG 320M ($D/N = 86$)	0.565	0.273	0.111	0.045	0.283	0.509	0.552	0.368	0.041	4.023	1.968	0.941
PBG 640M ($D/N = 48$)	**0.729**	**0.251**	0.103	0.041	**0.462**	**0.402**	**0.536**	0.415	**0.088**	**2.632**	**1.824**	0.959
PBG-Joint 160M ($D/N = 198$)	0.436	0.286	**0.101**	**0.039**	0.198	0.596	0.587	0.377	0.030	3.937	1.975	**0.940**

Once trained, the rESS^*, $\mathcal{E}$-W_2^* and $\mathbb{T}$g-$\mathcal{SW}_2^*$ evaluation metrics on the *Many Peptides* test systems and the terminal training loss were modeled using a shared factor scaling model based on (Rosenfeld, 2021; Kaplan et al., 2020) via Bayesian Inference (Figure 2) (Phan et al., 2019; Ruan et al., 2024). Briefly, a latent factor based on the common form $G = A\,(N/10^9)^{-\alpha} + B\,(D/10^9)^{-\beta}$ was estimated to model the training loss and three evaluation metrics jointly. We tied exponents ($\alpha = \beta$) given the limited set of observations and previous results of approximately proportional compute-optimal scaling of model size and training data in language modeling (Hoffmann et al., 2022). Each metric m, including the loss, was fitted with an *adapter* that maps G and the atom count n to the observed metric value through a shared functional form:

$$Y_m = a_m + p_m \log\left(\frac{n}{n_0}\right) + k_m \log G + \delta_m[b], \qquad \delta_m[4096] = 0. \tag{4}$$

where a_m is the intercept, p_m the atom-scaling coefficient, k_m the G-sensitivity, and $\delta_m[b]$ is a per bin-size offset. For the NLL, we set $p_{NLL} = 0$ and thereby reduce the adapter to G plus a bin offset $\delta_m[b]$ and an irreducible error a_{NLL}, in line with previous scaling models (Rosenfeld, 2021). For rESS^*, the output $-\log(\mathrm{rESS})$ is additionally bounded by the finite sample count via a sigmoid: $-\log(\mathrm{rESS}) = C \cdot \mathrm{sigmoid}(Y_{\mathrm{rESS}})$ (Appendix G).

We estimated atom-count coefficients $p_E = 2.17$ for $\log(\mathcal{E}\text{-}W_2^*)$ and $p_R = 1.79$ within the sigmoid model of $-\log(\mathrm{rESS}^*)$. At fixed training resources and discretization, energetic error grows approximately quadratically with atom count. Similarly, outside of saturated rESS^* regimes, we observe a near-quadratic growth of $-\log(\mathrm{rESS}^*)$, implying a rapid loss of effective samples as systems grow. In contrast, $\log(\mathbb{T}\text{g-}\mathcal{SW}_2^*)$, $p_m = 1.01 \approx 1$ showed linear scaling with system size (Figure 10).

To understand the degree to which the discretization of atomic coordinates alone leads to an increase in energetic error, we analyzed the expected energy bias caused by midpoint reconstruction of quantized conformers (Appendix D.2). In the fine-bin regime, midpoint reconstruction induces an expected energy bias proportional to the atom count $\mathbb{E}[\Delta U^{\mathrm{mid}}] \propto n\Delta^2$ which is thereby lower-bounding the unweighted roundtrip energy Wasserstein error, $\mathcal{E}\text{-}W_2 \geq \beta|\mathbb{E}[\Delta U^{\mathrm{mid}}]|$. Experiments with quantization–reconstruction *roundtrips* of equilibrium *Many Peptides* conformers support this result, suggesting that the steep decay of sampling quality metrics with atom count is in part due to discretization-induced error.

Overall, the scaling model implies a compute-optimal ratio of $D^*/N = 39.9$ (95% CI 35.1–44.8) (Table 5). Based on the observed scaling model, we proceeded to train larger Prefix-Conditioned Boltzmann Generators, spanning 160M, 320M, and 640M parameters on the *Many Peptides* dataset. In line with the model, we observed strong results on the *Many Peptides* test set (Table 2 and Appendix Figure 12).

Discretization Resolution Has a Model-Dependent Optimum. Besides varying model parameters and training data, we also investigated the effect of varying the number of discretization bins b, and thus the spatial resolution Δ with which conformers are sampled. In contrast with previous reports (Rehman et al., 2026), we observed an optimum of discretization depth with both low ($b \leq 2^{10} = 1024$ bins, $\Delta \geq 3.2$ pm) and high ($b \geq 2^{13} = 8192$ bins, $\Delta \leq 0.4$ pm) spatial resolution leading to reduced sampling quality. These observations are in line with a two-regime model, where both too few discretization bins (quantization-limited regime) and too many discretization bins (vocabulary-limited regime) lead to reduced sampling quality.

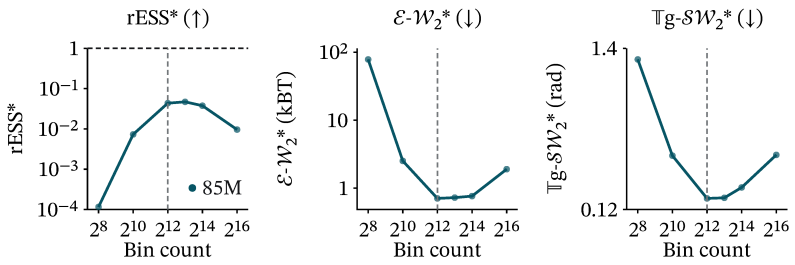


Figure 3: **Discretization Resolution Effect on Evaluation Metrics.** Sampling Quality versus discretization bin count and resolution. Shown are results for PBGs trained on a resolution grid of 2^8–2^{16} bins. Points are observed per-checkpoint medians of *Many Peptides* test systems at $D/N = 192$. Dashed line represents 2^{12} bin reference.

4.2 Increasing Sampling Rate via Speculative Decoding

Speculative-Decoding Increases Sampling Efficiency. In practice, the utility of a sampling method, its sampling efficiency, is a function of both the sampling quality and the sampling rate on available hardware. Previously, autoregressive Boltzmann generators with discretized coordinates were shown to produce samples at a lower rate, than existing flow based alternatives (Rehman et al., 2026).

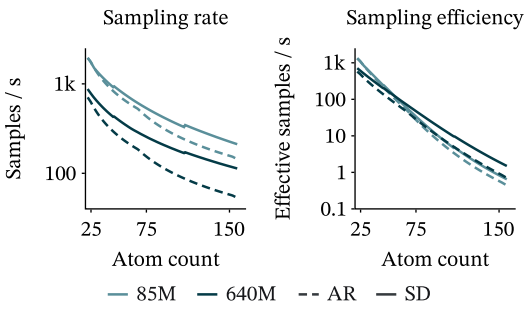


Figure 4: **Sampling Rate and Efficiency.** Sampling rates (left) and rESS*-weighted sampling efficiency (right) for 85M and 640M PBGs. Dashed: autoregression (AR); solid: best AR or speculative decoding (SD) with a 15M drafter.

To improve the sampling rate of autoregressive Boltzmann generators while maintaining exact base-model likelihoods, we performed speculative decoding with an existing 15M PBG draft model (Leviathan et al., 2023). We sampled peptides at varying length and fixed batch size of 512 on a H100 NVL from representative, state-of-the-art PBGs. The 85M and 640M models achieved mean observed speedups relative to standard autoregressive sampling of $1.27\times$ and $1.55\times$, with maxima of $1.64\times$ and $2.14\times$, respectively (Figure 4 and 17).

Despite these improvements in sampling rate alone, we generally notice a steep decline in overall sampling efficiency of Boltzmann generators as atom count increases: As both sampling rate and sampling quality decline with atom count independently, Boltzmann generator's sampling efficiency drops by a factor of ca. 10^3 as system size increases by a factor of ca. 2^3. These results highlight the need to improve both sampling quality and sampling rate as Boltzmann generators are extended to larger systems.

4.3 Towards General Molecular Sampling

Prefix-Conditioned Boltzmann Generators sample across chemistries and solvent environments. In order to estimate the properties of a material, for example a therapeutic small molecule, its Boltzmann distribution needs to be sampled across different environments. For example, the estimation of solvation free energies requires conformer samples in vacuum and water.

To enable solvation free energy estimation using Boltzmann Generators, we aimed to extend their transferability towards small molecules and across solvent environments. We trained a Prefix-

Conditioned Boltzmann Generator on a collection of public and newly generated datasets including *Many Peptides*, *MDQM9-nc* (Viguera Diez & Olsson, 2024) (non-cyclic small molecules in vacuum), *Vacuum Peptides* (peptides from *Many Peptides* in vacuum) and *FreeSolv* (small molecules sourced from Mobley & Guthrie (2014) in vacuum and implicit solvent). The resulting model was termed PBG-Joint (160M, $D/N = 198$).

PBG-Joint achieved strong performance on *Many Peptides*, with Wasserstein errors comparable to the 320M peptide-only model at approximately half the non-embedding parameter count and estimated training compute, although its rESS remained lower (Table 2). Sampling quality across small molecules varied with chemical complexity and was generally equivalent to dedicated, fine-tuned models (Appendix Table 10 and Figure 18).

Boltzman Generators enable solvation free-energy estimation through the Zero Free Energy Proposal cycle (Ding & Zhang, 2021) (Appendix A). We assessed the model's ability to generate solvation free energy estimates for previously unseen test peptides and FreeSolv small molecules (Figure 5 and 20). BAR estimates reached overlaps ≥ 0.03 in both states for 42 of 90 peptide systems (Klimovich et al., 2015) and 21 of 94 FreeSolv test systems. In this subset, estimates correlated with Kyte–Doolittle GRAVY hydropathy scores (Kyte & Doolittle, 1982) ($r = +0.41$) for short peptides and experimental solvation free energy measurements ($r = +0.51$) for small molecules within the *FreeSolv* dataset. Of note, also direct Zwanzig estimation using PBG-Joint, which does not depend on reference MD samples, yielded comparable estimates for the FreeSolv dataset (Figure 22).

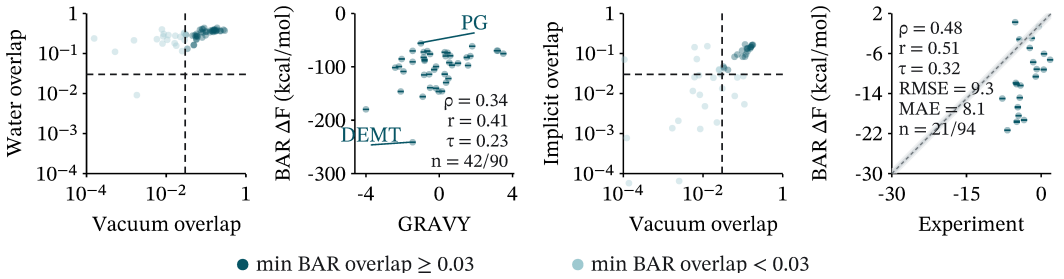


Figure 5: **Solvation Free Energy Estimation using Prefix-Conditioned Boltzmann Generators.** Peptides (left pair) and FreeSolv (right pair): BAR overlaps and free-energy estimates versus GRAVY score or experimental measurements. Correlations require both BAR overlaps ≥ 0.03. Error bars show propagated state uncertainty.

Compared to implicit solvation, solvation free energy estimation approaches that sample explicit solvent-solute interactions are considered the state-of-the-art (Alibay et al., 2025). We therefore prepared an additional *Microsolvated Peptides* and *FreeSolv* microsolvation dataset, containing conformers of solutes and their 128 nearest water molecules cropped from short explicit solvent MD trajectories run in 1nm solvent-padded boxes with periodic boundary conditions. To calculate potential energies for such a cropped system during inference time we resorted to a reaction field approach. Notably, this introduced a mismatch between the underlying data generating potential and the inference time potential, leading to collapsed effective sample sizes of trained models. Given these constraints, we assessed a dedicated PBG model's ability to sample favorable intermolecular interactions with a microsolvation task, in which a variable number of 32–80 water molecules were placed around fixed solute conformer from explicit-solvent MD. For evaluation, we measured total potential energy and wall-clock time. We compared our approach against existing water placement tools, ORCA's Solvator (Neese, 2012; FACCTs GmbH, 2024) and Amber's tleap (Case et al., 2023), and observed lower minimum and median energies, as well as faster microsolvation for octapeptides relative to these baselines (Table 3 and Figure 6).

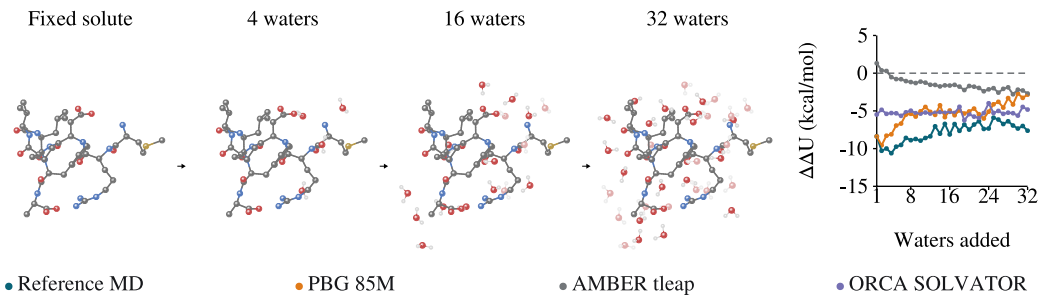


Figure 6: **Autoregressive microsolvation of an octapeptide.** Left: Microsolvation of 4, 16, and 32 waters around a fixed MRDPVLFA conformer in the original generation order by a PBG. Pale red color marks earlier waters. Right: Median energy change per added water for all 14 net-neutral octapeptide systems. Curves show adjacent differences of cohort-median $U(k) - U(0)$ in kcal/mol.

