Accelerating Protein Molecular Dynamics Simulation with DeepJump

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 computational acceleration while effectively recovering large-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, structurally diverse trajectories can successfully 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 structurally diverse mdCATH dataset, evaluating it on extended microsecond simulations of fast-folding proteins. We show how the learned simulator successfully

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

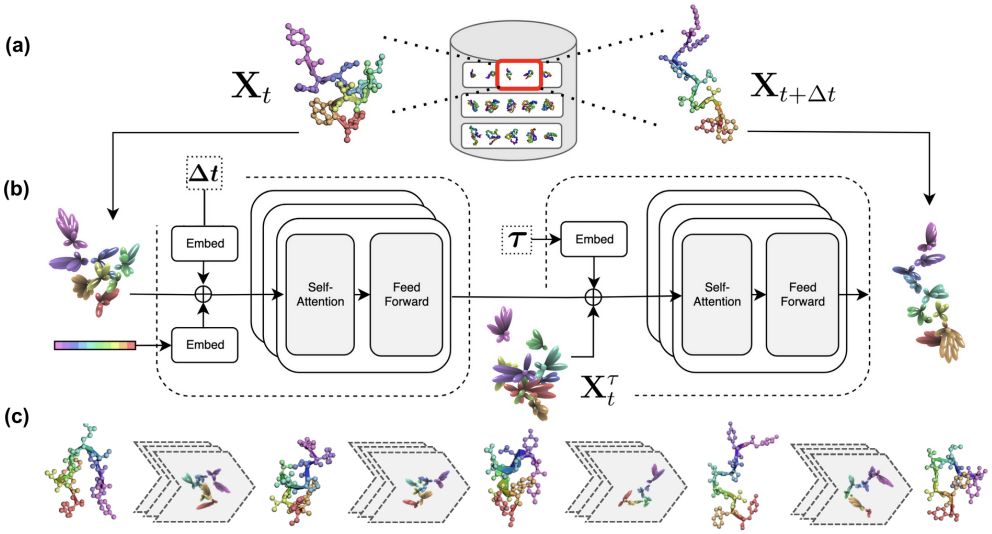

Figure 1: **DeepJump Architecture and Training**. (a) Training data consists of diverse protein trajectory snapshots from the mdCATH dataset, providing structural diversity across different protein folds. (b) Two-stage architecture with conditioning encoder and transport network using equivariant neural networks to predict conformational transitions. (c) Generative sampling produces long-timescale trajectories by iteratively applying learned jumps to explore protein conformational space.

generalizes beyond its training timescales to reproduce long-term protein dynamics while achieving orders-of-magnitude computational speedup. We analyze the trade-offs between acceleration and simulation quality across different model capacities and temporal jump sizes. Finally, we demonstrate the utility of our model in performing ab initio folding simulations from extended conformations to native states.

2 Related Work

Diverse efforts aim to reproduce and tackle the fundamental bottlenecks of running MD. 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 potentials while maintaining computational efficiency. However, MLFFs are constrained by the timestep limitations, as they still require integration of the full atomic equations of motion around femtosecond 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 mechanistic 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., 2024], incorporating multiple step sizes, as in ITO [Schreiner et al., 2023], while employing the efficient of guided Flow Matching [Lipman et al., 2022], as in F^{3}low [Li et al., 2024]. We extend these works to investigate the acceleration-accuracy trade-offs across diverse protein systems and demonstrate generalization to long-timescale dynamics from short trajectories.

3 Methods

3.1 Generative Model

We follow EquiJump [Costa et al., 2024] and model a protein $(\mathbf{R}, \mathbf{X})$ as a sequence $\mathbf{R} \in \mathcal{R}$ 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

