Accelerating Protein Molecular Dynamics Simulation with DeepJump

Anonymous authors
Paper under double-blind review

Abstract

Unraveling the dynamical motions of biomolecules is essential for bridging their structure and function, yet it remains a major computational challenge. Molecular dynamics (MD) simulation provides a detailed depiction of biomolecular motion, but its high-resolution temporal evolution comes at significant computational cost, limiting its applicability to timescales of biological relevance. Deep learning approaches have emerged as promising solutions to overcome these computational limitations by learning to predict long-timescale dynamics. However, generalizable kinetics models for proteins remain largely unexplored, and the fundamental limits of achievable acceleration while preserving dynamical accuracy are poorly understood. In this work, we fill this gap with DeepJump, an Euclidean-Equivariant Flow Matching-based model for predicting protein conformational dynamics across multiple temporal scales. We train DeepJump on trajectories of the diverse proteins of mdCATH, systematically studying our model's performance in generalizing to long-term dynamics of fast-folding proteins and characterizing the trade-off between computational acceleration and prediction accuracy. We demonstrate the application of DeepJump to ab initio folding, showcasing prediction of folding pathways and native states. Our results demonstrate that DeepJump achieves significant $\approx$1000$\times$ computational acceleration while effectively recovering long-timescale dynamics, providing a stepping stone for enabling routine simulation of proteins.

1 Introduction

Proteins form the functional infrastructure of biological systems, performing complex actions through intricate dynamic reconfiguration (Nelson et al., 2008). Uncovering protein motion is hence essential to elucidating the mechanisms of biological processes, and a key step towards developing strategies to combat disease at the molecular level. Classical Molecular Dynamics (MD) simulation describes the kinetics of molecules by integrating the Newtonian equations of motion yielded from atomic force-fields (Schlick, 2010). However, high-frequency motion components often limit the practical timestep of first-principles simulation, making relevant biological timescales computationally prohibitive to achieve (Freddolino et al., 2010). In contrast, Deep Learning models have demonstrated remarkable success in resolving challenging prediction tasks of protein thermodynamics, such as in structure prediction (Jumper et al., 2021) and in ensemble distribution generation (Jing et al., 2024a; Lewis et al., 2025). Still, while existing large-scale models enable generalizable prediction of static basin or equilibrium states, generalizable kinetics remains a challenging frontier for modeling biological processes, as it requires training and evaluation across vast conformational phase spaces through computationally expensive simulations.

In this work, we hypothesize that training on short, diverse trajectories is sufficient to capture generalizable dynamical behavior. To test this hypothesis, we develop DeepJump, a generative model that combines flow matching with equivariant neural networks to model protein conformational transitions. We train our model on the mdCATH (Mirarchi et al., 2024) dataset, evaluating it on extended simulations of fast-folding proteins (Majewski et al., 2023). We show how the learned simulator successfully generalizes beyond its training timescales, reproducing long-term protein dynamics while achieving orders-of-magnitude

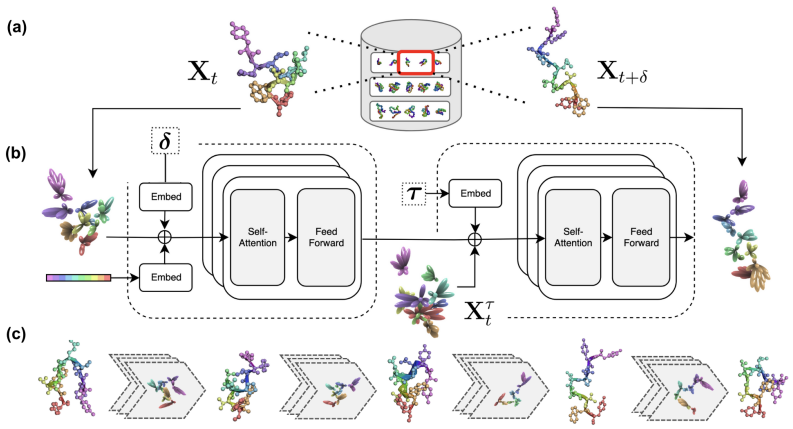


Figure 1: **DeepJump Architecture and Training**. (a) Training data consists of diverse protein trajectory snapshots from the mdCATH dataset, providing structural diversity across different protein folds. (b) Our model treats proteins as clouds of spherical harmonics, using a two-stage architecture with a current-step conditioner and a generative transport network to predict the next 3D state. (c) Our generative sampler produces long trajectories by iteratively jumping through large transitions.

computational speedup. We demonstrate the utility of our approach in performing ab initio folding simulations, reaching native states from extended conformations.