Table 3: **Microsolvation Task.** For each test net-neutral peptide, 512 distinct 300 K solute conformers are held fixed and solvated by different methods. Columns show the mean across systems.

	2AA + 80 H_2O (19 net neutral systems)			4AA + 64 H_2O (13 net neutral systems)			8AA + 32 H_2O (14 net neutral systems)		
Method	Avg. min ↓	Avg. median ↓	Avg. samples/s ↑	Avg. min ↓	Avg. median ↓	Avg. samples/s ↑	Avg. min ↓	Avg. median ↓	Avg. samples/s ↑
Reference MD	−2715	−2329	—	−2053	−1656	—	−1033	−548.1	—
AMBER tleap	**−1719**	**−1350**	10.79	−841.0	−448.0	9.228	−226.1	341.8	7.496
ORCA SOLVATOR	−1538	−915.8	0.4477	−1233	**−716.1**	0.4913	−663.8	−170.5	0.6494
PBG 85M	−1714	−809.2	**14.14**	**−1245**	−540.4	**15.03**	**−673.9**	**−206.1**	**16.59**

5 Conclusion

In this work we define Prefix-Conditioned Boltzmann Generators and demonstrate their ability to learn transferable proposals across peptides, small molecules, solvent environments and towards microsolvated states. These proposals enable initial solvation free energy estimations with just one generative model. Within the study's parameter range and data, Prefix-conditioned Boltzmann generators adhere to scaling laws which indicate a more data-intensive compute-optimal training at $D^*/N = 39.9$, compared to large language models (Kaplan et al., 2020), and an overall steep scaling of training compute requirement to maintain performance metrics, such as energetic error, as system size increases $C^*_{\mathcal{E}\text{-}W_2} \approx n^{5.56}$. Despite methods such as speculative decoding, both sampling quality *and* sampling rate decay with atom count, leading to strong reductions in overall sampling efficiency as BGs are applied to larger systems. Identifying sampling methods which can maintain the statistical properties of Boltzmann Generators while scaling more favorably remains an open research problem.

Limitations. The presented study is constrained by a range of experimental limitations, which include the scaling model and free energy estimation accuracy. The presented scaling model has a limited set of three extrapolating training runs which is not sufficient to draw generalized conclusions in the form of a *scaling law* for autoregressive Boltzmann generators. Additionally, a similar relationship has not yet been described for an independent, sufficiently sized dataset, such as *MDQM9-nc* (Viguera Diez & Olsson, 2024).

The sampling quality of small molecule and vacuum conformational ensembles is reduced compared to the *Many Peptides* reference results, which is also reflected in solvation free energy estimates. We attribute this to a set of factors, which include **(1)** limited size of newly sampled datasets compared to established baselines (Table 13), **(2)** reduced electrostatic screening in vacuum environments which might increase the energetic consequences of conformational errors in samples, and **(3)** challenges in prompting PBGs with small molecule sequence representations such as SMILES without pre-training on large, diverse small molecule data to learn this representation itself.

We anticipate the development of future methods which can improve sampling quality in such challenging, yet highly relevant, contexts, e.g. via iterative updating of decoded atom coordinates.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing code to run the experiments and analyses, searching the literature for related work, formulating mathematical claims, and text editing to improve clarity. We have not used generative AI tools for generating synthetic data sets. We take responsibility for the final content of this work, including text, claims and artifacts produced.

REPRODUCIBILITY STATEMENT

We plan to release the code required to reproduce the experimental results on GitHub. Model architectures, training settings and evaluation procedures are described in the appendix. We also plan to release the trained model checkpoints and the datasets created within the context of this project on HuggingFace.

REFERENCES

Irfan Alibay, Richard J. Gowers, David W. H. Swenson, Michael M. Henry, Benjamin Ries, Hannah M. Baumann, James R. B. Eastwood, Joshua A. Mitchell, David Dotson, Joshua T. Horton, Matthew Thompson, and Alyssa Travitz. The open free energy library. Version 1.7.0, 2025. URL https://zenodo.org/records/17410304. (← p. 8)

Emmanuel Bengio, Sanjeev Raja, Yui Tik Pang, Kerstin Klaeser, Cristian Gabellini, Nikhil Shenoy, Francesco Di Giovanni, and Prudencio Tossou. AquaGen: Scaling generative models to molecular dynamics precision on thousands of atoms. *arXiv preprint arXiv:2607.03513*, 2026. URL https://arxiv.org/abs/2607.03513. (← p. 18)

Charles H. Bennett. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976. doi: 10.1016/0021-9991(76)90078-4. (← p. 15)

David A. Case, Hasan Metin Aktulga, Kellon Belfon, David S. Cerutti, G. Andrés Cisneros, Vinícius Wilian D. Cruzeiro, Negin Forouzesh, Timothy J. Giese, Andreas W. Götz, Holger Gohlke, Saeed Izadi, Koushik Kasavajhala, Mehmet C. Kaymak, Edward King, Tom Kurtzman, Tai-Sung Lee, Pengfei Li, Jian Liu, Tyler Luchko, Ray Luo, Madushanka Manathunga, Matias R. Machado, Hai Minh Nguyen, Kurt A. O'Hearn, Alexey V. Onufriev, Feng Pan, Sergio Pantano, Ruxi Qi, Ali Rahnamoun, Ali Risheh, Stephan Schott-Verdugo, Akhil Shajan, Jason Swails, Junmei Wang, Haixin Wei, Xiongwu Wu, Yongxian Wu, Shi Zhang, Shiji Zhao, Qiang Zhu, Thomas E. Cheatham, III, Daniel R. Roe, Adrian Roitberg, Carlos Simmerling, Darrin M. York, Maria C. Nagan, and Kenneth M. Merz, Jr. Ambertools. *Journal of Chemical Information and Modeling*, 63(20):6183–6191, 10 2023. ISSN 1549-9596. doi: 10.1021/acs.jcim.3c01153. URL https://doi.org/10.1021/acs.jcim.3c01153. (← pp. 4 and 8)

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html. (← p. 2)

Austin H. Cheng, Chong Sun, and Alán Aspuru-Guzik. Scalable autoregressive 3D molecule generation. *arXiv preprint arXiv:2505.13791*, 2025. doi: 10.48550/arXiv.2505.13791. URL https://arxiv.org/abs/2505.13791. (← pp. 16, 16, and 18)

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In Hal Daumé, III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2133–2143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cornish20a.html. (← p. 3)

Alexandre G. de Brevern, Catherine Etchebest, and Serge Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins: Structure, Function, and Bioinformatics*, 41(3):271–287, 2000. doi: 10.1002/1097-0134(20001115)41:3<271::AID-PROT10>3.0.CO;2-Z. (← p. 18)

Xinqiang Ding and Bin Zhang. Deepbar: A fast and exact method for binding free energy computation. *The Journal of Physical Chemistry Letters*, 12(10):2509–2515, 03 2021. ISSN 1948-7185. doi: 10.1021/acs.jpclett.1c00189. URL https://doi.org/10.1021/acs.jpclett.1c00189. (← pp. 8, 15, and 18)

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx. (← pp. 2 and 3)

FACCTs GmbH. ORCA SOLVATOR: Automatic placement of explicit solvent molecules. ORCA 6.0 Manual, Section 6.10, 2024. URL https://www.faccts.de/docs/orca/6.0/manual/contents/typical/solvator.html. (← p. 8)

Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7566–7578, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a4d8e2a7e0d0c102339f97716d2fdfb6-Abstract.html. (← pp. 16 and 18)

Nathan Godey and Yoav Artzi. Lost in backpropagation: The lm head is a gradient bottleneck, 2026. URL https://arxiv.org/abs/2603.10145. (← p. 21)

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76232–76264. Curran Associates, Inc., 2024. doi: 10.52202/079017-2427. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8b970e15a89bf5d12542810df8eae8fc-Paper-Conference.pdf. (← pp. 6 and 33)

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. (← p. 18)

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. doi: 10.52202/068431-2176. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html. (← p. 6)

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/. (← p. 33)

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2. (← p. 2)

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. (← pp. 6, 9, and 19)

Leslie Kish. Confidence intervals for clustered samples. *American Sociological Review*, 22(2):154–165, 1957. doi: 10.2307/2088852. URL https://doi.org/10.2307/2088852. (← p. 5)

Leon Klein and Frank Noe. Transferable boltzmann generators. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=AYq6GxxrrY. (← pp. 2 and 3)

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59886–59910, 2023. doi: 10.52202/075280-2617. (← p. 2)

Pavel V. Klimovich, Michael R. Shirts, and David L. Mobley. Guidelines for the analysis of free energy calculations. *Journal of computer-aided molecular design*, 29:397–411, 2015. doi: 10.1007/s10822-015-9840-9. URL https://doi.org/10.1007/s10822-015-9840-9. (← pp. 8 and 32)

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370, 2020. URL https://proceedings.mlr.press/v119/kohler20a.html. (← p. 2)

Rachel Kolodny, Patrice Koehl, Leonidas Guibas, and Michael Levitt. Small libraries of protein fragments model native protein structures accurately. *Journal of Molecular Biology*, 323(2):297–307, 2002. doi: 10.1016/S0022-2836(02)00942-7. (← p. 18)

Jack Kyte and Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982. doi: 10.1016/0022-2836(82)90515-0. (← p. 8)

Kit Fun Lau and Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989. doi: 10.1021/ma00200a030. (← p. 18)

Tony Lelièvre, Mathias Rousset, and Gabriel Stoltz. *Free Energy Computations: A Mathematical Perspective*. Imperial College Press, 2010. doi: 10.1142/p579. URL https://inria.hal.science/hal-00781565. (← p. 15)

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html. (← pp. 2, 7, and 26)

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Adam E. Foster, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Roberto Sordillo, Ryota Tomioka, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761):eadv9817, 2025. doi: 10.1126/science.adv9817. (← p. 18)

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982. (← p. 33)

Shuqi Lu, Haowei Lin, Lin Yao, Zhifeng Gao, Xiaohong Ji, Yitao Liang, Weinan E, Linfeng Zhang, and Guolin Ke. Unified cross-scale 3D generation and understanding via autoregressive modeling. *arXiv preprint arXiv:2503.16278*, 2025. URL https://arxiv.org/abs/2503.16278. (← p. 18)

Xiao-Li Meng and Wing Hung Wong. Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 6(4):831–860, 1996. (← p. 15)

David L. Mobley and J. Peter Guthrie. FreeSolv: A database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design*, 28:711–720, 2014. doi: 10.1007/s10822-014-9747-x. (← pp. 8 and 31)

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001. doi: 10.1023/A:1008923215028. (← p. 3)

F. Neese. The orca program system. *WIRES Comput. Molec. Sci.*, 2(1):73–78, 2012. doi: 10.1002/wcms.81. (← p. 8)

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), 2019. doi: 10.1126/science.aaw1147. URL https://doi.org/10.1126/science.aaw1147. (← pp. 2, 3, and 3)

Simon Olsson. Generative molecular dynamics. *Current Opinion in Structural Biology*, 96:103213, 2026. doi: 10.1016/j.sbi.2025.103213. (← p. 1)

Danny Perez, Aidan Thompson, Stan Moore, Tomas Oppelstrup, Ilya Sharapov, Kylee Santos, Amirali Sharifian, Delyan Z. Kalchev, Robert Schreiber, Scott Pakin, Edgar A. Leon, James H. Laros, III, Michael James, and Sivasankaran Rajamanickam. Breaking the mold: Overcoming the time constraints of molecular dynamics on general-purpose hardware. *The Journal of Chemical Physics*, 162(7):072501, 2025. doi: 10.1063/5.0249193. (← p. 1)

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019. (← p. 6)

RDKit. RDKit: Open-source cheminformatics. Version 2026.03.3, 2026. URL https://github.com/rdkit/rdkit/releases/tag/Release_2026_03_3. (← pp. 5, 16, and 16)

Danyal Rehman, Charlie B. Tan, Yoshua Bengio, Avishek Joey Bose, and Alexander Tong. Autoregressive Boltzmann Generators. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/66096. (← pp. 2, 2, 3, 4, 5, 5, 7, 7, 16, 18, and 21)

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, 2015. URL https://proceedings.mlr.press/v37/rezende15.html. (← p. 2)

Jonathan S. Rosenfeld. Scaling laws for deep learning, 2021. URL https://arxiv.org/abs/2108.07686. (← pp. 6, 6, and 19)

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of langauge model performance. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 15841–15892. Curran Associates, Inc., 2024. doi: 10.52202/079017-0506. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1cded4f97cf5f01a284c574110b7e3b9-Paper-Conference.pdf. (← p. 6)

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *NeurIPS*, 2024. (← p. 5)

Andrej Šali, Eugene Shakhnovich, and Martin Karplus. How does a protein fold? *Nature*, 369(6477):248–251, 1994. doi: 10.1038/369248a0. (← p. 18)

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *NeurIPS*, 2024. (← p. 5)

Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 2013. (← p. 2)

Esteban G. Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8:217–233, 2010. doi: 10.4310/CMS.2010.v8.n1.a11. (← p. 3)

Charlie B. Tan, Joey Bose, Chen Lin, Leon Klein, Michael M. Bronstein, and Alexander Tong. Scalable equilibrium sampling with sequential boltzmann generators. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58467–58498, 2025a. URL https://proceedings.mlr.press/v267/tan25a.html. (← pp. 2, 3, and 5)