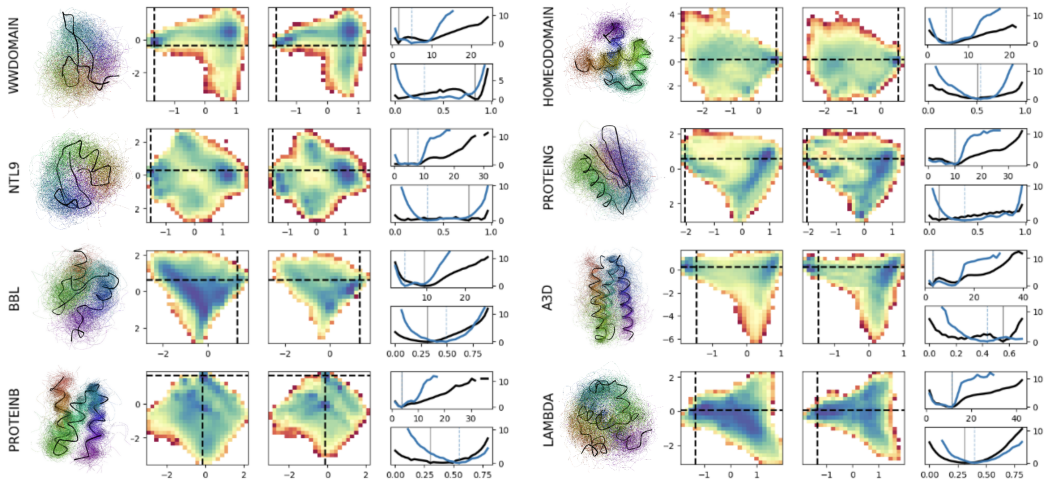

Figure 2: **Dynamics Generalization to the Fast Folder Proteins**. Ensemble visualization and free energy landscapes in TIC space comparing DeepJump simulations with reference molecular dynamics. Crystal 3D structure is shown in black. Free energy profiles for RMSD (top) and especially fraction of native contacts (FNC) (bottom) demonstrate strong agreement between model (blue) and reference data (black). The model successfully captures the main conformational basins and transition pathways of the fast-folding proteins.

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] 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 $\rho_1 = \rho(\mathbf{X}_{t+\delta}|\mathbf{X}_t)$. We learn a model to predict $\hat{\mathbf{X}}^1(\mathbf{X}^\tau|\tau) \approx \mathbf{X}^1$, and in sampling reparameterize via $b(\mathbf{X}^\tau|\tau) = \frac{1}{(1-\tau)}\big(\hat{\mathbf{X}}^1(\mathbf{X}^\tau|\tau) - \mathbf{X}_t\big)$

3.2 Architecture and Training

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 distances between all atoms within $d = 25$Å using the Huber Loss [Huber, 1992].

3.3 Datasets

To ensure the generalization power of our model, we train it using the diverse structures of the mdCATH dataset [Mirarchi et al., 2024]. This dataset consists of all-atom systems for 5,398 domains, modeled with a state-of-the-art classical force field, and simulated in five replicates of 500 ns from the crystal state, each at five temperatures from 320 K to 450 K. 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 per trajectory. 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]. In constrast to the training data, this set provides hundreds of microseconds of simulation time, enabling precise estimation of dynamical variables and asymptotic behavior.

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 (ns)	6928	346	471	10587	6796	7004	190511	22504	37687

Figure 3: **Mapping Acceleration Fronts**. We investigate the tradeoff between simulation fidelity and computational speedup by varying model scale and conditioned jump size δ. We find that larger jumps degrade simulation quality, with increased model capacity reducing error but only partially mitigating the effect.

4 Results

4.1 DeepJump Generalizes to Fast-Folder Phase Space

To assess DeepJump's ability to capture long-term dynamics beyond its short training trajectories, we extensively sample our model 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. Refer to Appendix B for further details. For each fast-folder, we start 1200 replicas uniformly across the clusters, and perform 1000 simulation steps. We fit a Markov State Model (MSM) to transition counts between clusters, and correct our measured observables to the MSM stationary distribution to estimate free energies (Figure 2). 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 2) analysis suggests a bias towards compact conformations, the model overall shows strong agreement with the reference data.

4.2 Mapping the Frontiers of MD Acceleration

To better understand the trade-offs between simulation accuracy and computational speedup, we analyze the MSMs constructed from simulations with different model capacities and jump step sizes, comparing them to MSMs built from the reference data. Table 1 shows the quantitative comparison of MSM properties across different configurations. We present these results in condensed form in Figure 3, 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 significantly impacts simulation quality, model scaling can modestly compensate for this degradation. Nevertheless, our results reveal that substantial acceleration remains achievable within acceptable quality bounds.

4.3 Accelerating ab initio Folding

To evaluate DeepJump in a practical application, 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 4, we show folding trajectories and sampled structures with the closest match to the native state. Investigation of the curves reveals that our simulation successfully captures 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, while 100ns steps fail to fold some proteins entirely. 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 and intermediate states that are crucial for successful folding, while larger jumps may bypass conformational pathways or become trapped in local minima. Refer to Appendix C for further discussion.