2 Related Work

Diverse deep learning efforts aim to reproduce MD and tackle its fundamental bottlenecks. Machine learning force fields (MLFFs) like ANI (Smith et al., 2017) NequIP (Batzner et al., 2022) and MACE (Batatia et al., 2022) have demonstrated remarkable accuracy in reproducing atomic potentials. However, MLFFs are constrained by timestep limitations, as they require integration of motion at high resolution. Recent breakthroughs have instead turned to generative models, with approaches like AlphaFlow (Jing et al., 2024a) and BioEmu (Lewis et al., 2025) training over trajectory data to generate Boltzmann ensembles. Still, ensemble models fail to capture the whole picture of dynamics, as trajectories are needed for causal understanding. In further development towards capturing kinetics, recent models (Schreiner et al., 2023; Li et al., 2024; Jing et al., 2024b) enable sampling of dynamics trajectories with large steps, suggesting a pathway for accelerated MD simulation where the large time that is skipped outweighs the cost of evaluating the neural network. In this work, we build upon EquiJump (Costa et al., 2024b) and JAMUN (Daigavane et al., 2025), leveraging equivariant neural networks (Geiger and Smidt, 2022) and generative transport for predicting proximal structure states.

3 Methods

3.1 Model Architecture and Training

We follow EquiJump (Costa et al., 2024b) and model a protein $(\mathbf{R}, \mathbf{X})$ as a length N sequence $\mathbf{R} \in \mathcal{R}^N$ and a 3D gas of geometric features $\mathbf{X} = (\mathbf{P}, \mathbf{V})$, where $\mathbf{P} \in \mathbb{R}^{N\times 3}$ are coordinates of $\mathbf{C}_\alpha$ atoms and $\mathbf{V} \in \mathbb{R}^{N\times 13\times 3}$ are 3D features listing for each residue the relative position of heavy atoms in relation to the $\mathbf{C}_\alpha$. We train the model using trajectory data $[\mathbf{X}_t]_{t=1}^L$. Given a starting point t and a jump time δ, we take state transition $(\mathbf{X}_t, \mathbf{X}_{t+\delta})$ as the endpoints of a Conditional Flow Matching/Stochastic Interpolant (Li et al., 2024; Lipman et al., 2022; Albergo et al., 2023) model that maps from a noised source state $\rho_0 = \rho(\mathbf{X}_t + \mathbf{Z})$, where $\mathbf{Z} \sim \mathcal{N}$, into a future time step $\rho_1 = \rho(\mathbf{X}_{t+\delta}|\mathbf{X}_t)$ (Figure 1.b). We learn a model

to predict the final state directly, $\hat{\mathbf{X}}^{\mathbf{1}}(\mathbf{X}^\tau|\tau) \approx \mathbf{X}^1$, and in sampling reparameterize via $b(\mathbf{X}^\tau|\tau) = \frac{1}{(1-\tau)}\left(\hat{\mathbf{X}}^1(\mathbf{X}^\tau|\tau) - \mathbf{X}_t\right)$ (Jing et al., 2024a).

Our architecture consists of two main stages (Figure 1.b). First, a conditioning encoder computes $\mathbf{H}_t = \mathbf{f}_{\text{cond}}(\mathbf{X}_t, \mathbf{R}, \delta)$ from the current structural state $\mathbf{X}_t$, sequence $\mathbf{R}$, and jump time δ. Second, a transport network $\mathbf{f}_{\text{transp}}(\mathbf{X}^\tau|\tau, \mathbf{H}_t)$ iteratively updates the latent state $\mathbf{X}^\tau$ to generate a new configuration. Both networks use Euclidean-equivariant architectures (Geiger and Smidt, 2022) inspired by Transformer mechanisms (Vaswani et al., 2017) adapted to equivariant space (see Appendix A for details). During training, we optimize pairwise 3D vectors between all atoms within $d = 25$Å using the Huber Loss (Costa et al., 2024a; Huber, 1992).

