CONFORMFLOW: SCALABLE NORMALIZING FLOW FOR PROTEIN CONFORMATIONAL ENSEMBLE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular dynamics (MD) simulations remain the standard tool for characterizing protein conformational landscapes, but their high computational cost limits large-scale and long-timescale applications. Recent generative models, especially diffusion-based approaches, provide promising alternatives by learning equilibrium conformational distributions across diverse protein systems. We present **ConformFlow**, the first scalable normalizing-flow framework for sequence-conditioned protein conformational ensemble generation. ConformFlow combines a continuous backbone latent representation with a RealNVP-style flow parameterized by sequence-aware Transformer coupling networks, enabling exact likelihood training, single-step sampling, and plug-and-play conditioning on flexible geometric constraints. Across diverse protein systems, ConformFlow generates ensembles that agree well with reference MD simulations, generalizes to proteins beyond its training data, and achieves substantially faster sampling than diffusion-based baselines. These results establish ConformFlow as an efficient and controllable alternative for protein conformational ensemble generation.

1 INTRODUCTION

Proteins are essential molecular components of living systems, carrying out diverse functions such as catalysis, signaling, and transport. These functions are governed not only by static structures, but also by conformational flexibility and dynamic transitions. Characterizing protein conformational ensembles is therefore critical for understanding the molecular mechanisms underlying protein function Whisstock & Lesk (2003). Molecular dynamics (MD) simulations provide a principled approach for studying biomolecular processes such as protein folding and unfolding Shaw et al. (2010); Robustelli et al. (2018). However, MD remains computationally demanding, making exhaustive simulation of long-timescale protein dynamics computationally prohibitive.

Recent progress in deep generative modeling has opened new opportunities for protein conformational modeling. A growing body of work has developed generative approaches for sampling equilibrium ensembles Lu et al. (2023); Lewis et al. (2025); Jing et al. (2024a); Noé et al. (2019); Zheng et al. (2024); Wayment-Steele et al. (2024); Jing et al. (2024b); Raja et al. (2025); Lelièvre et al. (2023); Du et al. (2024); Fu et al. (2022); Arts et al. (2023). These methods offer a promising route to studying protein–protein interactions, allosteric regulation, biomolecular condensation, and conformational heterogeneity Brandsdal & Smalås (2000); Guo & Zhou (2016). Most recent scalable approaches are based on diffusion models, which are flexible, expressive, and have shown promising generalization across diverse protein systems. However, diffusion models require iterative denoising at inference time, making sampling less efficient. Moreover, because exact likelihood evaluation is generally unavailable for diffusion models, incorporating likelihood-based physical constraints or posterior guidance is less direct.

Motivated by these opportunities and limitations, we introduce **ConformFlow**, a Normalizing-Flow framework for protein Conformational ensemble generation. To our knowledge, ConformFlow is the first scalable normalizing-flow framework for sequence-conditioned protein conformational ensemble generation. It combines a learned structural latent representation with a scalable RealNVP-style generative model parameterized by masked Transformer coupling networks. The model is trained on large-scale single-structure datasets together with equilibrium MD trajectories, allowing it to learn transferable distributions over protein conformational space. Unlike diffusion-based models, Con-

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.

formFlow provides single-step sampling and exact likelihood evaluation. These properties make it computationally efficient and naturally compatible with inference-time guidance under flexible geometric constraints.

Our contributions are fivefold:

1. **A scalable architecture for sequence-conditioned normalizing flow.** We introduce a latent-space RealNVP architecture with residue-masked affine coupling layers parameterized by bidirectional Transformer as backbones, enabling expressive sequence-conditioned density modeling while preserving parallel sampling and exact likelihood evaluation.
2. **Large-scale transferable training.** We train ConformFlow on hundreds of thousands of protein sequences and millions of conformations from single-structure databases and equilibrium MD trajectories, enabling ensemble generation across diverse protein families.
3. **Fast and controllable inference.** ConformFlow enables one-step generation with exact likelihood evaluation, yielding substantially faster sampling than iterative diffusion models and supporting plug-and-play guidance with flexible, user-specified geometric constraints.
4. **Scaling behavior of flow depth.** We show that ConformFlow benefits substantially from increasing the number of coupling blocks, improving ensemble fidelity on training proteins while also enhancing generalization to held-out protein systems.
5. **Ablation study of NF architecture choices.** We systematically ablate coordinate representation and coupling-mask design, showing that the learned latent representation and odd–even residue coupling are both important for accurate and transferable sequence-conditioned protein ensemble generation.

2 Background

2.1 Normalizing Flows and RealNVP

Normalizing flows are likelihood-based generative models for continuous random variables. Let $x \in \mathbb{R}^D$ denote a data sample drawn from an unknown data distribution p_{data}. A normalizing flow defines an invertible and differentiable transformation

$$z = f_\theta(x), \qquad x = f_\theta^{-1}(z), \tag{1}$$

where z is mapped to a simple base distribution $p_0(z)$, typically a standard Gaussian $\mathcal{N}(0, I)$. By the change-of-variables formula, the model density is

$$p_\theta(x) = p_0(f_\theta(x)) \left| \det \frac{\partial f_\theta(x)}{\partial x} \right|. \tag{2}$$

Maximum-likelihood training therefore minimizes the negative log-likelihood

$$\mathcal{L}_{\text{NF}}(\theta) = \mathbb{E}_{x \sim p_{\text{data}}} \left[-\log p_0(f_\theta(x)) - \log \left| \det \frac{\partial f_\theta(x)}{\partial x} \right| \right]. \tag{3}$$

When $p_0(z) = \mathcal{N}(0, I)$, this objective can be written, up to an additive constant, as

$$\mathcal{L}_{\text{NF}}(\theta) = \mathbb{E}_{x \sim p_{\text{data}}} \left[\frac{1}{2} \| f_\theta(x) \|_2^2 - \log \left| \det \frac{\partial f_\theta(x)}{\partial x} \right| \right]. \tag{4}$$

The first term encourages data samples to be mapped to high-probability regions of the base distribution, while the second term accounts for the local volume change induced by the transformation. After training, generation is performed exactly by sampling $z \sim p_0(z)$ and applying the inverse map $x = f_\theta^{-1}(z)$.

A key challenge in normalizing flows is designing transformations that are both expressive and have tractable Jacobian determinants. RealNVP Dinh et al. (2016) addresses this using affine coupling layers. Given an input x, split it into two parts x_a and x_b. A coupling layer keeps one part unchanged and affinely transforms the other:

$$z_a = x_a, \tag{5}$$

$$z_b = x_b \odot \exp(s_\theta(x_a)) + t_\theta(x_a), \tag{6}$$

where s_θ and t_θ are neural networks producing scale and translation parameters. The inverse transformation is analytic:

$$x_a = z_a, \tag{7}$$
$$x_b = (z_b - t_\theta(z_a)) \odot \exp(-s_\theta(z_a)). \tag{8}$$

Because the Jacobian of this transformation is triangular, its log-determinant is cheap to compute:

$$\log \left| \det \frac{\partial z}{\partial x} \right| = \sum_j s_\theta(x_a)_j. \tag{9}$$

Stacking multiple coupling layers with alternating masks or permutations allows all dimensions to be transformed while preserving exact likelihood evaluation and exact sampling.