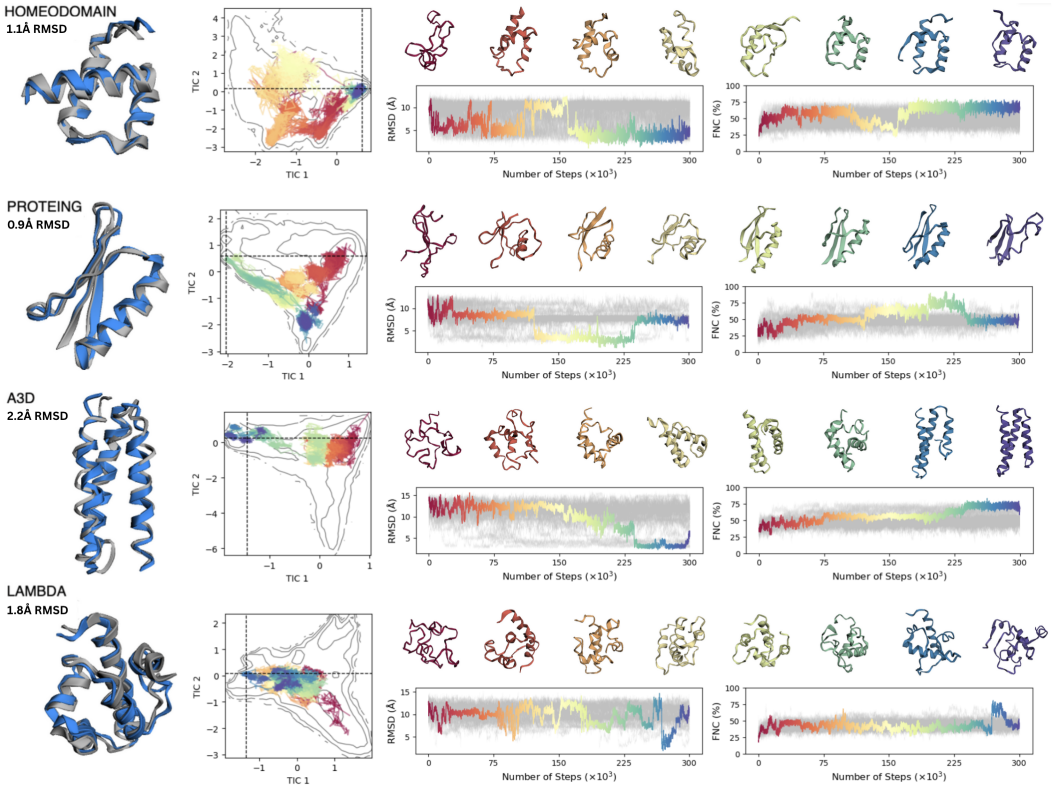

Figure 4: **Generative Simulation of Protein Folding**. We run thousands of simulation steps for the fast-folding proteins, showing the evolution of TIC coordinates, RMSD, and fraction of native contacts (FNC). We highlight a trajectory achieving highest FNC structure.

4.4 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 2) 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).

Table 2: **Folding from Scratch**. We quantify folding success by identifying trajectories that reach the TICA-based cluster corresponding to the native crystal state. For the proteins that fold, we count success per replica and Mean First Passage Time (MFPT). 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 (Model Steps)	97322.98	81795.69	**12686.58**

Conclusion

We have presented DeepJump, a generative model that leverages flow matching and equivariant neural networks to accelerate protein molecular dynamics simulations by learning conformational transitions from diverse trajectories. Our approach successfully reproduces key dynamical properties of fast-folding proteins, including realistic folding pathways and equilibrium distributions, while achieving orders-of-magnitude acceleration compared to traditional force-field simulations. Through analysis of acceleration fronts, we demonstrate important trade-offs between simulation speed and accuracy, where larger jump sizes provide greater computational speedup at the cost of simulation quality, with model scaling offering compensation. Additionally, in ab initio folding experiments, we show that the model can successfully fold proteins from extended conformations to native-like states, with folding success depending on the chosen temporal step size. In conclusion, DeepJump represents a promising step toward practical machine learning-accelerated molecular simulations, offering a path to building simulators to previously inaccessible timescales.

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, 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*, 2024.