We train our model with the diverse structures of the mdCATH dataset (Mirarchi et al., 2024) at 320 K, consisting of 5 replicas of 500 ns trajectories for 5,398 domains. While this dataset encompasses a broad range of different proteins, it is not sufficient for capturing long-timescale dynamical behavior and equilibrium properties due to its limited simulation time. Instead, for evaluating our dynamics we test our model on the dataset of 12 fast-folder proteins of (Majewski et al., 2023) based on (Lindorff-Larsen et al., 2011). This set provides significant accumulated simulation time with thousands of trajectory snippets across the phase space, enabling precise estimation of dynamical variables.

4 Results

4.1 Mapping the Frontiers of MD Acceleration

To evaluate our approach in capturing long-term dynamics beyond its short training trajectories, we sample our model starting across the conformational phase space of the fast-folding proteins. For that, we employ Time-lagged Independent Component Analysis (TICA) (Molgedey and Schuster, 1994; Pérez-Hernández et al., 2013) and find clusters that represent macrostates in the reduced dimensional space. To evaluate a simulation, we fit a Markov State Model (MSM) to transition counts between clusters, comparing its properties to those

Figure 2: **Acceleration Fronts**. We investigate the tradeoff between simulation fidelity and computational speedup by varying model scale and conditioned jump size δ.

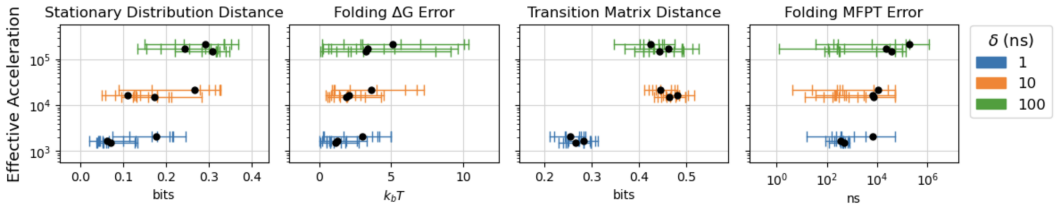


Table 1: **Model Performance across Jump Sizes**. We fit Markov State Models (MSM) to the transitions between TIC-based clusters, and compare obtained MSMs from reference and from learned models. Results are averaged over the fast-folding proteins. We use Jensen Shannon Divergence to measure distribution distance for stationary distributions and transition matrix (averaging over rows), and absolute differences otherwise. To estimate folding metrics, we compare energetics and timescales between clusters corresponding to the α-helix state and the crystal state.

δ (ns)	1			10			100		
Model Dimensionality	**32**	**64**	**128**	**32**	**64**	**128**	**32**	**64**	**128**
Stationary Distribution Distance (bits)	0.18	0.06	0.07	0.27	0.11	0.17	0.29	0.24	0.31
Folding ΔG Error (k_bT)	3.02	1.24	1.14	3.64	2.05	1.88	5.11	3.37	3.24
Transition Matrix Distance (bits)	0.25	0.28	0.27	0.45	0.48	0.46	0.42	0.46	0.44
Folding MFPT Error (us)	69.2	0.3	0.4	10.5	6.7	7.0	190.5	225.0	37.4

of reference. To better understand the trade-offs between simulation accuracy and computational speedup, we compare the quality of MSMs from simulations of models with different capacities and conditioned at differing jump step sizes. For each fast-folder, we start 1200 replicas uniformily across the clusters, and perform 1000 simulation steps. Table 1 compares MSM metrics across these model configurations. We present these results in condensed form in Figure 2, where we estimate effective acceleration relative to Amber force-field (Wang et al., 2004) simulations (32 real s / simulation ns on A6000 (Exxact Corp.)) for the Lambda protein. Our plots show that while jump size impacts simulation quality, model scaling can modestly compensate for this degradation, with substantial acceleration achievable within acceptable quality bounds.