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 *Advances in Neural Information Processing Systems*, 2025b. URL https://neurips.cc/virtual/2025/poster/118702. (← p. 2)

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. (← p. 18)

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, 2024. doi: 10.1038/s41587-023-01773-0. (← p. 18)

Juan Viguera Diez and Simon Olsson. Mdqm9-nc dataset. Zenodo dataset, 2024. URL https://zenodo.org/records/10579242. (← pp. 8, 9, and 31)

Christopher von Klitzing, Denis Blessing, Henrik Schopmans, Pascal Friederich, and Gerhard Neumann. Learning boltzmann generators via constrained mass transport, 2026. URL https://arxiv.org/abs/2510.18460. (← p. 3)

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 02 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL https://doi.org/10.1021/ci00057a005. (← pp. 4 and 4)

Ziyang Yu, Yi He, Wenbing Huang, Wen Yan, and Yang Liu. CARD: Coarse-to-fine autoregressive modeling with radix-based decomposition for transferable free energy estimation. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64712. (← pp. 18 and 18)

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74348–74369, 2025. URL https://proceedings.mlr.press/v267/zhai25d.html. (← p. 5)

Tianhong Zhang, Hongli Li, Hualin Xi, Robert V. Stanton, and Sergio H. Rotstein. Helm: A hierarchical notation language for complex biomolecule structure representation. *Journal of Chemical Information and Modeling*, 52(10):2796–2806, 09 2012. ISSN 1549-9596. doi: 10.1021/ci3001925. URL https://doi.org/10.1021/ci3001925. (← p. 4)

Daniel M. Zuckerman. *Statistical Physics of Biomolecules: An Introduction*. CRC Press, Boca Raton, 2010. doi: 10.1201/b18849. (← p. 15)

Robert W. Zwanzig. High-temperature equation of state by a perturbation method. I. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954. doi: 10.1063/1.1740409. (← p. 15)

A Additional Background

Free Energy Estimation Consider two clearly identifiable states of our atomistic system of interest, for example the same molecule exposed to vacuum (A) and to water (B), with potentials U_A and U_B. At equilibrium and fixed temperature, their free-energy difference is defined by the implied partition functions Z_A and Z_B

$$\Delta F_{A\to B} = F_B - F_A = -\beta^{-1}\log(Z_B/Z_A)\,. \tag{5}$$

When both states are accessible, this corresponds to the log-odds of the system occupying state A over state B: $P(A)/P(B) = e^{\beta\Delta F_{A\to B}}$, a direct correspondence to the Boltzmann odds that neighboring microstates receive in (1), with F in place of U (Zuckerman, 2010).

Multiple approaches to free energy estimation exist (Lelièvre et al., 2010). Zwanzig's exponential averaging formula estimates the free energy difference from samples of a single state (Zwanzig, 1954). It is the one-sided importance-sampling estimator of $\log(Z_B/Z_A) = -\beta\Delta F_{A\to B}$, with unnormalized densities $\tilde{p}_B = e^{-\beta U_B}$ and $\tilde{p}_A = e^{-\beta U_A}$ and raw importance weight $e^{-\beta\Delta U}$. The forward estimator draws $\mathbf{x}\sim p_A$ and the reverse estimator draws $\mathbf{x}\sim p_B$,

$$\begin{aligned}\Delta F_{A\to B} &= -\beta^{-1}\log\mathbb{E}_{\mathbf{x}\sim p_A}\left[\exp\big(-\beta\Delta U(\mathbf{x})\big)\right]\\ &= +\beta^{-1}\log\mathbb{E}_{\mathbf{x}\sim p_B}\left[\exp\big(+\beta\Delta U(\mathbf{x})\big)\right],\end{aligned} \tag{6}$$

where $\Delta U(\mathbf{x}) = U_B(\mathbf{x}) - U_A(\mathbf{x})$. Jensen's inequality implies that a finite-sample reverse estimate is biased low and a finite-sample forward estimate is biased high, $\mathbb{E}[\widehat{\Delta F}^{\text{rev}}] \le \Delta F_{A\to B} \le \mathbb{E}[\widehat{\Delta F}^{\text{fwd}}]$.

Bennett's acceptance ratio (BAR) combines samples from both states through optimal bridge sampling (Bennett, 1976; Meng & Wong, 1996). The Boltzmann densities and the earlier free-energy relation give $p_B(\mathbf{x})/p_A(\mathbf{x}) = e^{\beta[\Delta F_{A\to B} - \Delta U(\mathbf{x})]}$. Pooling n_A samples from A and n_B from B, a logistic function converts these density ratios into probabilities of sample origin. BAR estimates $\Delta F_{A\to B}$ by fitting a single parameter δ through logistic maximum likelihood, where $p_\delta(B \mid \mathbf{x})$ models the probability of origin B given $\mathbf{x}$:

$$p_\delta(B\mid\mathbf{x}) = \sigma\left(\beta[\delta - \Delta U(\mathbf{x})] + \log\frac{n_B}{n_A}\right), \qquad \sigma(t) = (1+e^{-t})^{-1}. \tag{7}$$

Zero Free Energy Proposals Consider a Boltzmann Generator trained as a proposal $q_{\theta,A}$ for one physical state A, with potential U_A and Boltzmann density p_A. Unconventionally, we can treat the proposal $q_{\theta,A}$ as an standalone thermodynamic state A° with potential $U_{A^\circ}(\mathbf{x}) = -\beta^{-1}\log q_{\theta,A}(\mathbf{x})$ (Ding & Zhang, 2021). Because $q_{\theta,A}$ is normalized, this proposal state has unit partition function and therefore zero free energy $F_{A^\circ} = -\beta^{-1}\log Z_{A^\circ} = 0$.

Sampling $\mathbf{x}\sim q_{\theta,A}$ is sampling the proposal state A°, and the unnormalized importance weights routinely estimated during SNIS (2) can be directly used within a free energy estimator, such as the Zwanzig method, yielding the absolute free energy of A due to the proposal state A°'s zero free energy. A free energy difference between two physical states, such as vacuum (A) and water (B), now reduces to two such independent proposal-to-target absolute free energy estimates (Figure 7). Each physical state has its own learned proposal state, accessible via its corresponding q_θ, and the proposal-to-proposal leg vanishes, $\Delta F_{A^\circ\to B^\circ} = 0$. The resulting cycle identity is $\Delta F_{A\to B} = \Delta F_{A\to A^\circ} - \Delta F_{B\to B^\circ}$, where each leg can be estimated by the forward or reverse Zwanzig formula, or by BAR, depending on the availability of samples from the physical state. In summary, the Zero Free Energy Proposal approach enables straightforward free energy estimation using Boltzmann Generators.

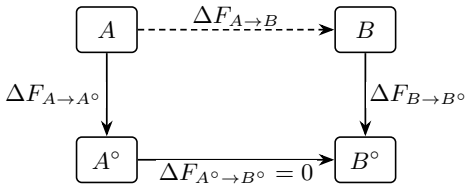


Figure 7: **Zero-Free-Energy Proposal Cycle.** Normalized generative proposal state A° and B° have vanishing free energy, so $\Delta F_{A^\circ\to B^\circ} = 0$.

B Discretized Boltzmann Generators

Quantization The coordinates of an atomistic system similar to the one described in Section 2 can be coarse-grained through a quantization of each spatial dimension. We write each atomic coordinate as $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$ with $i \in \{1, \dots, n\}$ and $d \in \{1, 2, 3\}$. An interval $I = [a, a + L)$ along each axis, with origin a and width L, contains the coordinates to be discretized. In practice this width may match the cubic periodic box of a molecular simulation. The interval is partitioned into $b \geq 1$ equal cells of width $\Delta = L/b$,

$$\Delta = \frac{L}{b}, \qquad I_l = [a + l\Delta, a + (l + 1)\Delta), \qquad l \in \mathcal{A}_b = \{0, \dots, b - 1\}. \tag{8}$$

The scalar quantizer is $\kappa_\Delta(x) = l$ iff $x \in I_l$. Applied independently to each Cartesian component, it yields the voxel code $\mathbf{k} = (k_{id}) \in \mathcal{A}_b^{n\times 3}$ with $k_{id} = \kappa_\Delta(x_{id})$.

Each code $\mathbf{k}$ is a voxel macrostate: $C(\mathbf{k}) = \prod_{i=1}^{n} \prod_{d=1}^{3} I_{k_{id}}$. Its configurational partition function, free energy, and equilibrium probability are

$$Z_\Delta(\mathbf{k}) = \int_{C(\mathbf{k})} \exp[-\beta U(\mathbf{x})]\, \mathrm{d}\mathbf{x}, \qquad F_\Delta(\mathbf{k}) = -\beta^{-1} \log Z_\Delta(\mathbf{k}), \qquad p_\Delta(\mathbf{k}) = \frac{Z_\Delta(\mathbf{k})}{Z_\mathbf{x}}. \tag{9}$$

The cells partition the continuous configuration space Ω, so $Z_\mathbf{x} = \sum_\mathbf{k} Z_\Delta(\mathbf{k})$. In practice, a sample from p_Δ is obtained by drawing $\mathbf{x} \sim p$ and setting $k_{id} = \kappa_\Delta(x_{id})$.

Tokenization. A tokenizer τ is a bijection from a voxel code $\mathbf{k}$ to a length-m sequence $\mathbf{z} = \tau(\mathbf{k})$. As such, it does not change the cells, but only serializes them into a sequence.

Dimensional tokenization emits the three axis indices (k_{i1}, k_{i2}, k_{i3}) as separate symbols, so $m = 3n$. Each symbol is a bin index on one Cartesian axis, and the alphabet size equals the number of cells b on that axis of the cube. This approach is used within this study and Rehman et al. (2026).

Octree tokenization serializes the same cube by nested octants. At each step the current cell is split into eight equal children, and the symbol records which child contains the atom. A sequence of such octants is a path from the cube to a leaf voxel. With T symbols per atom, $m = Tn$. The leaf grid matches dimensional tokenization when the path depth matches the degree of axis binning.

Decoding Order. Given a tokenization, we are required to select at least one decoding order σ. We choose a pre-defined atom-inserting decoding order following an atom order π of the chemical graph of the system G. Because the space of possible decoding orders is large, we constrain it empirically. Based on previous work (Gebauer et al., 2019; Cheng et al., 2025) and physical intuition, we aim to decode atoms in an order that is spatially local. Intuitively, atoms that are close to each other within the decoding order should be more likely to be close in coordinate space: During training, we place the solute first, followed by solvent molecules ordered by $\rho = \chi_{\text{inter}}(G, \mathbf{x})$. *Within* a given molecule (χ_{intra}) we rely on the ordering induced by molecular graph canonicalization heuristic (RDKit; Cheng et al., 2025) to arrive at an ordering.

Once a decoding order σ has been defined, each atom within the graph order has to be assigned to a 3D coordinate for training. We define $\mathrm{Aut}(G) \leq S_n$ as the atom permutations that preserve bonds and labels. If G is a disjoint union of molecular components, this group splits into intra-component automorphisms and permutations of isomorphic components. Each $\gamma \in \mathrm{Aut}(G)$ acts on a frame by row permutation $\mathbf{x} \mapsto \boldsymbol{P}_\gamma \mathbf{x}$ and leaves the unlabeled conformer unchanged and, for a symmetry-invariant potential, its Boltzmann density unchanged.

Within a given molecule, equivalent atom labels can be permuted among coordinate positions (e.g. terminal methyl-group rotamers). We restrict ourselves to independent terminal orbits of $\mathrm{Aut}(G)$. During training we either sample these orbits, or break them with an RDKit canonicalization heuristic (RDKit). In practice we have found the active re-sampling of atom assignment order in rotamer orbits to be non-consequential.

Between identical molecules, we select their training order with $\rho = \chi_{\text{inter}}(G, \mathbf{x})$. For water, this permutation satisfies $d_{\rho(1)} \leq \cdots \leq d_{\rho(K)}$, where $d_j = \min_{i \in V(G_0)} \|\mathbf{x}_{o_j} - \mathbf{x}_i\|_2$ and o_j denotes the oxygen atom of water j. Thus, waters are ordered by increasing distance

from their oxygen to the nearest solute atom. The complete atom order is $\pi = \chi(G, \mathbf{x}) = \chi_{\text{intra}}(G_0) \,\|\, \chi_{\text{intra}}(G_{\rho(1)}) \,\|\, \cdots \,\|\, \chi_{\text{intra}}(G_{\rho(K)})$.