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 and scalar 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 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 $(\mathbf{V}, \mathbf{P})$

$\mathbf{k}, \mathbf{q}, \mathbf{v} \leftarrow \text{Linear}^{3\times N_h \times H}(\mathbf{V})$

$\mathbf{v}_{ijh} \leftarrow \mathbf{v}_{jh} \oplus Y(\mathbf{P}_i - \mathbf{P}_j)$

$\mathbf{s}_{ij} \leftarrow \mathbf{k}_{ih} \cdot \mathbf{q}_{jh} + f(i-j, ||\mathbf{P}_i - \mathbf{P}_j||)_h$

$\mathbf{v}_h \leftarrow \sum_j^N \text{Softmax}(\mathbf{s}_{ijh}) \cdot \mathbf{v}_{ijh}$

$\mathbf{V}' \leftarrow \text{Linear}^H(\bigoplus_h^{N_h} \mathbf{v}_h)$

return $(\mathbf{V}, \mathbf{P})$

Algorithm 2 DeepJump FeedForward

Require: Tensor Cloud $(\mathbf{P}, \mathbf{V})$

$\mathbf{V}^0, \mathbf{V}^g \leftarrow \text{Linear}^{2\times(f\times H)}(\mathbf{V}^0)$

$\mathbf{V}^1, \mathbf{V}^n \leftarrow \text{Linear}^{2\times(f\times H)}(\mathbf{V}^1)$

$\mathbf{V} \leftarrow \sigma(\mathbf{V}^0) \oplus \sigma(\mathbf{V}^g) \cdot \mathbf{V}^1 \oplus ||\mathbf{V}^n||_2^2$

$\mathbf{V} \leftarrow \text{Linear}^H(\mathbf{V})$

return $(\mathbf{V}, \mathbf{P})$

B Markov State Model and Dynamical Equilibration

We fit 4 TIC components [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 δ-th matrix power to account for the different temporal resolutions.

C Extended ab initio Plots

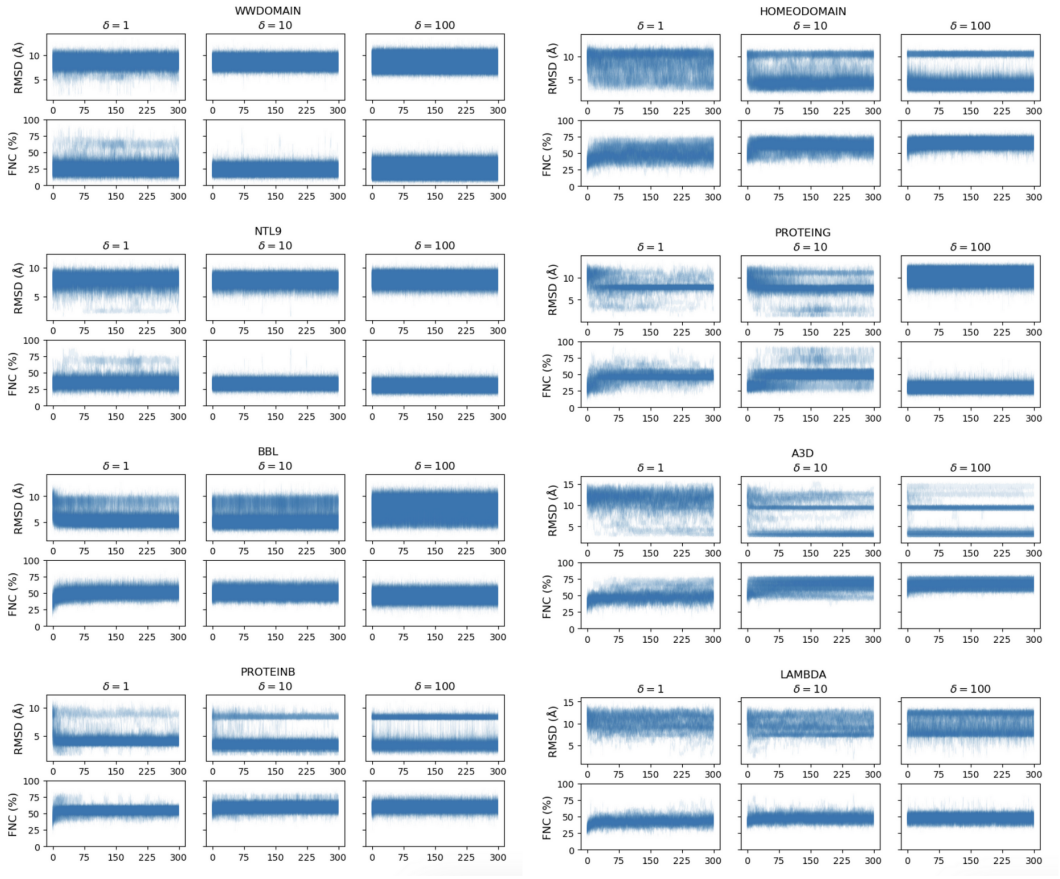

Figure 5: **Extended Plots for Folding Simulation**.

In Figure 5 we plot the evolution of RMSD and FNC over 300k model simulation steps. 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.