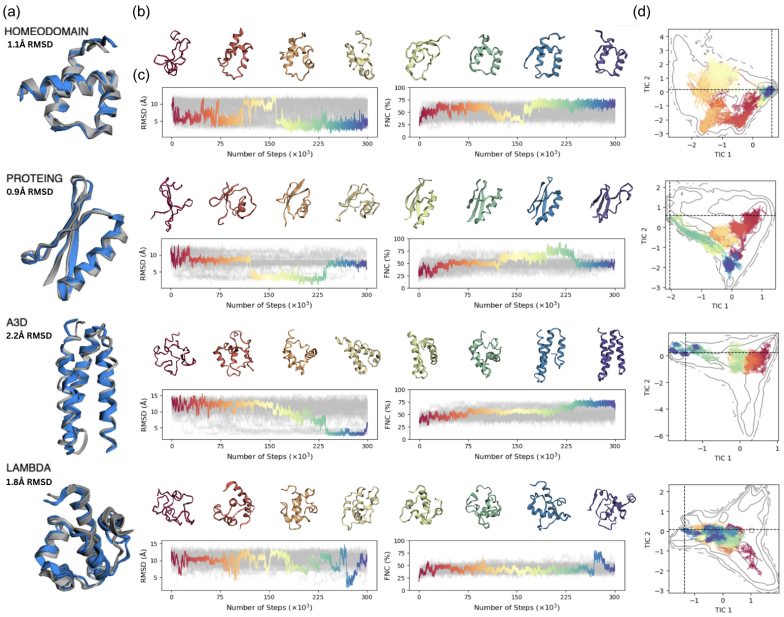
4.2 Accelerating ab initio Folding

To evaluate DeepJump in a practical setting, we investigate its performance on the challenging task of ab initio protein folding. For each fast-folder, we start 64 replicas from an extended β-sheet state and perform 300k simulation steps. In Figure 3, we show folding trajectories and sampled structures with the closest match to the native state. Investigation of the curves reveals that our simulation successfully generates smooth folding pathways with physically realistic conformational transitions. Table 2 compares the performance of models using different jump sizes δ. We find that folding success varies with step size: models with 1ns and 10ns steps achieving the highest quality results, with 100ns steps failing to fold some proteins entirely. Refer to Appendix D for further discussion.

4.3 Experimental Correlation

We evaluate the kinetic quality of DeepJump folding simulations by comparing predicted folding times to experimental measurements. For that, we use 17 proteins of the dataset curated by [**?**], which provides folding kinetic rates for PDB crystal structures. We start 32 replicas from the β-sheet state and simulate 1 μs with learned 1 ns-step accelerated

Figure 3: **Generative Simulation of Folding Paths**. We show 300k 1ns simulation steps for 64 replicas of fast-folding proteins. (a) We show best aligning sample (blue) to crystal structure (gray). (b) We highlight a trajectory achieving highest fraction of native contacts (FNC) and plot its steps aligned to crystal. (c) We show the time evolution of RMSD and FNC for the replicas (gray) and highlighted trajectory (color). (d) We plot TIC coordinates against the reference phase space (gray).



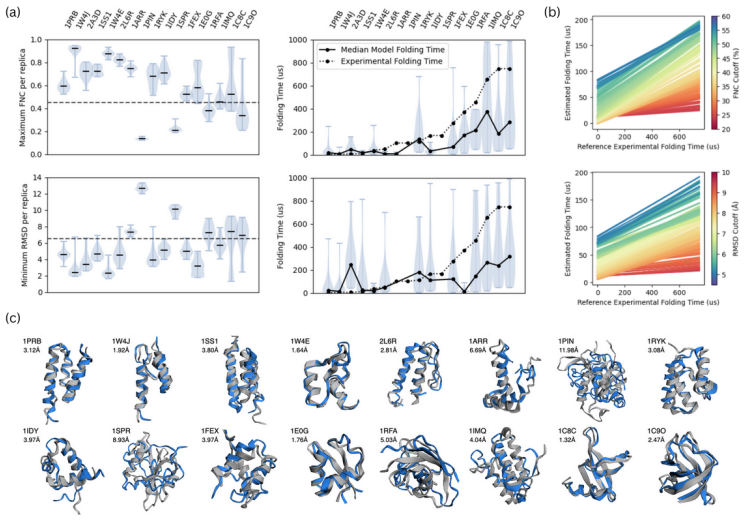


Figure 4: **Correspondence between Model and Experimental Observables**. **(a)** We plot the distribution of best performing score per trajectory replica and cutoff (dashed) for folding arrival time (Left). We show the distribution of folding times at the specified cutoff (Right). **(b)** We fit linear regressions at different folding cutoffs of FNC and RMSD, finding only positive correlations between model estimation and experimental measures. **(c)** We show samples with lowest RMSD to crystal.