Historically, normalizing flows have faced a tradeoff between expressivity and tractable Jacobians. Coupling-based models such as RealNVP are efficient, but each layer only transforms part of the input, which can limit expressivity. Autoregressive flows Papamakarios et al. (2017) improve expressivity by allowing each dimension or block to depend on previous ones while preserving a triangular Jacobian. Recent models such as TARFLOW Zhai et al. (2024) scale this idea with causal Transformer blocks. However, inverse sampling becomes sequential along the autoregressive order, making generation slower than parallel coupling-based flows.

2.2 Deep Generative Modeling for Protein Conformational Ensembles

Deep generative models have recently been applied to protein conformational ensemble generation by learning sequence-conditioned distributions over structures from structural databases or equilibrium MD trajectories. Early likelihood-based approaches used normalizing flows to model equilibrium molecular distributions Kohler et al. (2023). These methods were mainly demonstrated on individual molecular systems and did not establish scalability or transferability across diverse protein sequences. Most recent scalable ensemble generators instead adopt diffusion- or score-based formulations Lu et al. (2023); Lewis et al. (2025); Jing et al. (2024a); Noé et al. (2019); Zheng et al. (2024); Wayment-Steele et al. (2024). These models have shown strong performance by learning transferable conformational distributions across broad protein families. At the same time, the lack of exact likelihood has motivated renewed interest in modern normalizing-flow architectures for transferable learning across molecular systems Tan et al. (2026); Klein & Noé (2024). However, existing NF-based approaches remain primarily focused on peptides and small proteins, and have not yet demonstrated large-scale sequence-conditioned protein ensemble generation.

2.3 Protein Conformational Representations

Recent advances have shown that protein conformations can be effectively represented through learned structural representations, ranging from residue-level discrete tokenizers to continuous coordinate encoders. For example, ESM3 introduces a structure tokenizer that maps a protein conformation to a discrete token sequence. In parallel, SimpleFold uses continuous atom-level coordinate encoders within a flow-matching framework Wang et al. (2025). These representations provide compact alternatives to directly modeling raw three-dimensional coordinates and can serve as intermediate spaces for generative modeling. In ConformFlow, we adopt a continuous structural latent space, allowing the NF model to perform exact likelihood-based density estimation in a latent representation while retaining a decoder back to protein coordinates.

3 Method

In this section, we provide a high-level overview of the ConformFlow architecture. ConformFlow consists of two components. First, a structure autoencoder maps backbone Cartesian coordinates into a compact continuous latent representation. Second, a sequence-conditioned normalizing flow models the distribution over this latent space using RealNVP-style coupling layers parameterized by modern masked Transformer networks. An overview is shown in Fig. 1.

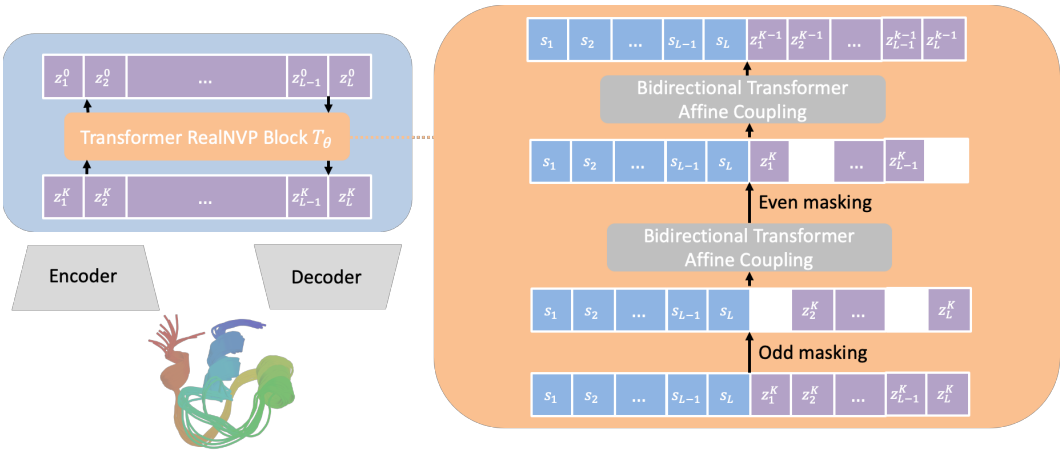

Figure 1: Overview of **ConformFlow**. ConformFlow (1) maps backbone Cartesian coordinates $\mathbf{x} \in \mathbb{R}^{L\times 3\times 3}$, consisting of N, C$\alpha$, and C atoms, into a compact continuous latent representation $\mathbf{y} = z^K \in \mathbb{R}^{L\times d_y}$ using an autoencoder model. (2) A sequence-conditioned RealNVP then learns the latent ensemble distribution $p_\theta(\mathbf{y} \mid s)$ through a stack of invertible affine coupling blocks. During sampling, Gaussian noise is transformed through the invertible flow into latent conformations, which are then decoded back to backbone coordinates. Each coupling block keeps part of the latent variables fixed under an odd–even masking scheme and updates the remaining variables using bidirectional transformer coupling networks conditioned on the amino-acid sequence s.

3.1 STRUCTURE AUTOENCODER

Let $\mathbf{x} \in \mathbb{R}^{L\times 3\times 3}$ denote the backbone coordinates of a protein with L residues, where each residue contains the atoms $\{N, C_\alpha, C\}$. We encode each conformation into a continuous per-residue latent representation

$$\mathbf{y} = \mathrm{Enc}_\phi(\mathbf{x}) \in \mathbb{R}^{L\times M}, \tag{10}$$

where M is the latent dimension per residue.

Our autoencoder is adapted from the ESM3 structure tokenizer. The original ESM3 tokenizer maps backbone coordinates to discrete structure tokens through a vector-quantized bottleneck. In contrast, we replace the vector quantizer with a continuous bottleneck. Specifically, the frozen ESM3 structure encoder produces per-residue features in $\mathbb{R}^{1280}$, which are mapped by a trainable adapter into $\mathbb{R}^M$. The decoder applies a second trainable adapter from $\mathbb{R}^M$ back to $\mathbb{R}^{1280}$, followed by the frozen ESM3 structure decoder:

$$\hat{\mathbf{x}} = \mathrm{Dec}_\psi(\mathbf{y}). \tag{11}$$

Thus, the pretrained geometric encoder and decoder are kept fixed, while only the lightweight bottleneck adapters are trained. This design preserves the geometric inductive biases of ESM3 while producing a continuous latent space suitable for likelihood-based modeling. Additionally, because the continuous latent representation is derived from the ESM3 structure tokenizer, it inherits invariance to global SE(3) transformations of the input structure.