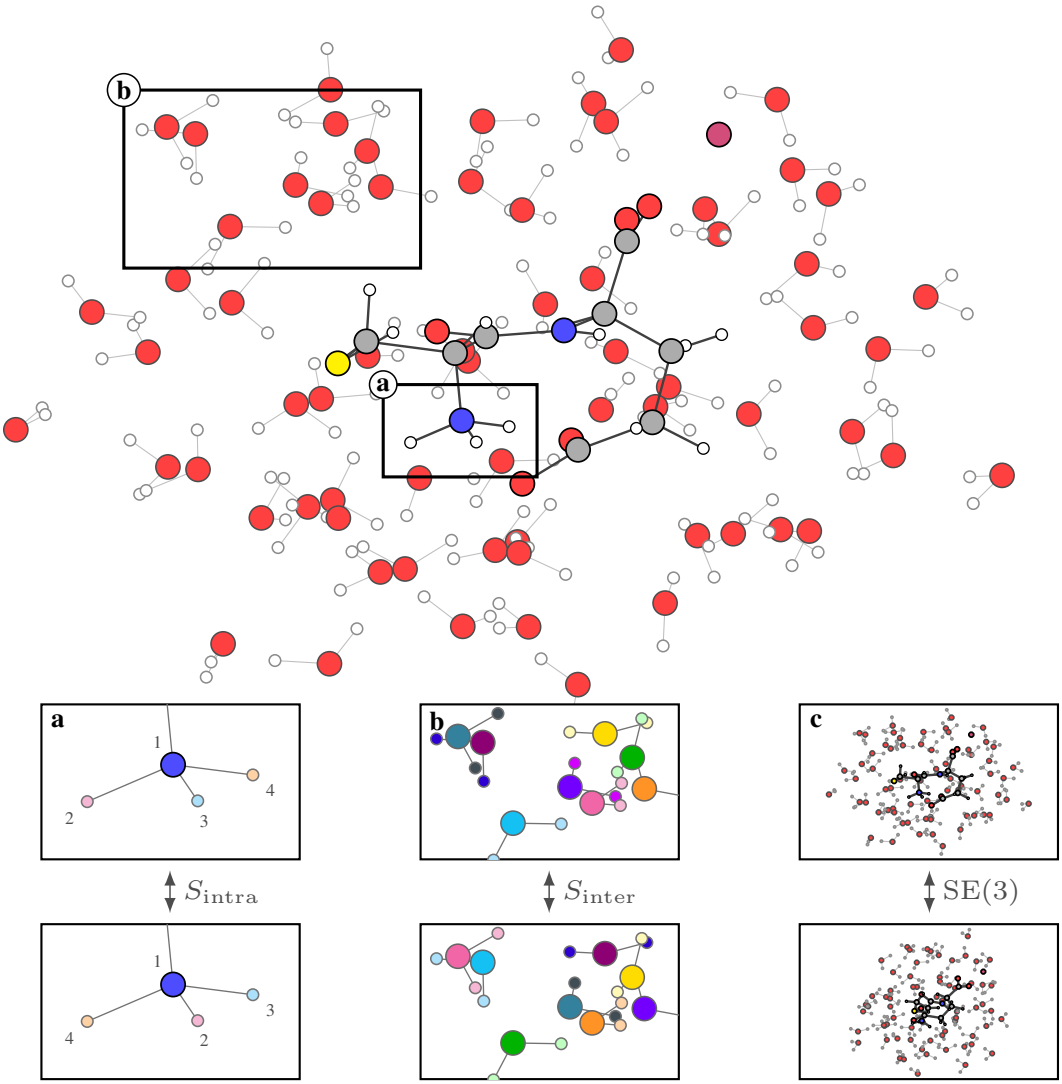


Figure 8: **Symmetries Within Serialised Atomistic Systems.** Shown are examples for (a) a *within* molecule terminal rotamer with its permutation S_{intra}, (b) a *between* molecule solvent indistinguishability S_{inter}, and (c) rigid rototranslation $\text{SE}(3)$ of the overall conformation.

Dequantization. A reconstruction kernel $\mathcal{R}_\Delta$ maps a voxel $\mathbf{k}$ to a continuous conformation in that cell, so that test functions $\phi(\mathbf{x})$ and the potential $U(\mathbf{x})$ can be evaluated. We consider uniform sampling within a cell and its deterministic midpoint $r_l = a + (l + \frac{1}{2})\Delta$, the conditional expectation of the former, applied independently to each Cartesian component,

$$\mathcal{R}_\Delta^{\text{mid}}(\mathrm{d}y \mid l) \equiv \delta_{r_l}(\mathrm{d}y), \qquad \mathcal{R}_\Delta^{\text{unif}}(\mathrm{d}y \mid l) \equiv \frac{\mathbf{1}_{y \in I_l}}{\Delta}\,\mathrm{d}y, \tag{10}$$

The two kernels provide estimates of the same cell integral $Z_\Delta(\mathbf{k})$. Under uniform reconstruction, $e^{-\beta U(\mathbf{y})}$ is an unbiased one-sample Monte Carlo estimate of $Z_\Delta(\mathbf{k})/\Delta^{3n}$. In contrast, the midpoint kernel acts as a zeroth-order cubature at the cell center,

$$F_\Delta(\mathbf{k}) \approx U(\mathbf{r}_\mathbf{k}) - \beta^{-1}\, 3n \log \Delta, \tag{11}$$

which is approximate but has zero reconstruction variance conditional on $\mathbf{k}$. The log-volume term is constant across $\mathbf{k}$ and cancels in self-normalized ratios for a fixed system.

C Related Work

Discretized Representations of Biomolecules Discretizing biomolecular structure is an established approach to improve the sampling of and search over molecular conformations, albeit primarily for proteins and at considerable coarse graining. Historic examples include simple 2D lattice protein models confining residue positions to discrete sites (Lau & Dill, 1989; Šali et al., 1994), 3D fragment libraries (Kolodny et al., 2002) and basic, learned structural motifs (de Brevern et al., 2000). Recently, learned vector-quantized codebooks of protein conformation (van den Oord et al., 2017; van Kempen et al., 2024) have been identified, which have also been paired with discrete generative protein models (Hayes et al., 2025), albeit without any goal of matching the Boltzmann distribution or accessible likelihoods. While the presented work relies on a relatively finegrained quantization to reduce the error during importance weighting more coarse grained and learned quantization methods for PBGs are conceivable, although challenges related to non-uniformly sized quantization cells and high variance during the macrostate free energy estimation need to be anticipated.

Autoregressive Generation of Biomolecule Conformers Autoregressive models have primarily sampled 3D structure of biomolecules via iterative atom insertion. Early work, such as *G-SchNet* inserted atoms sequentially into continuous space using learned pairwise, internal, distance distributions (Gebauer et al., 2019). More recent work applied similar approaches to the sampling of global coordinates (Cheng et al., 2025). In contrast, octree-based methods (Lu et al., 2025) autoregressively denoise atom locations with a final step mapping atom coordinates into continuous space. In contrast to these works Autoregressive Boltzmann Generators (Rehman et al., 2026) and the family of Prefix-Conditioned Boltzmann Generators we describe operate over discrete coordinate representations and rely on parameter-free quantization and dequantization methods.

Generative Modeling for Free Energy Estimation Boltzmann generators have previously been used for single-system free energy estimation using the zero free energy proposal approach, including for Guest-Host binding free energy (Ding & Zhang, 2021). Recently, this approach was extended to transferable Boltzmann generators, and applied to implicit solvation free energy estimations of small molecules (Yu et al., 2026). Our work is closely related to (Yu et al., 2026), and primarily differs by choosing a purely discrete coordinate modeling, a simple prefix conditioning based prompt, atom insertion over atom denoising, random $SE(3)$ augmentations to achieve learned equivariance instead of conformer pose canonicalization, and by moving towards explicit solvation. Alternative generative modeling approaches to facilitate free energy estimation include Boltzmann emulators which sample high quality starting conformers for free energy simulations, but lack accessible and exact likelihoods (Lewis et al., 2025). Recently, such emulators, which can be referred to as alchemical Boltzmann emulators, have been extended to sample alchemical states, too, thereby further accelerating alchemical free energy simulations (Bengio et al., 2026).

D Additional Results

D.1 Scaling Model

Scaling Data. In order to study the relationship of PBG model size and training compute with sampling quality and system size, a series of 130 models were trained. These models spanned 5 distinct model sizes between 8M and 85M parameters, ca. 8 training compute budget ranges spanning $D/N = 12$ to $D/N = 1760$, as well as 12 distinct bin counts / discretization resolutions from $b = 256$, $\Delta = 12.8\,\mathrm{pm}$ to $b = 65{,}537$, $\Delta \approx 0.05\,\mathrm{pm}$. All models were trained with *Many Peptides* data using a Warmup-Stable-Decay schedule for amortized estimation of training compute effects. Once trained, 91 molecular systems were evaluated per checkpoint. Next to the terminal training loss, the three previously introduced, unclipped and sliced metrics were estimated (Table 4) yielding a final dataset of 130 checkpoints with $91 \times 3 + 1$ values each.

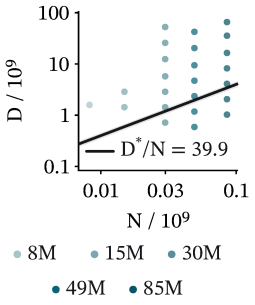


Figure 9: **Scaling Model Data.** Points show evaluated D and N combinations for 8M–85M models with 4096 coordinate bins. The compute-optimal scaling model estimate $D^*(N)$ is shown as a line.

Scaling Model. Our goal is to fit a simplified scaling model using bayesian inference. Starting from the general scaling law $G = AN^{-\alpha} + BD^{-\beta}$ (Rosenfeld, 2021), we tie the two exponents $\alpha = \beta$ for modelling stability and based on the assumption of approximate equivalence (Kaplan et al., 2020), yielding $G = AN^{-\alpha} + BD^{-\alpha}$. Next to the exponent equivalence assumption, we rely on the assumption that training compute C is proportional to model size and loss tokens seen $C \approx kND$, which is often simplified to $C = 6ND$. Given these assumptions, our primary interest is the compute-optimal ratio of $r = D^*/N^*$. Preferably, we would like to directly place a prior over the this ratio.

We proceed to rewrite the general scaling law to directly estimate the compute-optimal ratio. At fixed compute, we must choose how to allocate resources between model size N and training tokens D to minimize G. Using $C \approx kND$, we rewrite:

$$D = \frac{C}{kN}, \qquad G(N \mid C) = \frac{A}{N^\alpha} + \frac{B}{(C/(kN))^\alpha} = \frac{A}{N^\alpha} + B\left(\frac{k}{C}\right)^\alpha N^\alpha.$$

Increasing the model parameter size N reduces the first term but increases the second. To find the minimum at fixed compute, we differentiate G with respect to N:

$$\frac{\partial G}{\partial N} = -\alpha A N^{-\alpha-1} + \alpha B\left(\frac{k}{C}\right)^\alpha N^{\alpha-1} = 0 \qquad \text{at } N = N^*.$$

Rearranging and substituting $D^* = C/(kN^*)$ gives

$$\frac{A}{(N^*)^\alpha} = \frac{B}{(D^*)^\alpha}.$$

Inserting $r = D^*/N^*$, we obtain

$$r = \left(\frac{B}{A}\right)^{1/\alpha}. \tag{12}$$

Knowing the compute-optimal ratio r does not determine both coefficients A and B. Therefore we choose to define $G_0 = 2\sqrt{AB}$. Using $B = Ar^\alpha$, we obtain

$$G_0 = 2Ar^{\alpha/2}, \qquad A = \frac{G_0}{2}r^{-\alpha/2}, \qquad B = \frac{G_0}{2}r^{\alpha/2}.$$

Substituting these expressions into the scaling law gives a rewritten equivalent form, dependent on G_0, r, and α:

$$\boxed{G = \frac{G_0}{2}\left[(N\sqrt{r})^{-\alpha} + (D/\sqrt{r})^{-\alpha}\right]}. \tag{13}$$

Here, G_0 controls the overall scale of the resource-dependent G, r specifies the compute-optimal allocation, and α controls how quickly G decreases as resources increase.

Table 4: **Sampling with Unclipped Importance Weights.** Median metrics across *Many Peptides* test peptides, with 10^4 samples per system. Best values are **bold**; second-best values are underlined.

	2AA (30 systems)			4AA (30 systems)			8AA (29 systems)¶		
Model	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓
ROBIN ($T = 1.0$)	0.002	2.462	0.481	0.002	3.299	0.454	0.001	8.834	0.585
PBG 30M ($D/N = 1760$)	0.331	0.281	0.092	0.047	0.879	0.150	0.002	6.646	0.465
PBG 49M ($D/N = 416$)	0.270	0.272	0.094	0.053	0.708	0.147	0.001	9.063	0.588
PBG 85M ($D/N = 768$)	0.608	0.256	**0.078**	0.218	0.541	0.106	0.005	4.002	0.343
PBG 160M ($D/N = 352$)	0.488	0.260	0.095	0.132	0.534	0.115	0.002	4.361	0.457
PBG 320M ($D/N = 86$)	0.428	0.261	0.090	0.107	0.623	0.122	0.004	4.672	0.356
PBG 640M ($D/N = 48$)	**0.703**	**0.206**	0.079	**0.271**	**0.420**	**0.096**	**0.011**	**2.239**	**0.250**
PBG-Joint 160M ($D/N = 198$)	0.194	0.317	0.099	0.073	0.630	0.129	0.002	5.615	0.416

Model Specification. Consider a checkpoint with index i and a system with index j. Let N_i be the non-embedding parameter count, D_i the delivered loss-token count, b_i the bin count, and n_j the atom count.

$$N_{9i} = N_i/10^9, \quad D_{9i} = D_i/10^9, \quad x_j = \log(n_j/65)$$

The terminal training loss L_i and the transformed metrics Y_{mij} are defined as

$$Y_{Eij} = \log\Big(\mathcal{E}\text{-}W_{2\,ij}^*\Big), \quad Y_{Sij} = \log\Big(\mathbb{T}\text{g-}\mathcal{SW}_{2ij}^*\Big), \quad Y_{Rij} = -\log(\text{rESS}^*{}_{ij})$$

Here $m \in \{E, S, R\}$ indexes $\mathcal{E}$-W_2^*, $\mathbb{T}$g-$\mathcal{SW}_2^*$, and rESS*, respectively. The transformed metrics are drawn from a student t distribution, with a sigmoid applied to rESS* given its upper ($= 1$) and lower ($= 1/n_{sample}$) bounds.