dynamics (1M model steps). We find that DeepJump can fold 15 out of the 17 tested proteins. Our results are summarized in Figure 4, where we find positive correlation between experimental folding times and explicitly calculated folding mean first passage times (MFPT) from DeepJump simulations. While the correlation strength is moderate, the positive trend persists across different folding criteria (FNC and RMSD cutoffs). These results demonstrate that DeepJump can successfully reproduce folding kinetics across diverse protein systems, providing a stepping stone for learning accelerated simulators.

Conclusion

In this work we introduced DeepJump, a generative model that leverages flow matching and equivariant networks to accelerate protein dynamics simulations by learning large conformational transitions. Our approach successfully generalizes to reproduce kinetic properties of fast-folding proteins, including folding pathways and long-term distributions, while achieving orders-of-magnitude acceleration compared to traditional methods. Our model represents a promising step toward practical machine learning-accelerated simulations, suggesting a path to previously inaccessible long timescales.

Table 2: **Folding from Scratch**. We quantify folding success by counting replicas that reach the TICA-based cluster of the native crystal. For the proteins that fold, we count success per replica and Mean First Passage Time from beta sheet to folded state. Results are averaged over the fast-folders.

δ **(ns)**	**1**	**10**	**100**
Proteins Folded (%) ↑	**100.00**	**100.00**	62.50
Replicas Folded (%) ↑	50.59	**61.13**	57.23
Mininum Crystal RMSD (Å) ↓	**1.54**	1.64	2.35
Maximum FNC (%) ↑	86.40	**87.10**	77.96
MFPT ($\times 10^3$ Model Steps) ↓	97.3	81.8	12.7

Acknowledgments

We thank Ilan Mitnikov, Emine Kucukbenli, Franco Pellegrini, Mario Geiger and Mit Kotak for insightful discussions. This research was made possible through the support of DARPA (AI-BTO SCA-24-01), the Eleven Eleven Foundation, the Center for Bits and Atoms, and the MIT Media Lab Consortium.

References

M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.

I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.

S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

A. D. S. Costa, I. Mitnikov, M. Geiger, M. Ponnapati, T. Smidt, and J. Jacobson. Ophiuchus: Scalable modeling of protein structures through hierarchical coarse-graining SO(3)-equivariant autoencoders. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024a. URL https://openreview.net/forum?id=hnhhCfYeWU.

A. D. S. Costa, I. Mitnikov, F. Pellegrini, A. Daigavane, M. Geiger, Z. Cao, K. Kreis, T. Smidt, E. Kucukbenli, and J. Jacobson. Equijump: Protein dynamics simulation via so (3)-equivariant stochastic interpolants. *arXiv preprint arXiv:2410.09667*, 2024b.

A. Daigavane, B. P. Vani, D. Davidson, S. Saremi, J. Rackers, and J. Kleinhenz. Jamun: Bridging smoothed molecular dynamics and score-based learning for conformational ensembles, 2025. URL https://arxiv.org/abs/2410.14621.

Exxact Corp. Amber 24 nvidia gpu benchmarks. https://www.exxactcorp.com/blog/molecular-dynamics/amber-molecular-dynamics-nvidia-gpu-benchmarks, 2024.

P. Freddolino, C. Harrison, Y. Liu, and K. Schulten. Challenges in protein 524 folding simulations: timescale, representation, and analysis. *Nat. Phys*, 6(751-758):525, 2010.

M. Geiger and T. Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

B. Jing, S. Eismann, P. Suriana, R. J. Townshend, and R. Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

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

B. Jing, H. Stark, T. Jaakkola, and B. Berger. Generative modeling of molecular dynamics trajectories. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*, 2024b.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

S. Lewis, T. Hempel, J. Jiménez-Luna, M. Gastegger, Y. Xie, A. Y. Foong, V. G. Satorras, O. Abdin, B. S. Veeling, I. Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, page eadv9817, 2025.