We train the autoencoder to reconstruct backbone geometry using the first four geometric Stage-1 objectives of ESM3:

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{bbdist}} + \mathcal{L}_{\mathrm{bbdir}} + \mathcal{L}_{\mathrm{distbin}} + \mathcal{L}_{\mathrm{dirbin}}. \tag{12}$$

Here, $\mathcal{L}_{\mathrm{bbdist}}$ preserves pairwise backbone-atom distances, $\mathcal{L}_{\mathrm{bbdir}}$ preserves relative backbone directions, $\mathcal{L}_{\mathrm{distbin}}$ predicts discretized C_β–C_β distance bins, and $\mathcal{L}_{\mathrm{dirbin}}$ predicts discretized directional bins. Together, these losses encourage the continuous latent representation to preserve both metric and orientational information needed for accurate backbone reconstruction.

3.2 SEQUENCE-CONDITIONED LATENT NORMALIZING FLOW

Once the autoencoder is trained, we freeze it and learn a sequence-conditioned density over the continuous latent representation,

$$p_\theta(\mathbf{y} \mid \mathbf{s}), \qquad \mathbf{y} \in \mathbb{R}^{L\times M}, \quad \mathbf{s} \in \{1,\dots,20\}^L, \tag{13}$$

where L is the number of residues, M is the latent dimension per residue, and $\mathbf{s}$ is the amino-acid sequence. The flow maps a standard Gaussian base variable to a protein latent:

$$\mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{L\times M}), \qquad \mathbf{y} = T_\theta(\mathbf{z}^{(0)}; \mathbf{s}). \tag{14}$$

The transformation T_θ is a stack of K invertible blocks,

$$T_\theta = T^{(K)} \circ \cdots \circ T^{(1)}, \tag{15}$$

with intermediate states

$$\mathbf{z}^{(b)} = T^{(b)}(\mathbf{z}^{(b-1)}; \mathbf{s}), \qquad b = 1,\dots,K, \qquad \mathbf{z}^{(K)} = \mathbf{y}. \tag{16}$$

Residue-masked affine coupling.

Each block $T^{(b)}$ contains two affine coupling layers. The first updates the odd-indexed residues while keeping the even-indexed residues fixed; the second updates the even-indexed residues while keeping the odd-indexed residues fixed:

$$T^{(b)} = C^{(b)}_{\text{even}} \circ C^{(b)}_{\text{odd}}. \tag{17}$$

Thus, within each block, every residue latent vector is updated once. Stacking multiple blocks yields a deep sequence-conditioned bijection over the full protein latent representation.

For a generic coupling layer C_m, let $\Omega_m \subset \{1,\dots,L\}$ denote the active residues to be updated and let

$$A_m = \{1,\dots,L\} \setminus \Omega_m \tag{18}$$

denote the frozen residues. Given the current state $\mathbf{z} \in \mathbb{R}^{L\times M}$, the coupling network observes only the frozen residue latents $\mathbf{z}_{A_m,:}$, together with the full sequence embedding $\mathbf{e}(\mathbf{s})$. It predicts scale and translation fields for the active residues:

$$(\boldsymbol{\sigma}^m_\theta, \boldsymbol{\mu}^m_\theta) = g^m_\theta\left(\mathbf{z}_{A_m,:}, \mathbf{e}(\mathbf{s})\right), \qquad \boldsymbol{\sigma}^m_\theta, \boldsymbol{\mu}^m_\theta \in \mathbb{R}^{|\Omega_m|\times M}. \tag{19}$$

The active residue latents are then transformed by a per-residue affine map:

$$\mathbf{z}'_{i,c} = \begin{cases} \mathbf{z}_{i,c} \exp\left(\boldsymbol{\sigma}^m_{\theta,i,c}\right) + \boldsymbol{\mu}^m_{\theta,i,c}, & i \in \Omega_m, \\ \mathbf{z}_{i,c}, & i \in A_m. \end{cases} \tag{20}$$

The conditioner g^m_θ is implemented as a bidirectional masked Transformer coupling network. It attends bidirectionally across residue positions, using the full amino-acid sequence embedding and only the frozen residue latents as input. The active residue latents are masked from the conditioner, preserving the triangular Jacobian required for exact likelihood computation. The final output head predicts the scale and translation fields jointly.

Exact likelihood training.

Because each coupling layer transforms only the active residues using parameters computed from the frozen residues, the Jacobian of Eq. equation 20 is block triangular. Its log-determinant is

$$\log|\det J_{C_m}| = \sum_{i\in\Omega_m} \sum_{c=1}^{M} \boldsymbol{\sigma}^m_{\theta,i,c}. \tag{21}$$

The full-flow log-determinant is the sum over all coupling layers and blocks.

We train the flow by exact maximum likelihood on latents by change of variables,

$$\log p_\theta(\mathbf{y} \mid \mathbf{s}) = \log p_{\mathcal{N}}(\mathbf{z}^{(0)}) + \log\left|\det J_{T^{-1}_\theta}(\mathbf{y}; \mathbf{s})\right|. \tag{22}$$

Sampling.

To generate an ensemble for a sequence $\mathbf{s}$, we sample

$$\mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{L\times M}), \tag{23}$$

apply the forward flow

$$\mathbf{y} = T_\theta(\mathbf{z}^{(0)}; \mathbf{s}), \tag{24}$$

and decode the latent conformation to backbone coordinates:

$$\hat{\mathbf{x}} = \mathrm{Dec}_\psi(\mathbf{y}). \tag{25}$$

3.3 Plug-and-play conditional generation

ConformFlow supports inference-time control by combining the learned sequence-conditioned density with differentiable geometric constraints. Let $f : \mathbb{R}^{L\times 3\times 3} \to \mathbb{R}$ be a user-specified differentiable observable, such as radius of gyration or pairwise distance, and let v be a target value. We define a soft-constrained posterior over latent conformations:

$$\pi_\sigma(\mathbf{y} \mid \mathbf{s}, v) \propto p_\theta(\mathbf{y} \mid \mathbf{s}) \exp\left(-\frac{(f(\mathrm{Dec}_\psi(\mathbf{y})) - v)^2}{2\sigma^2}\right), \tag{26}$$

where σ controls the constraint strength.

We sample from this posterior using unadjusted Langevin dynamics. In latent space, the update is

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \eta\nabla_{\mathbf{y}} \log \pi_\sigma(\mathbf{y}_t \mid \mathbf{s}, v) + \sqrt{2\eta T}\,\boldsymbol{\xi}_t, \qquad \boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{27}$$

where η is the step size and T is an optional sampling temperature.