$$\begin{aligned} L_i &\sim t_{\nu_L}(\mu_{Li}, \sigma_L), \\ Y_{mij} &\sim t_{\nu_m}(\mu_{mij}, \sigma_m), \\ \mu_{Eij} &= \eta_{Eij}, \quad \mu_{Sij} = \eta_{Sij}, \\ \mu_{Rij} &= \log(10^4)\,\text{sigmoid}(\eta_{Rij}). \end{aligned}$$

The latent variables η are dependent on G_i via a set of adapters for each transformed metric. E denotes the irreducible training loss and a_m an analogous metric specific intercept. The covariate p_m captures the scaling of the transformed metric with atom count, while k_m does the equivalent for G. A δ_m intercept represents discretization resolution dependent effects.

$$\begin{aligned} G_i &= \frac{G_0}{2}\left[(N_{9i}\sqrt{r})^{-\alpha} + (D_{9i}/\sqrt{r})^{-\alpha}\right], \\ \mu_{Li} &= E + G_i + \delta_L[b_i], \\ \eta_{mij} &= a_m + p_m x_j + k_m \log G_i + \delta_m[b_i]. \end{aligned} \tag{14}$$

Parameter	Prior
E	$\mathcal{N}(3.55, 0.15^2)$
$\log G_0$	$\mathcal{N}\Big(\log\big(2\sqrt{0.003 \cdot 0.05}\big), \frac{1.5^2}{2}\Big)$
r	$\mathcal{N}^+(20, 10)$, truncated to $r > 0$
$\log \alpha$	$\mathcal{N}(\log 0.8, 0.75^2)$
$\delta_L[b], b \neq 4096$	$\mathcal{N}(0, 3^2)$
$\delta_m[b], b \neq 4096$	$\mathcal{N}(0, 1^2), m \in \{E, S, R\}$
$a_E,\ a_S,\ a_R$	$\mathcal{N}(5, 2^2), \quad \mathcal{N}(1, 2^2), \quad \mathcal{N}(4.5, 2^2)$
$p_E,\ p_S,\ p_R$	$\mathcal{N}(2, 0.5^2), \quad \mathcal{N}(0.7, 0.5^2), \quad \mathcal{N}(1.2, 0.8^2)$
$k_E,\ k_S,\ k_R$	$\mathcal{N}(1.2, 0.8^2), \quad \mathcal{N}(0.7, 0.8^2), \quad \mathcal{N}(1.2, 0.8^2)$
$\log \sigma_L$	$\mathcal{N}(\log 0.01, 1^2)$
$\log \sigma_E,\ \log \sigma_S$	$\mathcal{N}(\log 0.5, 0.75^2)$ for each parameter
$\log \sigma_R$	$\mathcal{N}(\log 1.5, 0.75^2)$
$\log(\nu_m - 2)$	$\mathcal{N}(\log 10, 0.75^2), m \in \{L, E, S, R\}$

A generalized model posterior was sampled via MCMC. In order to weight the model towards accuracy of high-compute checkpoints, likelihoods were tilted by relative compute intensity.

$$w_i = \frac{(6N_iD_i)^{1/2}}{\frac{1}{130}\sum_{k=1}^{130}(6N_kD_k)^{1/2}}.$$

$$\pi_w(\theta \mid \mathcal{D}) \propto p(\theta)\prod_{i=1}^{130}\left[t_{\nu_L}(L_i;\mu_{Li},\sigma_L)\prod_{j=1}^{91}\prod_{m\in\{E,S,R\}}t_{\nu_m}(Y_{mij};\mu_{mij},\sigma_m)\right]^{w_i}.$$

Model Estimates. The scaling model matched the terminal training NLL with high accuracy, and explained ca. 60-70% of variance in the transformed evaluation metrics (Table 6). Among these evaluation metrics, rESS* had the strongest coupling to the model loss, followed by $\mathcal{E}$-W_2^* and $\mathbb{T}$g-$\mathcal{SW}_2^*$ (Table 5). Both $\mathcal{E}$-W_2^* and rESS* presented a near-quadratic scaling with increases in atomic system size, while $\mathbb{T}$g-$\mathcal{SW}_2^*$ showed linear scaling.

Table 5: **Scaling Model Parameters.** Joint fit to training NLL and sampling quality metrics. Values are posterior medians ± half the width of the 95% credible interval.

Group	Parameter	Posterior median ± half 95% CI
core G	G_0	0.0785 ± 0.0123
	r	39.94 ± 4.893
	α	0.216 ± 0.0251
train NLL	E	3.522 ± 0.0122
	σ_L	$0.00333 \pm 7.57 \times 10^{-4}$
	ν_L	2.693 ± 0.575
$\log(\mathcal{E}\text{-}\mathcal{W}_2^*)$	a_E	9.280 ± 0.701
	p_E	2.168 ± 0.0203
	k_E	3.620 ± 0.427
	σ_E	0.513 ± 0.0111
	ν_E	2.519 ± 0.116
$\log(\mathbb{T}\text{g-}\mathcal{SW}_2^*)$	a_S	4.614 ± 0.476
	p_S	1.012 ± 0.0155
	k_S	2.468 ± 0.292
	σ_S	0.406 ± 0.00949
	ν_S	4.253 ± 0.344
$-\ln(\text{rESS}^*)$	a_R	11.39 ± 0.893
	p_R	1.788 ± 0.0226
	k_R	4.733 ± 0.550
	σ_R	0.757 ± 0.0190
	ν_R	2.501 ± 0.135

Table 6: **Scaling Model Fit.** In-sample R^2 for NLL and the transformed sampling metrics.

Response	R^2
train NLL	0.9744
$\log(\mathcal{E}\text{-}\mathcal{W}_2^*)$	0.6534
$\log(\mathbb{T}\text{g-}\mathcal{SW}_2^*)$	0.5732
$-\ln(\text{rESS}^*)$	0.6782

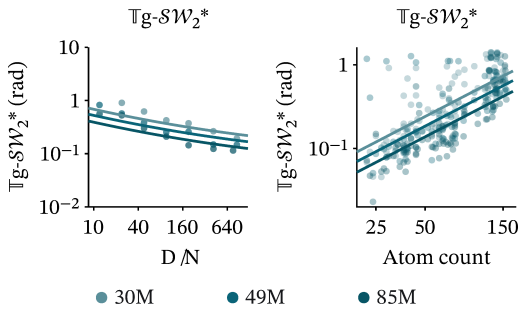


Figure 10: **Torsion Sampling Error Scales with Compute and System Size.** Left: $\mathbb{T}$g-$\mathcal{SW}_2^*$ on *Many Peptides* versus loss tokens per non-embedding parameter (D/N). Right: Per-system $\mathbb{T}$g-$\mathcal{SW}_2^*$ versus atom count for 15M–85M models at $D/N = 192$. Both panels use 4096 coordinate bins. Lines show scaling-model estimates at $n = 65$ atoms (left) and $D/N = 192$ (right).

Among the discretization resolutions explored, we observed a steep penalty in sampling quality at coarse resolutions of $b \leq 2^{10} = 1024$ bins per axis ($\Delta \geq 3.2\,\text{pm}$) (Figure 11). In contrast to (Rehman et al., 2026), we also observed detrimental effects of high discretization resolutions, an effect we ascribe to logit gradient bottleneck as we increase the vocabulary size and keep the latent token dimension fixed (Godey & Artzi, 2026).

When extrapolating to larger models of 160M, 320M, and 640M, we observed reasonable adherence to the empirical scaling model (Figure 12).

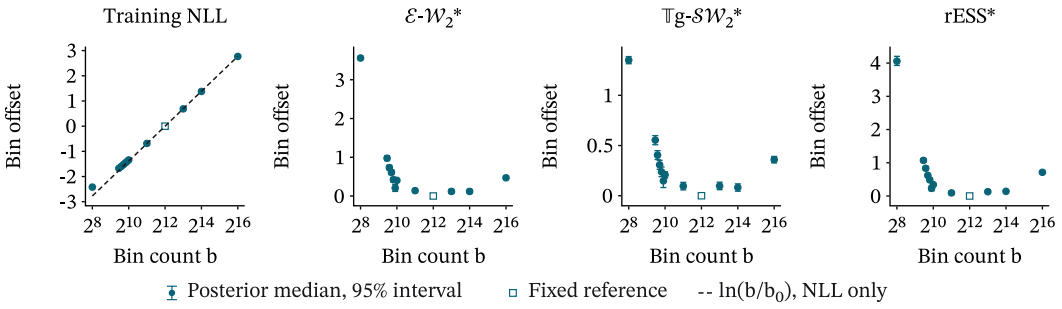


Figure 11: **Fitted Discretization Resolution Offsets.** Shown are estimated discretization effects on training NLL, $\mathcal{E}$-W_2^*, $\mathbb{T}$g-$\mathcal{SW}_2^*$, and rESS^*. Points show posterior medians with 95% credible intervals. Open squares mark offsets fixed at zero for $b_0 = 4096$. The dashed curve in the NLL panel represents $\ln(b/b_0)$.

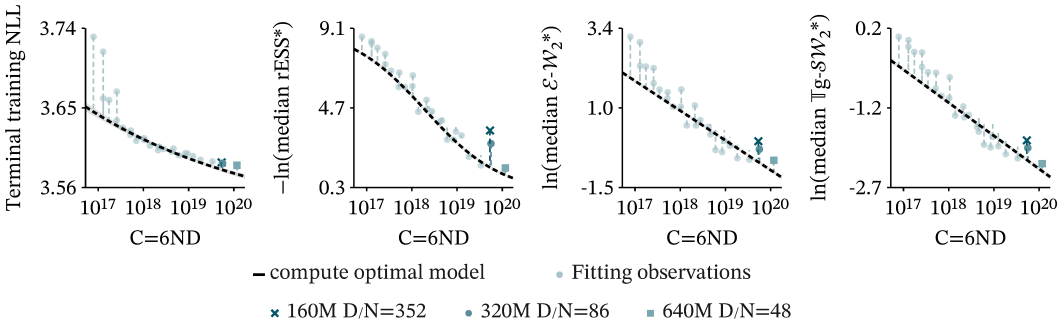


Figure 12: **Compute Scaling and Held-Out Checkpoint Predictions.** From left to right: terminal training NLL, $-\ln(\mathrm{rESS}^*)$, $\ln(\mathrm{median}\,\mathcal{E}\text{-}W_2^*)$, and $\ln(\mathbb{T}\text{g-}\mathcal{SW}_2^*)$ versus the training-compute proxy $C = 6ND$. All evaluation summaries use the same *Many Peptides* systems ¶. Filled markers show observed checkpoints and their connecting lines end at the scaling model's predicted sampling quality. The black curves shows compute-optimal sampling quality estimates with 95% generalized-posterior intervals. Three extrapolated and held-out model checkpoints, labeled 160M, 320M, and 640M checkpoints are shown.

D.2 Quantization and Reconstruction Effects

Sampling error in Prefix-Conditioned Boltzmann Generators can be ascribed to either *Quantization Error*, an insufficiency of the discretization resolution, or *Generator Error*, an insufficiency in learning the quantized target distribution p_Δ. In the following we describe a set of experiments and theoretical considerations to understand *Quantization Error* in isolation.

Recall the midpoint and uniform reconstruction kernels, $\mathcal{R}_\Delta^{\mathrm{mid}}$ and $\mathcal{R}_\Delta^{\mathrm{unif}}$, respectively (Appendix B). Given a quantized macrostate $\mathbf{k}$, the midpoint kernel returns the cell center $\mathbf{r_k} \in \mathbb{R}^{n\times 3}$, while the uniform kernel draws a conformation $\mathbf{y} \sim \mathcal{R}_\Delta^{\mathrm{unif}}(\cdot \mid \mathbf{k})$ uniformly within the cell $C(\mathbf{k})$, with $\mathbf{y} \in \mathbb{R}^{n\times 3}$.