S. Li, Y. Wang, M. Li, J. Zhang, B. Shao, N. Zheng, and J. Tang. F^3low: Frame-to-frame coarse-grained molecular dynamics with se(3) guided flow matching, 2024.

Y.-L. Liao and T. Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28 (2):129–137, 1982.

M. Majewski, A. Pérez, P. Thölke, S. Doerr, N. E. Charron, T. Giorgino, B. E. Husic, C. Clementi, F. Noé, and G. De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature Communications*, 14(1):5739, 2023.

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

L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical review letters*, 72(23):3634, 1994.

D. L. Nelson, A. L. Lehninger, and M. M. Cox. *Lehninger principles of biochemistry*. Macmillan, 2008.

G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1), 2013.

T. Schlick. *Molecular modeling and simulation: an interdisciplinary guide*, volume 2. Springer, 2010.

M. Schreiner, O. Winther, and S. Olsson. Implicit transfer operator learning: Multiple time-resolution surrogates for molecular dynamics, 2023.

J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004.

Appendix

A Architecture and Training Details

To operate efficiently on large proteins, we adapt a form of the attention mechanism to handle equivariant vectors (Algorithm 1). Drawing from GVP (Jing et al., 2020), our feedforward layers (Algorithm 2) interact vector ($l = 1$) and scalar ($l = 0$) features by incorporating vector norms into scalar processing, and gating vectors through scalars. All network modules incorporate residual connections and equivariant LayerNorm (Liao and Smidt, 2022) for stable training. We train all models with 6 layers for the conditional network and 6 layers for the transport network. Models of dimensionality H of 32, 64 and 128 have 260k, 1M and 4M parameters respectively. We train our models on 4 A6000 machines. Models are trained for 500k steps with batch size of 128 and crop length of 256. We use the Adam optimizer with learning rate decaying linearly from 5×10^{-3} to 3×10^{-3}, and gradient norm clip of 0.1.

Algorithm 1 DeepJump Self-Attention

Require: Tensor Cloud $(\mathbf{V}, \mathbf{P})$
Require: Number of Heads N_h
Require: Spatial Distance Embedding f_d
Require: Sequence Distance Embedding f_s
$\mathbf{k}, \mathbf{q}, \mathbf{v} \leftarrow \text{Linear}^{3\times(H\times N_h)}(\mathbf{V})$
$\mathbf{v}_{ijh} \leftarrow \mathbf{v}_{jh} \oplus Y(\mathbf{P}_i - \mathbf{P}_j)$
$\mathbf{s}_{ijh} \leftarrow \mathbf{k}_{ih} \cdot \mathbf{q}_{jh} + f_s(i-j) + f_d(||\mathbf{P}_i - \mathbf{P}_j||_2)$
$\mathbf{v}_{ih} \leftarrow \sum_j^N \text{Softmax}(\mathbf{s}_{ijh}) \cdot \mathbf{v}_{ijh}$
$\mathbf{V}'_i \leftarrow \text{Linear}^H(\bigoplus_h^{N_h} \mathbf{v}_{ih})$
return $(\mathbf{V}, \mathbf{P})$

Algorithm 2 DeepJump FeedForward

Require: Tensor Cloud $(\mathbf{V}, \mathbf{P})$
Require: Scaling Factor F
Require: Activation Function σ
$\mathbf{V}^0_{\shortparallel}, \mathbf{V}^0_{\times} \leftarrow \text{Linear}^{2\times(F\times H)}(\mathbf{V}^0)$
$\mathbf{V}^1_{\shortparallel}, \mathbf{V}^1_{\times} \leftarrow \text{Linear}^{2\times(F\times H)}(\mathbf{V}^1)$
$\mathbf{V}_{\shortparallel} \leftarrow \sigma(\mathbf{V}^0_{\shortparallel}) \oplus \sigma(\mathbf{V}^1_{\shortparallel})$
$\mathbf{V}_{\times} \leftarrow \left(\sigma(\mathbf{V}^1_{\times}) \cdot \mathbf{V}^0_{\times}\right) \oplus \left(\sigma(\mathbf{V}^0_{\times}) \cdot \mathbf{V}^1_{\times}\right)$
$\mathbf{V} \leftarrow \text{Linear}^H(\mathbf{V}_{\shortparallel} \oplus \mathbf{V}_{\times})$
return $(\mathbf{V}, \mathbf{P})$