Alternatively, we can sample in the Gaussian base space by writing

$$\mathbf{y} = T_\theta(\mathbf{z}; \mathbf{s}). \tag{28}$$

This defines the corresponding target

$$\pi_\sigma(\mathbf{z} \mid \mathbf{s}, v) \propto p_{\mathcal{N}}(\mathbf{z}) \exp\left(-\frac{(f(\mathrm{Dec}_\psi(T_\theta(\mathbf{z}; \mathbf{s}))) - v)^2}{2\sigma^2}\right). \tag{29}$$

In both parameterizations, f only needs to be differentiable, so new geometric constraints can be introduced at inference time without modifying or retraining the model.

4 Experiment

We evaluate ConformFlow on protein conformational ensemble generation across diverse protein systems. Our results demonstrate that ConformFlow accurately recovers equilibrium distributions from reference MD simulations, generalizes to unseen proteins systems, and enables substantially faster sampling. We further show that the learned likelihood-based model supports plug-and-play conditional generation under differentiable geometric constraints without retraining. Finally, we demonstrate comprehensive ablation studies on scaling behaviors and key architecture design.

4.1 Experimental setup

Data. For training the continuous structure autoencoder, we use the Swiss-Prot subset of the AlphaFold Database, containing 542,378 protein predicted native structures. This dataset provides broad coverage of protein structures and is used to learn latent representations of geometry.

For training the sequence-conditioned normalizing flow, we combine single-structure data with equilibrium MD trajectories. The single-structure component uses the Swiss-Prot subset of the AlphaFold Database. The MD component combines mdCATH Mirarchi et al. (2024) with the BioEmu simulation corpus Lewis et al. (2025), including CATH1, CATH2, Octapeptides, and MEGAsim Charron et al. (2025); Sillitoe et al. (2021); Tsuboyama et al. (2023). ATLAS Vander Meersche et al. (2024) is held out and used only as test dataset.

Table 1: JS divergence against MD and wall-time on test proteins from each dataset.

	CATH1 ($N = 50$)		**Octapeptide** ($N = 50$)		**ATLAS** ($N = 40$)	
Quantity	BioEmu	ConformFlow	BioEmu	ConformFlow	BioEmu	ConformFlow
End-to-end	0.062 (0.006)	**0.043 (0.001)**	0.039 (0.000)	**0.035 (0.000)**	0.291 (0.005)	**0.255 (0.006)**
R_g	**0.083 (0.007)**	0.085 (0.004)	0.038 (0.000)	**0.032 (0.000)**	0.323 (0.019)	**0.305 (0.010)**
RMSD	0.124 (0.012)	**0.058 (0.003)**	0.038 (0.000)	**0.033 (0.000)**	0.366 (0.032)	**0.330 (0.023)**
2D TICA	0.182 (0.011)	**0.132 (0.003)**	0.336 (0.005)	**0.329 (0.004)**	0.434 (0.001)	**0.403 (0.000)**
Time (s)	451.9	**10.7**	71.1	**4.7**	2181.1	**25.8**

Model details. ConformFlow is trained on precomputed continuous latents, with the structure autoencoder frozen during flow training. We use latent dimension $M = 4$ per residue and a 12-block RealNVP model with Transformer coupling networks. Additional training details are provided in Appendix B.1.

Baselines. Our main baseline is BioEmu Lewis et al. (2025), a recent state-of-the-art model for transferable protein conformational ensemble generation trained at comparable scale. Although several diffusion- and flow-matching-based methods have been proposed for ensemble generation, BioEmu is among the few models trained on large-scale, high-quality MD simulation data and is therefore the most direct comparison for our setting.

We also compare against normalizing-flow-based baselines in the ablation study. While prior NF-based molecular ensemble generators have demonstrated promising results, they have primarily focused on small molecules, peptides, or small protein systems. To evaluate the effect of NF architecture design under our large-scale training setting, we adapt PROSE Tan et al. (2026), a TARFlow-based architecture, and retain its major architectural components while training it on our dataset. This provides a controlled NF baseline for assessing the impact of the learned latent representation and coupling-mask design used in ConformFlow.

4.2 Results

Distributional similarity. We evaluate ensemble quality by comparing generated conformational distributions against reference MD simulations. We report Jensen–Shannon divergence (JSD) between reference and generated ensembles projected onto four collective variables: end-to-end distance (EED), radius of gyration (R_g), RMSD to the native structure, and the top two TICA components computed from C_α pseudo-torsion angles Schultze & Grubmuuller (2021).

We evaluate three settings: CATH1, a seen-protein benchmark with 50 proteins; held-out octapeptides, an unseen sequence benchmark in a length-matched regime; and ATLAS, an unseen-protein benchmark outside the training corpus Vander Meersche et al. (2024). Table 1 summarizes the results. On CATH1, ConformFlow improves EED, RMSD, and 2D TICA while matching BioEmu on R_g. On held-out octapeptides, ConformFlow achieves consistently lower JSD across all reported collective variables, demonstrating generalization to unseen sequences in the peptide regime. On ATLAS, ConformFlow also improves all metrics despite the dataset being fully held out from training, indicating transferability to external protein simulation data. Overall, these results show that ConformFlow captures both low-dimensional geometric observables and slow collective modes across seen, held-out, and external test settings. We provide visual free-energy comparisons on representative CATH1 systems in Appendix Figures 2 and 3. ConformFlow recovers the main low-free-energy regions observed in reference MD and closely tracks the MD landscapes.

Plug-and-play conditional generation. One important advantage of diffusion-based generative models is their flexibility for inference-time guidance Ho & Salimans (2022); Chung et al. (2022). Prior work has shown that diffusion-based ensemble generators can be guided by physical forces Wang et al. (2024) and experimental constraints Liu et al. (2025), enabling conditional generation without retraining. In this study, we evaluate whether ConformFlow can be controlled at inference time without retraining. We use RMSD to the native structure as a differentiable constraint observable and draw target values from the empirical MD RMSD distribution. We compare ULA guidance

Table 2: Effect of plug-and-play RMSD guidance on ConformFlow ensembles, evaluated on ATLAS dataset. Guidance targets are drawn from the MD empirical RMSD distribution, so the RMSD JSD measures constraint satisfaction for the guided variants.

	ConformFlow	ConformFlow ($\mathbf{y}$-guided)	ConformFlow ($\mathbf{z}$-guided)
End-to-end	0.255 (0.006)	0.209 (0.008)	**0.195 (0.003)**
R_g	0.305 (0.010)	0.177 (0.011)	**0.145 (0.008)**
RMSD	0.330 (0.023)	0.061 (0.009)	**0.043 (0.007)**
2D TICA	0.403 (0.000)	0.348 (0.002)	**0.312 (0.004)**

Table 3: JS divergence against MD on test proteins from each dataset for different numbers of coupling blocks.