Energetic Discretization Error under Midpoint Reconstruction. To isolate Quantization Error, we draw $\mathbf{x} \sim p$ and set $\mathbf{k} = \kappa_\Delta(\mathbf{x})$, applying the quantizer componentwise. We then directly reconstruct the conformer in a *roundtrip*, starting with the midpoint reconstruction kernel. This gives us a midpoint reconstruction induced energy error and its expectation

$$\begin{aligned}\Delta U^{\mathrm{mid}} &\equiv U(\mathbf{r_k}) - U(\mathbf{x}),\\ \mathbb{E}[\Delta U^{\mathrm{mid}}] &= \sum_{\mathbf{k}} p_\Delta(\mathbf{k})\left[U(\mathbf{r_k}) - \mathbb{E}_p[U(\mathbf{x}) \mid \mathbf{k}]\right].\end{aligned}$$

Let's denote $\boldsymbol{\delta} = \mathbf{x} - \mathbf{r_k} \in \mathbb{R}^{n\times 3}$ the displacement of the original conformation from its cell center. Each Cartesian component of $\boldsymbol{\delta}$ lies in $[-\Delta/2, \Delta/2)$. Considering the limit $\Delta \to 0$ at fixed n and assuming a smooth potential and finite required moments, a Taylor expansion around the cell center gives

$$U(\mathbf{x}) = U(\mathbf{r_k}) + \nabla U(\mathbf{r_k})^\top \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top \nabla^2 U(\mathbf{r_k})\boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|^2).$$

The conditional Boltzmann density of the displacement favors lower-energy regions. Expanding this density within the cell gives

$$\begin{aligned}\mathbb{E}_p[\boldsymbol{\delta} \mid \mathbf{k}] &= -\frac{\beta\Delta^2}{12}\nabla U(\mathbf{r_k}) + o(\Delta^2),\\ \mathbb{E}_p[\boldsymbol{\delta}\boldsymbol{\delta}^\top \mid \mathbf{k}] &= \frac{\Delta^2}{12}\mathbf{I}_{3n} + o(\Delta^2).\end{aligned}$$

where $\mathbf{I}_{3n}$ is the identity matrix and coordinate arrays are treated as vectors in $\mathbb{R}^{3n}$. Taking the conditional expectation of $\Delta U^{\mathrm{mid}} = U(\mathbf{r_k}) - U(\mathbf{x})$ in the Taylor expansion gives

$$\mathbb{E}[\Delta U^{\mathrm{mid}} \mid \mathbf{k}] = -\nabla U(\mathbf{r_k})^\top \mathbb{E}_p[\boldsymbol{\delta} \mid \mathbf{k}] - \frac{1}{2}\operatorname{tr}\left[\nabla^2 U(\mathbf{r_k})\mathbb{E}_p[\boldsymbol{\delta}\boldsymbol{\delta}^\top \mid \mathbf{k}]\right] + o(\Delta^2).$$

Substituting the two displacement moments from above then yields

$$\mathbb{E}[\Delta U^{\mathrm{mid}} \mid \mathbf{k}] = \frac{\beta\Delta^2}{12}\|\nabla U(\mathbf{r_k})\|^2 - \frac{\Delta^2}{24}\operatorname{tr}\nabla^2 U(\mathbf{r_k}) + o(\Delta^2).$$

Averaging over $\mathbf{k} \sim p_\Delta$ and replacing the cell-center averages by their continuum limits in the leading coefficient leaves us with

$$\mathbb{E}[\Delta U^{\mathrm{mid}}] = \frac{\beta\Delta^2}{12}\mathbb{E}_p[\|\nabla U\|^2] - \frac{\Delta^2}{24}\mathbb{E}_p[\operatorname{tr}\nabla^2 U] + o(\Delta^2).$$

Provided the boundary term vanishes, integration by parts gives

$$\begin{aligned}0 &= \int_\Omega \nabla\cdot[p(\mathbf{x})\nabla U(\mathbf{x})]\,\mathrm{d}\mathbf{x}\\ &= \mathbb{E}_p[\operatorname{tr}\nabla^2 U] - \beta\mathbb{E}_p[\|\nabla U\|^2].\end{aligned}$$

The mean curvature term thus cancels half of the first contribution, leaving us with a leading-order description of the expected energetic discretization error under midpoint reconstruction as a function of the mean squared force $E_p[\|\nabla U\|^2]$ and the squared bin size Δ.

$$\boxed{\mathbb{E}[\Delta U^{\mathrm{mid}}] = \frac{\beta\Delta^2}{24}\mathbb{E}_p[\|\nabla U\|^2] + o(\Delta^2).} \tag{15}$$

Energetic Discretization Error under Uniform Reconstruction. We can now proceed to describe the expected energetic discretization error under uniform reconstruction. The uniform reconstruction can be expressed as an additional random displacement $\boldsymbol{\eta}$ from the cell center:

$$\mathbf{y} = \mathbf{r_k} + \boldsymbol{\eta}, \qquad \eta_{id} \overset{\text{iid}}{\sim} \text{Unif}[-\Delta/2, \Delta/2),$$

drawn independently of $\mathbf{x}$ conditional on $\mathbf{k}$. Its energy change satisfies

$$\Delta U^{\text{unif}} \equiv U(\mathbf{y}) - U(\mathbf{x}) = \Delta U^{\text{mid}} + U(\mathbf{r_k} + \boldsymbol{\eta}) - U(\mathbf{r_k}).$$

The uniform displacement has zero mean, so its linear contribution vanishes. Integrating the quadratic term over the cell gives

$$\mathbb{E}[\Delta U^{\text{unif}} \mid \mathbf{k}] = \mathbb{E}[\Delta U^{\text{mid}} \mid \mathbf{k}] + \frac{\Delta^2}{24} \operatorname{tr} \nabla^2 U(\mathbf{r_k}) + o(\Delta^2).$$

After averaging over cells and applying the same canonical identity, we obtain the analogous leading-order energetic discretization error under uniform reconstruction. Of note, to leading order, this expected error is double the expected energetic discretization error under midpoint reconstruction $\mathbb{E}[\Delta U^{\text{unif}}] = 2\,\mathbb{E}[\Delta U^{\text{mid}}] + o(\Delta^2)$

$$\boxed{\mathbb{E}[\Delta U^{\text{unif}}] = \frac{\beta \Delta^2}{12} \mathbb{E}_p[\|\nabla U\|^2] + o(\Delta^2).} \tag{16}$$

Empirical Extensivity of Mean Squared Force introduces Atom Count Dependence for Energetic Discretization Error. Recall that the total mean squared force is the sum over atom-wise mean squared forces. If the mean squared force per atom is approximately constant across systems (Extensivity), the total grows proportionally to n. At fixed temperature, the leading mean energy error then scales as

$$\boxed{\mathbb{E}[\Delta U^{\text{mid}}] \propto n\Delta^2,} \tag{17}$$

within the regime where the small-cell expansion remains valid. At sufficiently fine resolution, halving the bin width Δ reduces the expected energy error by approximately a factor of four, while doubling the atom count approximately doubles it under the stated extensive force-scaling assumption.

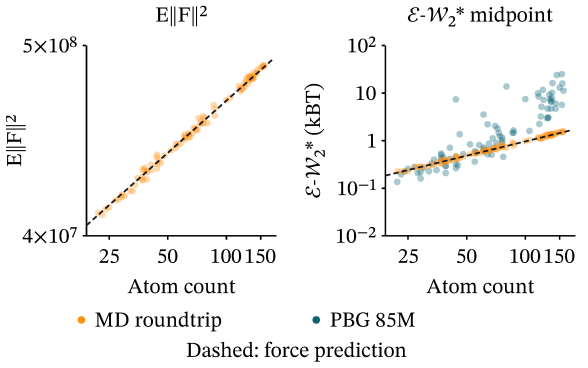


Figure 13: **Force Scaling and Roundtrip Quantization Error.** Total mean-squared force (left) and midpoint $\mathcal{E}$-W_2 (right) versus atom count at $b = 4096$ ($\Delta = 0.8\,\text{pm}$). MD roundtripsa are shown in orange, while petrol denotes PBG 85M ($D/N = 192$) samples. Dashed curves show force-based predictions (Equation 15).

The energetic Wasserstein distance $\mathcal{E}$-W_2^* between the uniformely weighted reconstructed and reference energy distributions is at least their absolute difference in mean energy:

$$\mathcal{E}\text{-}W_2 \geq \beta \left|\mathbb{E}[\Delta U]\right|. \tag{18}$$

To test the mean squared force extensivity assumption and dependent predictions, we inspect reference MD conformers from the *Many Peptides* dataset. In fact, the expected mean square force is well approximated as extensive property within this dataset (Figure 14). We then proceeded to quantize and reconstruct them in a roundtrip, using uniform importance weights to calculate an $\mathcal{E}$-W_2^*. Observed roundtrip energetic error was matched by Equation 18 and scaled with atom count as a power of 1 (Figure 14). In contrast, representative samples drawn from a PBG showed superlinear scaling with atom count, in line with the approximately power of 2 scaling in the empirical scaling model. Overall, these results suggest that parts of the sampling quality scaling observed in PBGs can be attributed to *Quantization Error* alone, which scales linearly with atom count.

In line with Equations 15 and 16, roundtrip samples produced by roundtrips with uniform reconstruction had twice the $\mathcal{E}$-W_2^* as equivalent samples produced via midpoint reconstruction (Figure 14). Relatedly, representative PBG samples showed an advantage for midpoint over uniform reconstruction, too.

Discretization Resolution. To evaluate the accuracy of Equations 15 and 16 along the bin size Δ axis, too, we performed roundtrip experiments at varying resolutions (Figure 15). Finer binning resolution reduces the discretization error. As previously described, while roundtrip experiments show decreasing $\mathcal{E}$-W_2^*, eventual generative model specific aspects limit furhter increases in quantization resolution without changes in the overall box size L or the model capacity.

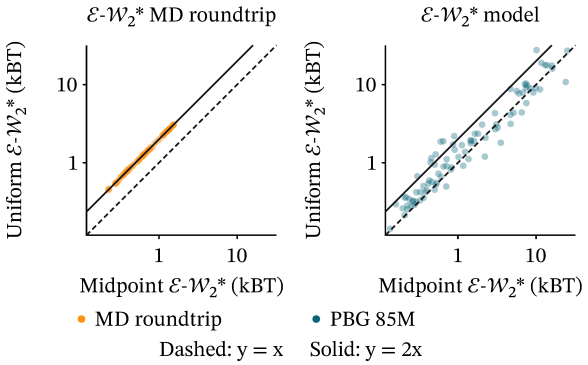


Figure 14: **Midpoint versus Uniform Reconstruction.** Energy $\mathcal{E}$-W_2^* comparisons for MD roundtrips (left) and representative PBG 85M samples (right) at $b = 4096$ ($\Delta = 0.8\,\mathrm{pm}$), with $D/N = 192$ for the generator. Dashed lines mark equality.

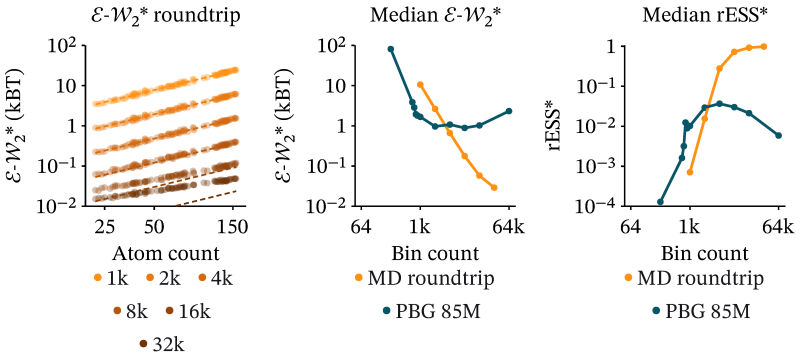


Figure 15: **Quantization Error across Resolutions.** Left: per-system midpoint roundtrip $\mathcal{E}$-W_2 versus atom count, with force-based predictions dashed. Middle and right: median $\mathcal{E}$-W_2 and rESS across systems for midpoint roundtrips (orange) and PBG 85M (petrol), as a function of bin count.

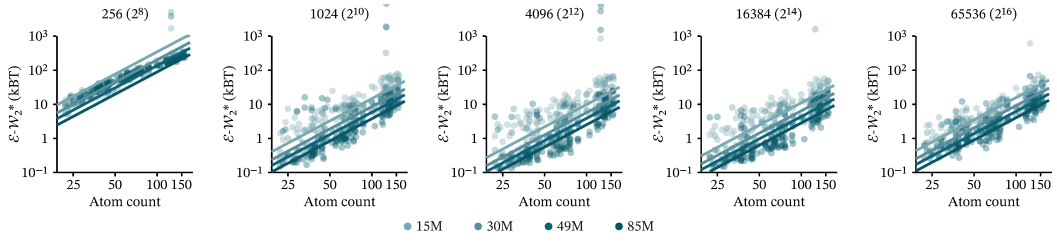


Figure 16: **Energetic Error versus System Size and Quantization Resolution.** Per-system $\mathcal{E}$-W_2^* for PBGs 15M–85M on *Many Peptides*. Solid curves show the fitted scaling law; the display is capped at 10^3.

D.3 Increasing Sampling Rate with Speculative Decoding

Sampling Rate Data. Sampling efficiency can be understood as the product of sampling quality and sampling rate, and measured in effective samples per second. To model the sampling rate of PBGs and investigate the ability to speed up their sampling via speculative decoding (Leviathan et al., 2023), we proceeded to observe their sampling rate at a fixed batch size of 512. We measured autoregressive (AR) sampling for 30M–640M PBG models trained on *Many Peptides*, and speculative decoding (SD) for 85M–640M targets using a 15M drafter at the available proposal strides $K = 1, \ldots, 4$.

Table 7: **Model Architectures and Inference Covariates.** n_{layers} is the layer count, d_{model} the hidden width, d_{ff} the SwiGLU intermediate width, and $H_{\text{Q}}/H_{\text{KV}}$ the query/KV head counts. All heads have width $d_{\text{head}} = 128$. N_{4096} counts executed dense parameters with the 4096-row coordinate output head, excluding normalization scales and embedding lookups. $w_{4096} = 2N_{4096}$ is the BF16 weight-read estimate per forward call, shared across the batch. v_{KV} is the BF16 KV size per cached token and sequence, summed over layers.

Model	n_{layers}	d_{model}	d_{ff}	$H_{\text{Q}}/H_{\text{KV}}$	N_{4096} [10^6]	w_{4096} [MB]	v_{KV} [KiB]
PBG 8M	4	384	1024	6/3	9.830	19.661	6
PBG 15M	4	512	1408	8/4	17.039	34.079	8
PBG 30M	8	512	1408	8/4	31.982	63.963	16
PBG 49M	8	640	1920	10/5	51.773	103.547	20
PBG 85M	10	768	2176	12/6	88.670	177.340	30
PBG 160M	24	768	2112	6/3	162.398	324.796	36
PBG 320M	28	1024	2816	8/4	334.496	668.991	56
PBG 640M	24	1536	4224	12/6	643.301	1286.603	72

Empirical Sampling Rate Model. We estimate batch sampling time using an architecture-based model of the decoding process for both autoregression (AR) and speculative decoding (SD). These estimates depend on model architecture, prompt and output length, and decoding method. For speculative decoding, they also depend on proposal stride and measured acceptance. We fit a shared empirical model:

$$\begin{aligned} T_B &= a + \alpha W + \beta \left(V_1 + V_{\text{write}} + \frac{D_{\text{read}}}{q^\gamma} \right) + \tau(J + \kappa U), \\ R &= B/T_B, \qquad S = T_B^{\text{AR}}/T_B^{\text{SD}}. \end{aligned} \tag{19}$$

Here W estimates cumulative weight reads. V_1, V_{write}, and D_{read} estimate cache traffic from single-token calls, target writes, and packed target reads, respectively. J counts layer calls, and U counts additional target layer-token positions during verification. Traffic is expressed in TB and execution counts in thousands. The call width is $q = 1$ for AR and $q = K + 1$ for SD. For SD, we infer a constant conditional acceptance probability from recorded accepted-to-proposed token ratios for each model.