In all models, we use $N_h = 4$, $F = 2$, $\sigma = \tanh(||\cdot||_2^2)$ where the norm squared is computed multipliticy-wise. We embed sequence distance with $f_s = \text{Embed}(\min(\max(i-j+k_s, 2k_s), 0))$ with $k_s = 32$. We use a gaussian basis function for the spatial distance embed f_d.

B Markov State Model and Dynamical Equilibration

We fit 4 Time-lagged Independent Components (TICs) (Pérez-Hernández et al., 2013) to the reference data with a lag time of 10 ns. To partition the TIC space, we apply k-means clustering (Lloyd, 1982) with 32 clusters. We construct a Markov State Model from transition counts with lag time of 1ns between clusters and estimate its stationary distribution. We correct sampling densities by reweighting each cluster according to the ratio of its stationary probability to its observed frequency in our simulations. When comparing the MSM transition matrices of learned models to reference data, we compare the matrix to the δ-th power to account for the different temporal resolutions.

C Free Energy Profiles

We correct our measured densities to the MSM stationary distribution to estimate free energies (Figure 5, Appendix B). Analysis of the TIC free energy profiles shows that the learned simulator is able to generalize to unseen proteins and across the phase space. Similarly, while RMSD and FNC energy (Figure 5) analysis suggests a bias towards compact conformations, the model overall shows strong agreement with the reference data.

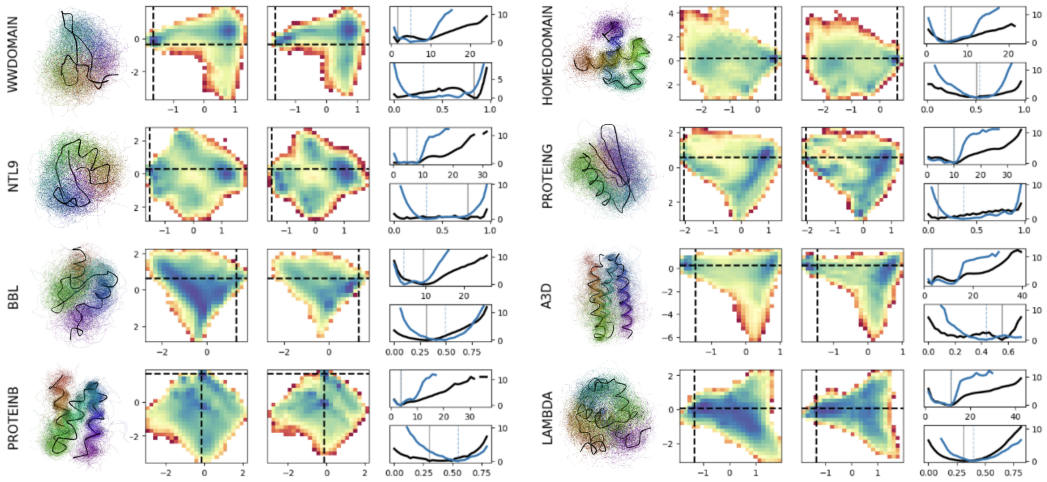


Figure 5: **Free Energy Profiles of the Fast Folder Proteins**. Ensemble visualization and free energy landscapes in TIC space (Appendix B) comparing DeepJump simulations with reference molecular dynamics. Crystal 3D structure is shown in black. Free energy profiles for RMSD (top) and especially fraction of native contacts (FNC) (bottom) demonstrate strong agreement between model (blue) and reference data (black).