Dataset	Quantity	$K = 4$	$K = 8$	$K = 12$
CATH1	End-to-end	0.071 (0.004)	0.052 (0.002)	**0.043 (0.001)**
	R_g	0.112 (0.008)	0.094 (0.005)	**0.085 (0.004)**
	RMSD	0.097 (0.007)	0.071 (0.004)	**0.058 (0.003)**
	2D TICA	0.196 (0.008)	0.143 (0.005)	**0.132 (0.003)**
Octapeptide	End-to-end	0.189 (0.005)	0.087 (0.001)	**0.035 (0.000)**
	R_g	0.154 (0.008)	0.102 (0.004)	**0.032 (0.000)**
	RMSD	0.284 (0.002)	0.129 (0.003)	**0.033 (0.000)**
	2D TICA	0.491 (0.009)	0.375 (0.006)	**0.329 (0.004)**
ATLAS	End-to-end	0.433 (0.014)	0.356 (0.010)	**0.255 (0.006)**
	R_g	0.412 (0.026)	0.437 (0.017)	**0.305 (0.010)**
	RMSD	0.553 (0.044)	0.376 (0.031)	**0.330 (0.023)**
	2D TICA	0.603 (0.009)	0.542 (0.002)	**0.403 (0.000)**

in the learned latent space $\mathbf{y}$ and in the Gaussian base space $\mathbf{z}$, denoted ConformFlow ($\mathbf{y}$-guided) and ConformFlow ($\mathbf{z}$-guided), respectively. Implementation details are provided in Appendix B.2.

Table 2 reports results on ATLAS. Both guided samplers improve agreement with reference MD across all evaluated collective variables, showing that knowledge of low-dimensional features can enhance ensemble generation quality. The $\mathbf{z}$-guided sampler performs best overall. Intuitively, applying Langevin updates in the gaussian base space avoids sampling on a rugged free energy surface and leads to smoother exploration. This is also a practical advantage of ConformFlow over autoregressive flows: Gaussian-space guidance would require mapping each updated base-space sample back to the latent space at every step, which is sequential and expensive for autoregressive models. In contrast, ConformFlow's RealNVP-style map is parallel, making $\mathbf{z}$-space guidance practical for long protein sequences.

Scaling behavior. We study how ConformFlow performance changes as the number of affine coupling blocks increases. Table 3 reports JSD against reference MD for models with 4, 8, and 12 coupling blocks. Increasing flow depth consistently improves distributional similarity across datasets and collective variables. The gains are especially pronounced on held-out octapeptides and ATLAS, suggesting that deeper coupling stacks improve not only fitting of training-related protein ensembles, but also transfer to unseen protein systems.

Ablation studies on NF architecture. We conduct comprehensive ablation on the role of the latent representation, normalizing flow architectures, and coupling-mask design in Table 4. We compare four variants: TARFlow trained directly on Cartesian backbone coordinates (PROSE baseline), TARFlow trained on the learned continuous latent space, a ConformFlow variant with a four-mask coupling schedule (odd–even–first half–second half), and the final ConformFlow model with stacked odd–even affine coupling blocks. Additional ablation studies details are provided in Appendix B.3.

Cartesian TARFlow performs poorly across all datasets, suggesting that direct likelihood modeling in backbone Cartesian space is challenging for simple coordinate-wise affine flows. Moving TARFlow to the learned latent space substantially improves performance on CATH1, but generalization to held-out octapeptides and ATLAS remains limited. The four-mask ConformFlow variant further improves transfer performance, while the final odd–even ConformFlow architecture achieves

Table 4: Ablation study of coordinate representation and coupling-mask design. We compare flow modeling directly in Cartesian space (PROSE baseline), TARFlow in the learned latent space, a latent ConformFlow variant with a four-mask coupling schedule (odd–even–first half–second half), and the final ConformFlow architecture with odd–even coupling blocks.

Dataset	Quantity	Cartesian TARFlow	Latent TARFlow	Four-mask ConformFlow	ConformFlow
CATH1	End-to-end	0.271 (0.016)	0.159 (0.003)	0.071 (0.004)	**0.043 (0.001)**
	R_g	0.250 (0.018)	0.110 (0.006)	0.102 (0.008)	**0.085 (0.004)**
	RMSD	0.310 (0.023)	0.124 (0.007)	0.077 (0.007)	**0.058 (0.003)**
	2D TICA	0.419 (0.012)	0.190 (0.009)	0.196 (0.008)	**0.132 (0.003)**
Octapeptide	End-to-end	0.410 (0.010)	0.256 (0.007)	0.189 (0.005)	**0.035 (0.000)**
	R_g	0.381 (0.013)	0.232 (0.009)	0.154 (0.008)	**0.032 (0.000)**
	RMSD	0.480 (0.010)	0.340 (0.005)	0.284 (0.002)	**0.033 (0.000)**
	2D TICA	0.595 (0.011)	0.498 (0.009)	0.491 (0.009)	**0.329 (0.004)**
ATLAS	End-to-end	0.583 (0.023)	0.445 (0.016)	0.433 (0.014)	**0.255 (0.006)**
	R_g	0.603 (0.027)	0.436 (0.021)	0.405 (0.026)	**0.305 (0.010)**
	RMSD	0.627 (0.050)	0.532 (0.043)	0.523 (0.044)	**0.330 (0.023)**
	2D TICA	0.680 (0.013)	0.583 (0.010)	0.569 (0.009)	**0.403 (0.000)**

the best distributional agreement overall. These results show that both the learned geometry-aware latent representation and the residue-level coupling architecture are important for transferable ensemble generation.

Sampling efficiency. ConformFlow generates samples by one inverse step through the normalizing flow, leading to substantially lower sampling cost. As shown in Table 1, ConformFlow substantially reduces wall-clock sampling time relative to BioEmu while maintaining competitive or improved ensemble fidelity. All timing experiments generate 2,000 i.i.d. samples on a single NVIDIA L40S GPU. ConformFlow uses a batch size of 64, while BioEmu uses its default protein-length-dependent batch size. We further demonstrate the sampling efficiency gain over autoregressive flows in Appendix Table 6. ConformFlow achieves substantially faster sampling, with the speedup increasing for longer protein sequences due to its parallel RealNVP-style coupling architecture.

5 Discussion

Limitations. ConformFlow generates ensembles through a learned continuous latent space rather than directly in Cartesian coordinates. This improves scalability and sampling efficiency, but the final ensemble quality depends on the fidelity and coverage of the encoder–decoder. Rare conformations or proteins far outside the training distribution may therefore be limited by reconstruction accuracy. Moreover, the flow provides exact likelihoods in latent space, not exact all-atom coordinate-space likelihoods.