Model Specification. For each model–peptide–stride setting i, we convert the median rate r_i across five repeats to batch latency $t_i = B/r_i$. We fit independent Gaussian residuals on log latency, with separate residual scales for AR and SD:

$$\log t_i \sim \mathcal{N}\left(\log T_{B,i}, \sigma^2_{m(i)}\right), \qquad m(i) \in \{\text{AR}, \text{SD}\}. \tag{20}$$

Prior scales use seconds for a, seconds per TB for α and β, and seconds per thousand layer calls for τ. The remaining parameters are dimensionless.

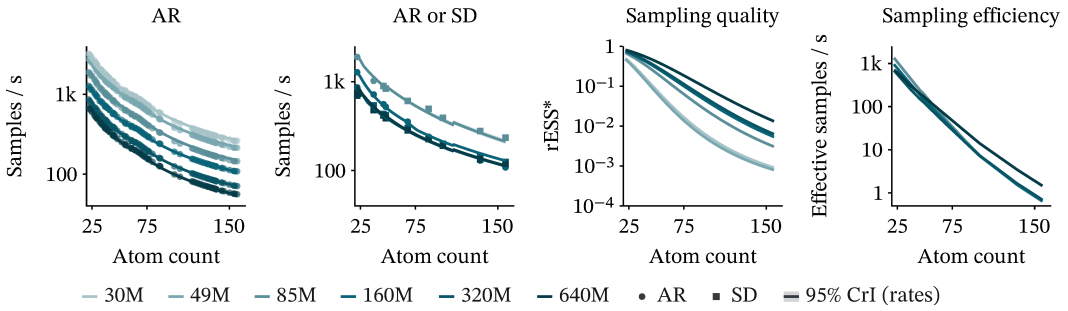


Figure 17: **Sampling Rate, Quality, and Efficiency.** From left to right: Measured autoregressive (AR) sampling rates and model estimates. Highest measured sampling rate using speculative-decoding (SD) strides with a 15M drafter across all strides. Refitted scaling-model estimates of rESS^* for large models that were previously part of the extrapolation holdout. Sampling efficiency of PBGs at optimal speculative decoding.

Parameter	Prior
a	$\mathcal{N}^+(0, 0.1^2)$
$\log \alpha$	$\mathcal{N}(0, 1^2)$
$\log \beta$	$\mathcal{N}(0, 1^2)$
$\log \tau$	$\mathcal{N}(\log 0.1, 1^2)$
$\log \gamma$	$\mathcal{N}(0, 0.75^2)$
$\log \kappa$	$\mathcal{N}(0, 0.75^2)$
σ_{AR}	$\mathcal{N}^+(0, 0.25^2)$
σ_{SD}	$\mathcal{N}^+(0, 0.25^2)$

Rate Model Estimates. The empirical sampling rate model has reported mean absolute percentage errors below 10% for AR rates, SD rates, and speedups (Tables 8 and 9). Given the optimal sampling rates for a given model and peptide (Figure 17, 2. from the left), and a model of sampling quality, the sampling efficiency, expressed as the rESS^*/s, was estimated (Figure 17, 3. and 4. from the left). Given that both sampling quality *and* sampling rate decay with atom count, the sampling efficiency drops from ca. 1000 effective samples per second to just 1 effective sample per second as atom count increases from ca. 25 to 150 atoms. This relationship is opening an opportunity for future work to improve sampling efficiency.

Table 8: **Rate Model Parameters.** Values are posterior medians with 95% CI.

Parameter	Median	95% interval
a	0.0463	[0.0439, 0.0487]
α	2.819	[2.740, 2.898]
β	0.964	[0.955, 0.972]
τ	0.0948	[0.0917, 0.0980]
γ	2.095	[1.550, 3.006]
κ	2.208	[2.075, 2.345]
σ_{AR}	0.0249	[0.0235, 0.0265]
σ_{SD}	0.1160	[0.1029, 0.1322]

Table 9: **Rate Model Fit.** Batch-latency R^2_T and mean absolute percentage errors (MAPE) for rates and speedups.

Metric	AR	SD	Speedup
R^2_T	0.9982	0.9359	—
MAPE (%)	2.02	9.57	9.75

D.4 Solvation Free Energy Estimation

Small Molecule Sampling. A prefix-conditioned Boltzmann generator, PBG-Joint, trained on a heterogenous dataset of peptide and small molecule conformers showed comparable sampling quality on *MDQM9-nc* holdout systems than dedicated models finetuned on this dataset (Table 10). Across both *MDQM9-nc* (Figure 18) and *FreeSolv* (Figure 19) sampling quality on holdout systems is limited and decays with atom count, stereocenters and the number of rotatble bonds.

Table 10: **Transferable Sampling of Small Molecules in MDQM9-nc.** Shown are results from fine-tuned 47M and 85M PBG models and the PBG-Joint 160M model at 300K. Molecules are grouped into Easy ($n \leq 19$ atoms and ≤ 3 rotatable bonds; 42 systems) and Hard (53 systems). Best values are **bold**; second-best are underlined.

	Easy (42 systems)			Hard (53 systems)		
Model	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓
PBG 47M	0.004	3.975	0.652	0.001	5.046	0.952
PBG 85M	**0.018**	3.168	0.596	**0.005**	**4.706**	**0.781**
PBG-Joint 160M	0.006	**2.693**	**0.593**	0.002	4.723	0.823

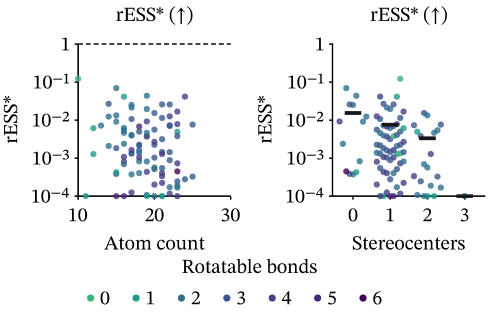


Figure 18: **MDQM9-nc Sampling Quality by Chemical Complexity.** Black bars show group means.

Table 11: **Transferable Sampling of Small Molecules in FreeSolv.** Shown are results from the PBG-Joint 160M model at 300K.

	≤ 12 atoms (31)			13–24 atoms (40)			≥ 25 atoms (23)		
Environment	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓
Vacuum	0.00066	2.278	$0.602^{(30/31)}$	0.00015	9.120	1.125	0.00010	28249.100	1.724
Implicit water	0.00281	0.633	$0.616^{(30/31)}$	0.00038	4.325	0.955	0.00010	16517.314	1.610

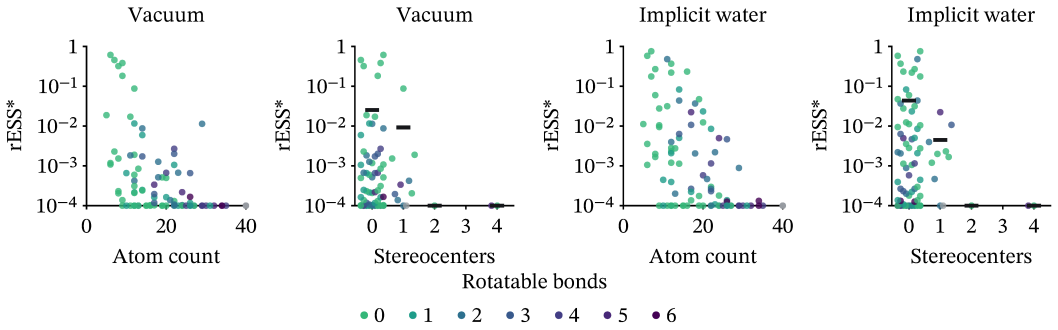


Figure 19: **FreeSolv Sampling Quality by Chemical Complexity.** Black bars show group means.

Peptide Solvation Free Energy. Analogous to the previous results for small molecules, a prefix-conditioned Boltzmann generator, PBG-Joint, trained on a heterogenous dataset of peptide and small molecule conformers showed comparable sampling quality to a dedicated fine-tuned model (Table 12).

In contrast to BAR estimates, forward-only estimation of solvation free energies using the Zwanzig equation does not require MD samples of the system of interest (Figure 20). Estimates of solvation free energy on *Many Peptides* holdout systems correlated with GRAVY hydrophobicity scores (r=0.34), albeit less than comparable BAR estimates.

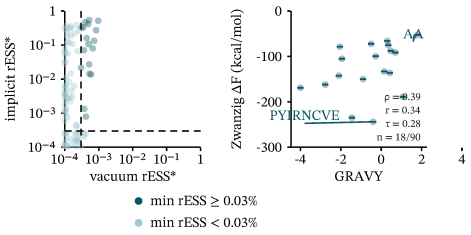


Figure 20: **Peptide Forward Zwanzig Estimates.** Vacuum versus implicit-water forward rESS (left) and unclipped forward ΔF versus GRAVY (right). Correlations require rESS $\geq 0.03\%$ in both states. Free energies are in kcal/mol. Bars show propagated analytical standard errors.

Table 12: **Vacuum Peptide Sampling across Training Conditions.** Implicit-water, vacuum-fine-tuned, and joint models evaluated at 300 K. Superscripts mark partial coverage. Best values are **bold**; second-best are underlined.

		2AA (30 systems)			4AA (30 systems)			8AA (30 systems)		
Model	T	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓	rESS* ↑	$\mathcal{E}$-$\mathcal{W}_2^*$ ↓	Tg-$\mathcal{SW}_2^*$ ↓
PBG 85M (Implicit Water)	300 K	0.00010	52.544	1.512	0.00010	65.228	1.361	0.00010	143.966	1.380
PBG 85M Vacuum	300 K	**0.00020**	7.192	**1.115**	0.00013	**9.628**	1.086	0.00011	**39.858**	1.275
PBG-Joint 160M	300 K	0.00019	**7.107**	1.142	**0.00014**	11.242	**1.068**	**0.00013**$^{(27/30)}$	46.838$^{(27/30)}$	**1.231**$^{(27/30)}$

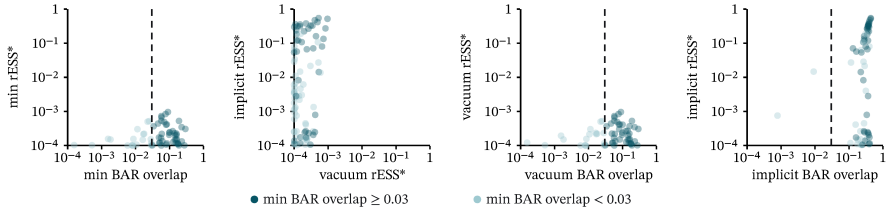


Figure 21: **Peptide Sampling Diagnostics.** Minimum and per-state rESS* and BAR overlap. Dashed lines mark overlap 0.03; values below the 10^{-4} display limits are omitted.

FreeSolv Solvation Free Energy. Zwanzig estimates of Solvation Free Energy for the FreeSolv dataset are identifiable for 32/94 systems and correlate weakly with experimental measurements (r=0.66).

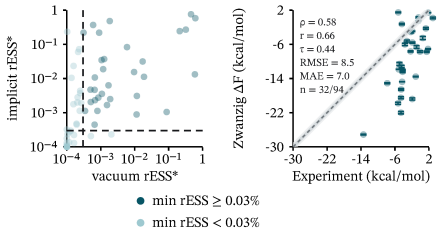


Figure 22: **FreeSolv Forward Zwanzig Estimates.** Vacuum versus implicit-water forward rESS (left) and unclipped forward ΔF versus experiment (right). Correlations require rESS $\geq 0.03\%$ in both states. Free energies are in kcal/mol. Bars show propagated analytical standard errors.

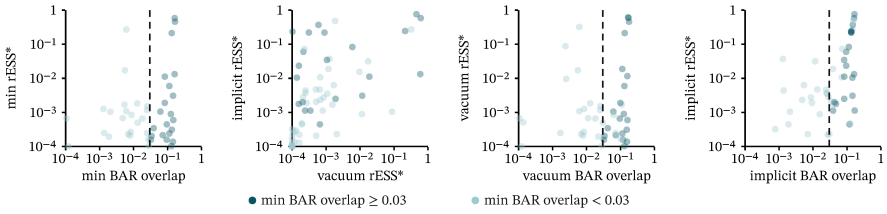


Figure 23: **FreeSolv Sampling Diagnostics.** Minimum and per-state forward rESS^* and BAR overlap. Dashed lines mark overlap 0.03; values below the 10^{-4} display limits are omitted.

D.5 Explicit Microsolvation

Stepwise microsolvation (Figure 24) shows early plateauing of potential energy changes as waters from reference MD runs are added while microsolvation by a finetuned PBG 85M leads to steadily worse energetic changes with eventual positive potential energy contribution.