D Extended ab initio Plots

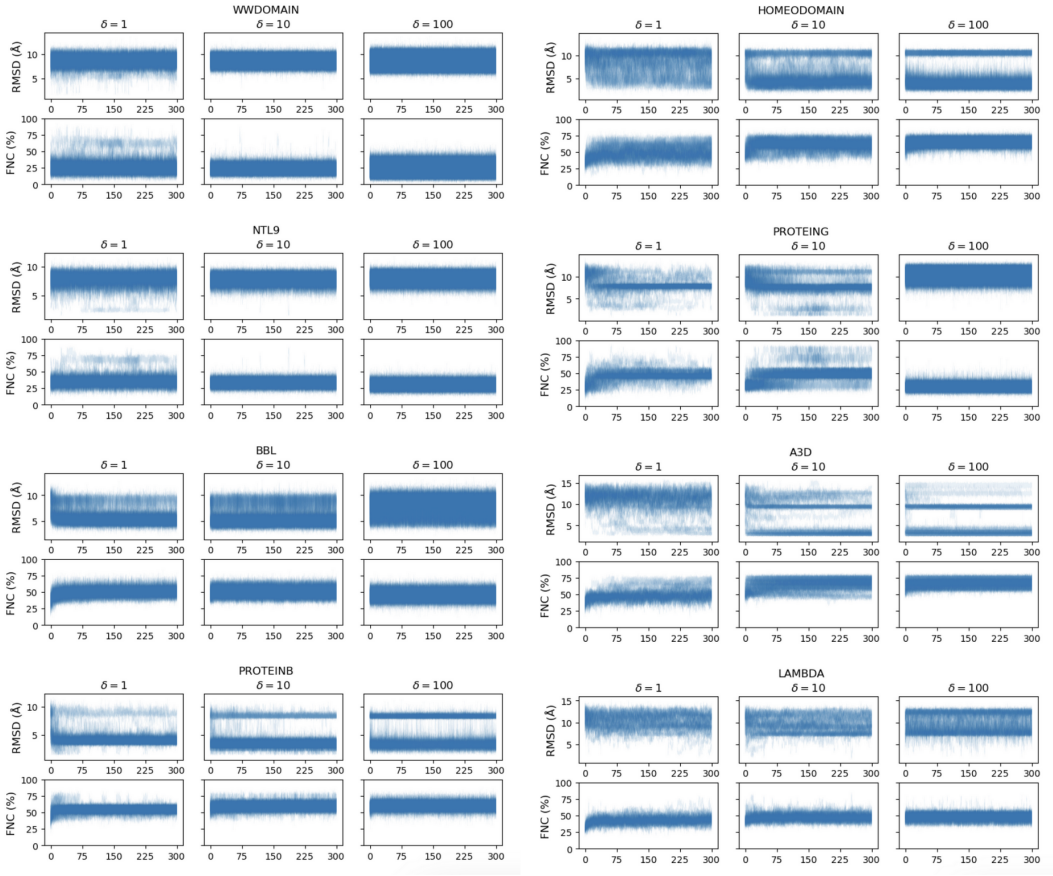


Figure 6: **Extended Plots for Folding Simulation**.

In Figure 6 we plot the evolution of RMSD and FNC over 300k model simulation steps for different learned steps sizes δ. We observe that $\delta = 1$ ns shows the most consistent folding success across proteins, frequently reaching native basins and maintaining stability whereas $\delta = 10$ ns demonstrates intermediate stability (as seen in WW domain and NTL9). While $\delta = 100$ ns manages to fold several proteins, it fails to sample high-energy transition pathways that require fine-grained conformational sampling, such as the complex folding routes observed in NTL9 and Protein G. This is due to the increasing difficulty of accurately modeling large conformational transitions over extended time intervals, where smaller steps enable the model to capture rare barrier-crossing events, while larger jumps may bypass conformational pathways or become trapped in spurious local minima.

E Model Limitations

While our model successfully generalizes to most fast-folder proteins, we found limits to its applicability across all systems. In particular, we found that it fails on proteins much smaller than those in the training data (e.g., Chignolin or Trp-Cage), generating chemically invalid states. We also highlight bias (Figure 5) toward globular conformations and basin states, as the training data predominantly consists of well-folded protein domains, limiting the model's ability to capture disordered or extended conformational states that are often crucial to pathway modeling. Finally, our modeling assumes standard residues, which prevents application to proteins with non-standard amino acids (e.g., fast-folder Villin).