ConformFlow is also limited by the coverage of available equilibrium MD data. Although trained on large-scale single-structure and MD datasets, the training corpus still spans only a subset of protein sequence and conformational space. Performance may degrade for very large proteins, multi-domain systems, intrinsically disordered proteins, ligand-bound states, or rare transitions that are underrepresented in current datasets. Finally, the current model generates backbone N, Cα, and C atoms only; extending it to side-chain or all-atom ensembles is an important future direction.

Opportunities. A key advantage of ConformFlow is the combination of fast single-pass sampling and exact latent likelihood evaluation. This allows generated conformations to be scored, filtered, or combined with physical and experimental constraints. Although this work demonstrates guidance using geometric observables, the same framework could incorporate constraints from real experiments such as SAXS, FRET, NMR, or cryo-EM. More broadly, ConformFlow provides a likelihood-based prior over protein conformations that can be combined with downstream inference objectives without retraining. Future work could also couple this equilibrium ensemble model with transition models, reweighting methods, or Markov state models to recover kinetic information.

6 AI USE STATEMENT

We used generative AI tools to assist with code implementation and with downloading and organizing research data. We did not use generative AI tools for other applicable tasks requiring disclosure. We reviewed all AI-assisted work, including the code and data preparation procedures, to assess correctness and consistency with the methodology described in this paper. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REFERENCES

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zugner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

BO Brandsdal and AO Smalås. Evaluation of protein–protein association energies by free energy perturbation calculations. *Protein Engineering*, 13(4):239–245, 2000.

Nicholas E Charron, Klara Bonneau, Aldo S Pasos-Trejo, Andrea Guljas, Yaoyi Chen, Félix Musil, Jacopo Venturin, Daria Gusew, Iryna Zaporozhets, Andreas Krämer, et al. Navigating protein landscapes with a machine-learned transferable coarse-grained model. *Nature chemistry*, pp. 1–9, 2025.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

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

Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noe, Carla P Gomes, Alan Aspuru-Guzik, and Kirill Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. *Advances in Neural Information Processing Systems*, 37:65791–65822, 2024.

Xiang Fu, Tian Xie, Nathan J Rebello, Bradley D Olsen, and Tommi Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *arXiv preprint arXiv:2204.10348*, 2022.

Jingjing Guo and Huan-Xiang Zhou. Protein allostery and conformational dynamics. *Chemical reviews*, 116(11):6503–6515, 2016.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. *Advances in Neural Information Processing Systems*, 37:40534–40564, 2024b.

Leon Klein and Frank Noé. Transferable boltzmann generators. *Advances in Neural Information Processing Systems*, 37:45281–45314, 2024.

Jonas Kohler, Yaoyi Chen, Andreas Kramer, Cecilia Clementi, and Frank Noé. Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 19(3):942–952, 2023.

Tony Lelièvre, Geneviève Robin, Innas Sekkat, Gabriel Stoltz, and Gabriel Victorino Cardoso. Generative methods for sampling transition paths in molecular dynamics. *ESAIM: Proceedings and Surveys*, 73:238–256, 2023.

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

Yikai Liu, Zongxin Yu, Richard J Lindsay, Guang Lin, Ming Chen, Abhilash Sahoo, and Sonya M Hanson. Exendiff: an experiment-guided diffusion model for protein conformational ensemble generation. *Prx Life*, 3(2):023013, 2025.

Jiarui Lu, Bozitao Zhong, Zuobai Zhang, and Jian Tang. Str2str: A score-based framework for zero-shot protein conformation sampling. *arXiv preprint arXiv:2306.03117*, 2023.

Antonio Mirarchi, Toni Giorgino, and Gianni De Fabritiis. mdcath: A large-scale md dataset for data-driven computational biophysics. *Scientific Data*, 11(1):1299, 2024.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Sanjeev Raja, Martin Šípka, Michael Psenka, Tobias Kreiman, Michal Pavelka, and Aditi S Krishnapriyan. Action-minimization meets generative modeling: Efficient transition path sampling with the onsager-machlup functional. *arXiv preprint arXiv:2504.18506*, 2025.

Paul Robustelli, Stefano Piana, and David E Shaw. Developing a molecular dynamics force field for both folded and disordered protein states. *Proceedings of the National Academy of Sciences*, 115 (21):E4758–E4766, 2018.

Steffen Schultze and Helmut Grubmuuller. Time-lagged independent component analysis of random walks and protein dynamics. *Journal of Chemical Theory and Computation*, 17(9):5766–5776, 2021.

David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

Charlie Tan, Majdi Hassan, Leon Klein, Saifuddin Syed, Dominique Beaini, Michael Bronstein, Alexander Tong, and Kirill Neklyudov. Amortized sampling with transferable normalizing flows. *Advances in Neural Information Processing Systems*, 38:94290–94325, 2026.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.

Yann Vander Meersche, Gabriel Cretin, Aria Gheeraert, Jean-Christophe Gelly, and Tatiana Galochkina. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic acids research*, 52(D1):D384–D392, 2024.

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

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Josh Susskind, and Miguel Angel Bautista. Simplefold: Folding proteins is simpler than you think. *arXiv preprint arXiv:2509.18480*, 2025.

Hannah K Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, 2024.

James C Whisstock and Arthur M Lesk. Prediction of protein function from protein sequence and structure. *Quarterly reviews of biophysics*, 36(3):307–340, 2003.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024.

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

A Experiment details

A.1 Distributional similarity evaluation

End-to-end distance. The end-to-end distance (EED) measures the global extension of a protein conformation. It is defined as the Euclidean distance between the terminal backbone atoms:

$$\mathrm{EED} = \|\mathbf{r}_L - \mathbf{r}_1\|_2 , \tag{30}$$

where $\mathbf{r}_1$ and $\mathbf{r}_L$ denote the coordinates of the first and last residues, respectively. In this work, EED is computed using C_α atoms.

Radius of gyration. The radius of gyration (R_g) measures the overall compactness of a protein conformation. It is defined as the root-mean-square distance of atoms from their geometric center:

$$R_g = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{r}_i - \mathbf{r}_{\mathrm{cm}}\|_2^2}, \tag{31}$$

where N is the number of atoms, $\mathbf{r}_i$ is the coordinate of atom i, and $\mathbf{r}_{\mathrm{cm}}$ is the centroid of the selected atoms. In this work, R_g is computed using backbone atoms.

Root mean square deviation. Root mean square deviation (RMSD) measures structural deviation from a reference conformation after optimal rigid-body alignment:

$$\mathrm{RMSD} = \min_{T_g \in \mathrm{SE}(3)} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\|T_g(\mathbf{r}_i) - \mathbf{r}_i^{\mathrm{ref}}\right\|_2^2}, \tag{32}$$

where $\mathbf{r}_i$ and $\mathbf{r}_i^{\mathrm{ref}}$ are corresponding atom coordinates in the generated and reference structures, and T_g is the optimal rigid-body transformation. In this work, RMSD is computed against the AlphaFold 3 predicted native structure.