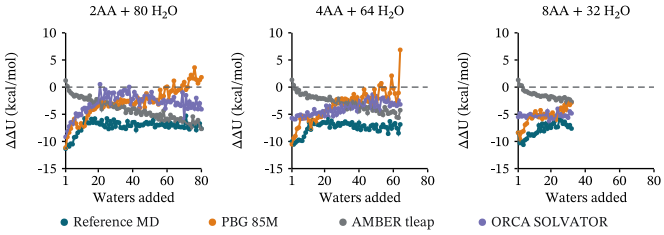


Figure 24: **Energy Change per Added Water.** Median potential energy differences $\Delta U(k) = U(k) - U(0)$ of nearest neighboring water insertion. The dashed line marks zero energy change. MD reference and AMBER tleap waters are ordered by the distance from the water geometric center to the nearest solute atom. ORCA SOLVATOR and PBG waters follow addition order. Energies use the non-periodic Amber14/TIP3P reaction-field potential with a $2\,\mathrm{nm}$ cutoff.

E DATASETS

Many Peptides. The *Many Peptides* dataset was used in its precollated webdataset format. The octapeptide YFPHAGYT was frequently observed to yield strongly insufficient samples with accurate stereochemistry in generated samples. We excluded this peptide as a result, leaving 29 octapeptide test systems in most evals. The paragraph symbol (¶) marks evaluation summaries that are affected.

MDQM9-nc. This dataset from Viguera Diez & Olsson (2024) was collated into a shuffled webdataset for training. The dataset will be released publicly.

Vacuum Peptides. We initialized vacuum simulations from conformers of *Many Peptides*, retaining its train/validation/test assignments. The simulations used Amber ff14SB with constrained bonds to hydrogen and no nonbonded cutoff. The dataset will be released publicly.

Microsolvated Peptides. We placed peptide conformers from *Many Peptides* in periodic TIP3P water boxes with 1.0 nm solvent padding. Simulations used Amber ff14SB, constrained bonds to hydrogen, rigid water, and particle-mesh Ewald electrostatics with a 0.9 nm real-space cutoff. Box dimensions remained fixed. We cropped each saved configuration to the solute and 128 nearby complete water molecules. For training, we ordered the retained waters by the minimum distance from each water oxygen to a solute atom. We retained train/validation/test assignments from *Many Peptides*. The dataset will be released publicly.

FreeSolv. We generated vacuum, implicit-water, and explicit-water ensembles for molecules from FreeSolv. Vacuum simulations used the original FreeSolv GAFF parameters and AM1-BCC charges, with constrained bonds to hydrogen and no nonbonded cutoff. Implicit-water simulations used GAFF2 with Antechamber AM1-BCC charges, OBC1 implicit solvent, a solvent dielectric of 78.5, and a 2.0 nm cutoff. These systems had no bond constraints. Explicit-water simulations used GAFF2/AM1-BCC and TIP3P, with constrained bonds to hydrogen, rigid water, and particle-mesh Ewald electrostatics with a 1.0 nm real-space cutoff. The Lennard–Jones switching function began at 0.9 nm. The microsolvated export retained 128 complete waters per frame, ordered by minimum oxygen-to-solute distance. We retained train/validation/test assignments from Mobley & Guthrie (2014). The dataset will be released publicly.

Many Cycles. We simulated head-to-tail cyclic peptides using Amber ff14SB with OBC1 implicit solvent, a solvent dielectric of 78.5, a 1.0 nm cutoff, and constrained bonds to hydrogen. The exported trajectories used eight temperature states from 300 to 500 K, a 2 fs Langevin timestep, and 10 ps exchange and save intervals. Each exported run contained 100 ns per replica. The dataset will be released publicly.

Table 13: **Overview of Training Datasets.** Tokens count three coordinate tokens per atom per conformer. Atom counts are min / median / max over distinct training systems.

Dataset	Conformers	Molecules	Conf./mol.	Tokens	Min	Median	Max
Many Peptides	4.34×10^8	21,700	2.00×10^4	1.45×10^{11}	17	117	178
MDQM9-nc	1.969×10^8	12,306	1.60×10^4	1.08×10^{10}	3	18	29
Vacuum Peptides	2.140×10^7	11,852	1.81×10^3	7.09×10^9	17	116	176
Solvated Peptides	1.245×10^7	7,340	1.70×10^3	1.85×10^{10}	401	500	559
FreeSolv (vacuum)	6.535×10^6	508	1.29×10^4	3.56×10^8	3	18	44
FreeSolv (implicit water)	6.440×10^6	508	1.27×10^4	3.51×10^8	3	18	44
FreeSolv (microsolvated)	2.158×10^6	262	8.24×10^3	2.60×10^9	389	402	428

F Evaluation Metrics

BAR bridge weight The sample-origin probabilities in (7) define the BAR bridge weights. At the fitted free-energy difference $\widehat{\Delta F}$, each sample receives the probability of belonging to the other state,

$$a_i^A = p_{\widehat{\Delta F}}(B \mid \mathbf{x}_i^A), \qquad a_j^B = 1 - p_{\widehat{\Delta F}}(B \mid \mathbf{x}_j^B). \tag{21}$$

Naturally, a weight approaching 0 indicates confident assignment to the sample's source state. The BAR overlap reported is the smaller mean bridge weight,

$$O_{\mathrm{BAR}} = \min\left(\frac{1}{n_A}\sum_{i=1}^{n_A} a_i^A, \frac{1}{n_B}\sum_{j=1}^{n_B} a_j^B\right). \tag{22}$$

Thus $O_{\mathrm{BAR}} \in [0, 1]$, with small values indicating little sampled overlap and warning of unreliable free-energy estimates. The metric depends on the retained sample counts: identical distributions give $O_{\mathrm{BAR}} = \min(n_A, n_B)/(n_A + n_B)$, which equals $1/2$ for equal counts. For solvation free energies, we compute it on each Zero Free Energy Proposal leg and use the minimum of the vacuum and water values as the overall overlap diagnostic. Klimovich et al. (2015) suggest that entries as low as 0.03 can be tolerable with enough samples and sufficiently small uncertainty, while smaller values warrant caution. We therefore require $O_{\mathrm{BAR}} \geq 0.03$ on both legs for the correlation analysis, while acknowledging that this screening criterion does not guarantee convergence.

Sliced Torus-Wasserstein (← Section 4.1) We consider each conformation's vector $\boldsymbol{\theta} \in [0, 2\pi)^d$ of d dihedral angles. To identify them, we enumerate all bonded four-atom paths whose central bond joins two non-hydrogen atoms, counting a path and its reverse only once. The selection is not restricted to backbone angles or rotatable bonds.

We compare one-dimensional circular projections, called slices. The integer vector $k = e_i$ selects angle i, while $k = e_i \pm e_j$ selects a pairwise sum or difference ($1 \leq i < j \leq d$); e_i is the ith unit vector. Let K be the retained slice pool. For each slice $k \in K$, define its projected coordinate $z_k(\boldsymbol{\theta})$, circle circumference L_k, and normalized slice weight a_k:

$$z_k(\boldsymbol{\theta}) = \frac{(k^\top \boldsymbol{\theta}) \bmod 2\pi}{\|k\|_2}, \qquad L_k = \frac{2\pi}{\|k\|_2}, \qquad a_k = \frac{(1 + \|k\|_2^2)^{-(d/2+2)}}{\sum_{h \in K}(1 + \|h\|_2^2)^{-(d/2+2)}}. \tag{23}$$

Here $\|k\|_2$ is 1 for individual angles and $\sqrt{2}$ for pairs. Let μ_k and ν_k be the generated and reference distributions of z_k. Valid generated conformers receive the normalized importance weights from (2); reference frames receive equal weights. The reported quantity estimates

$$\mathbb{T}\mathrm{g}\text{-}\mathcal{SW}_2^* \approx \left[\sum_{k \in K} a_k \, W_{2,\mathrm{circ}(L_k)}^2(\mu_k, \nu_k)\right]^{1/2}. \tag{24}$$

Circular W_2 is the square root of the least mean squared distance needed to move probability mass between the two distributions, using the shorter path around the circle. Thus the metric is a weighted root-mean-square discrepancy in radians. The slice weights a_k favor individual angles over angle pairs, increasingly so as d grows.

We evaluate all retained individual-angle slices and use up to 256 pair draws for peptides, or 128 for small molecules. Pair draws use replacement, with probabilities proportional to a_k. The candidate pools contain at most 4096 and 1024 slices, respectively, in deterministic order, with individual angles first. Large weighted distributions are approximated by 1024 equally weighted points selected at fixed weighted quantiles.

G METHODS

Replica-exchange simulations. We generated the additional peptide and FreeSolv datasets using replica-exchange molecular dynamics in OpenMM with openmmtools. Briefly, eight replicas covered geometrically spaced temperatures of 300.00, 322.71, 347.14, 373.42, 401.69, 432.10, 464.81, and 500.00 K. We ran Langevin dynamics with 2 fs timestep and a collision rate of $0.3\,\mathrm{ps}^{-1}$. We attempted replica exchanges and saved configurations every 5,000 steps (10 ps). The peptide and FreeSolv runs used 200 production intervals, corresponding to 2 ns per replica per simulation burst. For training, simulation data was collated into pre-shuffled webdatasets.

Architecture We use decoder-only transformers with causal grouped-query attention, pre-layer RMSNorm ($\epsilon = 10^{-6}$), query/key normalization, and SwiGLU feed-forward blocks. Input and output token embeddings were tied. Table 7 lists the architectures used in the scaling and application experiments. We refer to model-size based on non-embedding parameters. The smaller PBGs were randomly initialized.

Encoding Peptides are specified by PEPTIDE HELM strings; small molecules use SMILES embedded in CHEM HELM strings.

At the default resolution, the implementation rounds coordinates to 4096 equally spaced grid points with spacing $0.8\,\mathrm{pm}$ and endpoints $\pm 1.638\,\mathrm{nm}$ on each axis. This endpoint-inclusive implementation uses $b - 1$ intervals between grid points. Evaluation reconstructs the selected grid points deterministically using midpoint dequantization.

Half of the RoPE frequency pairs encode atom indices and half encode token positions. The RoPE base is 10^3. Models also add a fixed sinusoidal coordinate-axis (x, y, or z dimension) encoding with base 10. Conformations are greedily packed into rows of at most 1028 tokens, with attention restricted to each conformation's segment.

Training We minimize teacher-forced next-token cross-entropy on coordinate tokens and the terminator, excluding prompt and padding tokens. Thus a complete n-atom conformation contributes $3n + 1$ loss tokens when a terminator is present, and $3n$ otherwise. We report D as delivered loss tokens, rather than padded sequence capacity or the nominal checkpoint name. Dataset inventories are given in Table 13.

The dedicated 85M vacuum and microsolvation models continue from the pre-trained *Many Peptides* checkpoint on their respective 300–500 K datasets and exhausted all data within these datasets, respectively. For microsolvation, only complete water molecules that fit within the context are retained.

Optimizer, Learning Rate, Scheduler We used Muon for non-embedding weight matrices and AdamW for embeddings and remaining parameters (Jordan et al., 2024; Liu et al., 2025). Muon uses Nesterov momentum 0.95 and five Newton–Schulz iterations. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 10^{-8}$. Generally training runs use a peak learning rate of 10^{-3} and decoupled weight decay 0.1. Gradient clipping was initially performed with an initial global-norm limit of 10, then a per-tensor relative limit of 1.5 times its moving-average norm (decay 0.99) after 100 steps.

Warmup–stable–decay schedules (Hägele et al., 2024) permit several training budgets to share a stable-phase trajectory. A cooldown typically added 20% of the preceding token budget and reduced the learning rate to zero with a $1 - \sqrt{t}$ profile, where $t \in [0, 1]$ is cooldown progress.

Sampling and Evaluation We draw 10^4 proposals per test system at logit temperature 1.0 with KV-cached autoregressive decoding and deterministic grid-point reconstruction. During evaluation we correct a fully mirrored stereochemical assignment by global reflection of the conformer, while retaining the initially assigned likelihood. Remaining invalid stereoisomers are excluded (usually <3% of samples). Importance weights use the target reduced potential and proposal likelihood. Starred metrics retain unclipped weights.

The fixed-solute microsolvation comparison uses the same 512 reference solute conformers per system for all methods and a non-periodic Amber14/TIP3P energy with a $2\,\mathrm{nm}$ reaction-field cutoff.

Bayesian Inference Bayesian inference was performed with NumPyro NUTS using four chains, 2000 warmup steps and 2000 retained draws per chain. The target acceptance was 0.99, and maximum tree depth was set to 12.

Table 14: **Selected model training configurations.** Recorded allocations for the runs producing the PBG checkpoints in Table 2. Batch sizes count packed rows of at most 1028 tokens, not conformers. Each MI250X GPU exposes two compute devices, while each NVIDIA GPU exposes one.

Model	GPU type	Nodes	GPUs	Devices	Batch/device	Global batch	Loss tokens /step (k)
PBG 30M	RTX PRO 6000 Blackwell	1	2	2	128	256	182.3
PBG 49M	RTX PRO 6000 Blackwell	1	2	2	112	224	159.6
PBG 85M	RTX PRO 6000 Blackwell	1	2	2	80	160	114.2
PBG 160M	H100 NVL	1	2	2	64	128	91.5
PBG 320M	H100 NVL	1	2	2	32	64	45.8
PBG 640M	MI250X	4	16	32	16	512	365.2
PBG-Joint 160M	H100 NVL	1	2	2	64	128	91.2

Hardware The training implementation uses JAX with bfloat16 forward computation and float32 master parameters. Experiments use NVIDIA GPU clusters and AMD MI250X accelerators on LUMI-G; hardware and batch allocation vary by run. Training is parallelized using a simple distributed data parallel approach. The reported autoregressive and speculative-decoding throughput comparison uses an H100 GPU NVL at batch size 512.