Time-lagged independent component analysis. Time-lagged independent component analysis (TICA) extracts slow collective coordinates from MD trajectories by finding linear projections with maximal autocorrelation at lag time τ. It solves the generalized eigenvalue problem

$$C_\tau \mathbf{r}_i = C_0 \lambda_i \mathbf{r}_i, \tag{33}$$

where C_0 is the instantaneous covariance matrix, C_τ is the time-lagged covariance matrix, and λ_i is the autocorrelation of the i-th component. We fit TICA on reference MD trajectories using protein backbone torsion features and evaluate generated ensembles after projection onto the top two TICA components.

B Additional experiment details

B.1 Architecture details

ConformFlow models residue-wise continuous latents $\mathbf{y} \in \mathbb{R}^{L\times M}$, with $M = 4$ in all main experiments. The normalizing flow is a RealNVP-style architecture composed of 12 affine coupling blocks. Each block contains two residue-masked coupling layers: one updates odd-indexed residues while keeping even-indexed residues fixed, and the other updates even-indexed residues while keeping odd-indexed residues fixed. This odd–even masking ensures that every residue is updated once per block while preserving a block-triangular Jacobian.

The scale and translation fields in each affine coupling layer are predicted by a bidirectional masked Transformer conditioner. The conditioner receives trainable amino-acid sequence embeddings together with the frozen subset of latent variables, while the active subset is masked out to preserve exact invertibility. We use rotary positional embeddings (RoPE) over both sequence and structure-token positions to encode residue order. Each coupling network uses hidden dimension $d = 256$, 8 attention heads, and 4 Transformer layers. The Transformer blocks use pre-normalization, residual connections, multi-head self-attention, and feed-forward networks with gated nonlinear activations. The sequence embeddings, structure embeddings, Transformer conditioners, and affine output heads

are trained end-to-end. The output head predicts the affine scale and shift jointly and is initialized near zero so that each coupling layer starts close to the identity map.

The full flow contains 202M trainable parameters. The model is trained by maximum likelihood on precomputed continuous latents with the structure autoencoder frozen. We optimize with AdamW using learning rate 10^{-4}, weight decay 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.95$. The learning rate is linearly warmed up for 500 steps. We use gradient clipping with maximum norm 1.0 and train with bfloat16 mixed precision.

B.2 Plug-and-play guidance details

For each protein, we draw 25 RMSD targets from the empirical MD RMSD distribution. For each target, we run 200 independent ULA chains, producing 5000 guided samples per protein. All chains are initialized from unconditional ConformFlow samples and use sampling temperature $T = 1$. We use a Gaussian constraint width of $\sigma = 0.2$ Å. For $\mathbf{y}$-space guidance, we run 2000 ULA steps with step size 10^{-3}. For $\mathbf{z}$-space guidance, we run 2000 ULA steps with step size 5×10^{-4}. All guidance runs use 500 warm-up steps.

B.3 Architecture ablation details

We perform an ablation study to evaluate the effect of the latent representation and coupling-mask design. Table 4 compares four variants: TARFlow trained directly on Cartesian backbone coordinates, TARFlow trained on the learned continuous latent space, a ConformFlow variant using a four-mask coupling schedule, and the final ConformFlow model with stacked odd–even affine coupling blocks.

Both TARFlow baselines use 12 affine transformation layers, matching the final ConformFlow model in depth. For Cartesian TARFlow, we remove center-of-mass translation and use random rotation augmentation during training. Our TARFlow implementation follows PROSE as closely as possible. For Cartesian coordinates, we use per-atom positional embeddings. For both TARFlow baselines, residue-level permutations are applied before each affine transformation: the first permutation in the data-to-Gaussian direction is the identity permutation, while subsequent permutations use a head-to-tail residue flip. This differs from PROSE, which uses chemistry-aware sequence permutations.

The four-mask ConformFlow variant uses six coupling blocks, with each block containing four masked coupling layers over odd, even, first-half, and second-half residue partitions. This gives the same total number of coupling layers as the final model, but changes the masking pattern. The final ConformFlow model instead uses 12 stacked odd–even affine coupling blocks, where each block alternates between odd-residue and even-residue updates.

Cartesian TARFlow performs poorly across all datasets, indicating that direct likelihood modeling in backbone Cartesian space remains difficult even after removing global translation and augmenting rotations. Moving TARFlow to the latent space substantially improves performance on CATH1, but still generalizes poorly to held-out octapeptides and ATLAS. The four-mask ConformFlow variant further improves over latent TARFlow, especially on the held-out benchmarks, but remains weaker than the final odd–even coupling design.

These results do not imply that Cartesian-space flows are inherently unsuitable for protein ensembles. Rather, they suggest that simple coordinate-wise affine transformations lack sufficient geometric inductive bias for backbone conformations. More expressive Cartesian-space flows could incorporate dense pairwise interactions or equivariant message passing, but such architectures would likely increase computational cost. Similarly, generic head-to-tail permutations and first-half/second-half masks provide limited protein-specific inductive bias, especially across variable sequence lengths. This supports our use of a learned geometry-aware latent representation together with odd–even residue coupling, which provides a simple, length-agnostic masking scheme while preserving efficient likelihood evaluation and sampling.

Table 5: Autoencoder reconstruction quality across datasets. 50 test proteins per dataset. RMSD and lDDT measure global and local C_α reconstruction accuracy; the bond column reports the mean CA–CA distance (ideal 3.80 Å) as a sanity check that the decoded backbone is physically valid.

Dataset	RMSD (Å) ↓	lDDT-CA ↑	CA–CA bond (Å)
PDB	0.507 (0.0155)	0.951 (0.0040)	3.768 (0.0080)
CATH1	0.380 (0.0071)	0.983 (0.0001)	3.822 (0.0002)
CATH2	0.450 (0.0118)	0.975 (0.0002)	3.815 (0.0002)
Megasim	0.642 (0.0513)	0.980 (0.0003)	3.812 (0.0002)
mdcath	0.508 (0.0398)	0.974 (0.0003)	3.797 (0.0003)
ATLAS	0.763 (0.0250)	0.960 (0.0023)	3.797 (0.0011)

C Additional experiment results

C.1 Structure Autoencoder Evaluation.

Because all downstream sampling operates in the continuous latent space $\mathbf{y}$, the autoencoder should preserve protein backbone geometry with high fidelity. We evaluate reconstruction quality on 50 proteins from each of PDB, CATH1, CATH2, Megasim, mdcath, and ATLAS, using up to 1000 random frames per protein, and report three CA-based metrics in Table 5: RMSD, lDDT-CA (with neighbor cutoff 15 Å), and CA–CA bond length (ideal 3.80 Å). Across all datasets, the autoencoder achieves sub-Å mean RMSD, lDDT-CA above 0.95, and CA–CA bond lengths within 0.03 Åof the ideal value, indicating that both global topology and local backbone geometry are recovered with high fidelity. We therefore conclude that the learned latent representation $\mathbf{y}$ retains the backbone information needed for downstream distributional modeling.

C.2 FES visualization

We present additional free-energy surface (FES) visualizations on representative proteins from the CATH1 dataset. Figure 2 shows two-dimensional free-energy surfaces along the top two TICA components, while Figure 3 shows one-dimensional free-energy profiles along end-to-end distance (EED), radius of gyration (R_g), and RMSD to the native structure.

C.3 Parallel sampling efficiency.

We further compare the scaling behavior of RealNVP-style parallel sampling against an autoregressive flow baseline. Timing is measured on a single NVIDIA L40S GPU using 32 samples per batch. As shown in Table 6, RealNVP sampling remains highly efficient as sequence length increases, while autoregressive sampling becomes progressively more expensive due to its sequential generation procedure. The speedup grows from $1.9\times$ at length 8 to $38.3\times$ at length 500, demonstrating the advantage of parallel invertible coupling layers for long protein sequences.

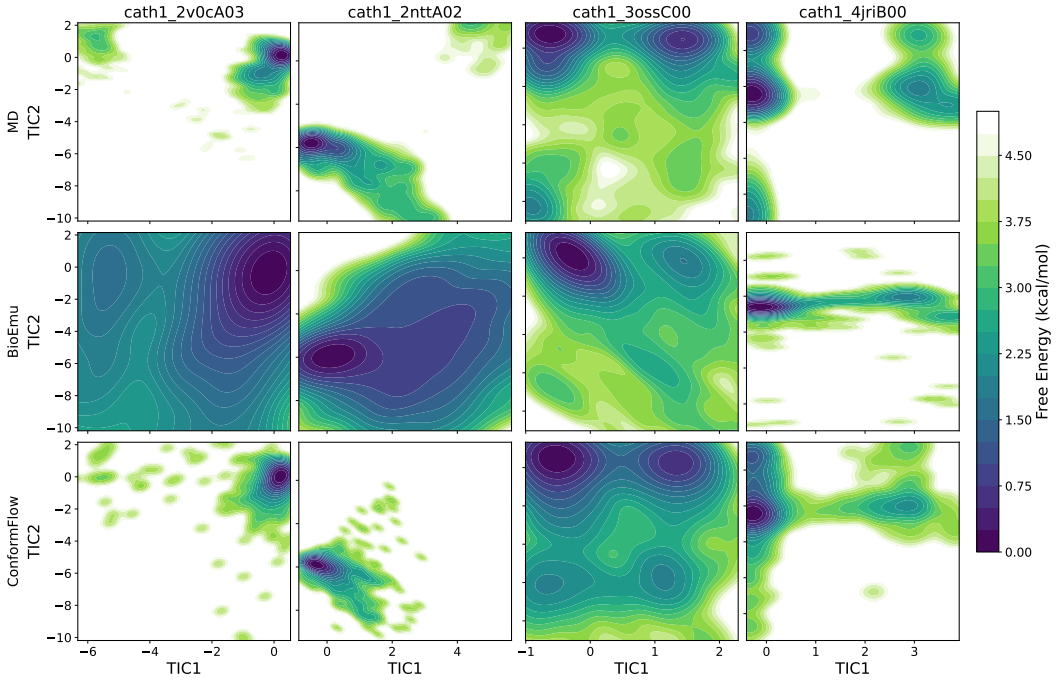

Figure 2: Free energy surfaces on representative CATH1 proteins along the top two TICA components. The TICA model is parameterized using C_α pseudo-torsion angles from the reference MD simulations, and the resulting projection is applied to conformational ensembles generated by ConformFlow and the BioEmu baseline.

Table 6: Sampling time comparison between RealNVP-style parallel sampling and an autoregressive flow baseline. Timing is measured on a single NVIDIA L40S GPU with 32 samples per batch. Lower time is better.

Sequence length	RealNVP (s)	AR (s)	Speedup
8	0.131	0.245	1.9×
16	0.142	0.481	3.4×
32	0.162	0.954	5.9×
48	0.193	1.454	7.5×
64	0.219	1.930	8.8×
96	0.257	2.850	11.1×
128	0.344	3.947	11.5×
192	0.524	8.815	16.8×
256	0.732	15.918	21.7×
384	1.199	35.348	29.5×
500	1.640	62.767	38.3×

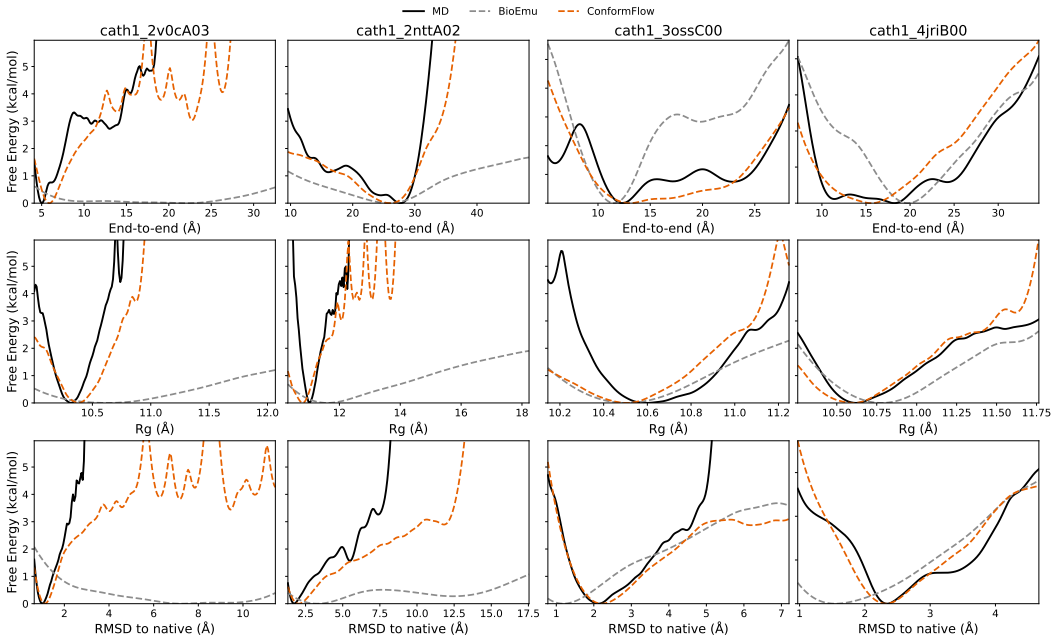

Figure 3: One-dimensional free energy profiles on representative CATH1 proteins along end-to-end distance, radius of gyration (R_g), and RMSD with respect to the native structure, constructed from reference MD simulations, ConformFlow samples, and BioEmu samples.