MARS-FM: GENERATIVE MODELING OF MOLECULAR DYNAMICS VIA MARKOV STATE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular Dynamics (MD) is a powerful computational microscope for probing protein function. However, the need for fine-grained integration and the long timescales of biomolecular events make MD computationally expensive. To address this, several generative models have been proposed to generate surrogate trajectories at lower cost. Yet, these models typically learn a fixed-lag transition density, causing the training signal to be dominated by frequent but uninformative transitions. We introduce a new class of generative models, **MSM Emulators**, which instead learn to sample transitions across discrete states defined by an underlying Markov State Model (MSM). We instantiate this class with MARKOV SPACE FLOW MATCHING (MARS-FM), whose sampling offers more than two orders of magnitude speedup compared to implicit- or explicit-solvent MD simulations. We benchmark MARS-FM's ability to reproduce MD statistics through structural observables such as RMSD, radius of gyration, and secondary structure content. Our evaluation spans protein domains (up to 500 residues) with significant *chemical* and *structural* diversity, including unfolding events, and enforces *strict sequence dissimilarity* between training and test sets to assess generalization. Across all metrics, MARS-FM outperforms existing methods, often by a substantial margin.

1 INTRODUCTION

Deep Learning has unlocked fast and accurate prediction of proteins' 3D structures (Jumper et al., 2021; Baek et al., 2021; Krishna et al., 2024; Abramson et al., 2024). However, these methods do not capture dynamic behavior of proteins (Lewis et al., 2024), whose conformational ensembles are governed by the Boltzmann distribution. To study such dynamics, the most reliable computational tool is **Molecular Dynamics (MD)** (Alder & Wainwright, 1959; Rahman, 1964; McCammon et al., 1977; Risken & Risken, 1996), which simulates atomic motion by integrating Netwon's Law. *Long* MD trajectories provide samples from the Boltzmann distribution and reveal the mechanisms of biomolecular interactions, a key asset in drug discovery (De Vivo et al., 2016). Nevertheless, MD is costly, as events in biology occur over timescales vastly longer than the simulation timestep. This challenge has spurred a range of *enhanced sampling* methods to accelerate dynamics, often through non-physical forces (Laio & Parrinello, 2002; Hamelberg et al., 2004; Jiang & Roux, 2010; Sabbadin & Moro, 2014).

Recent works have proposed to avoid the computational burden of MD by using generative flows to sample from the Boltzmann distribution (Noé et al., 2019; Zheng et al., 2024; Klein & Noe, 2024). A key subclass is the family of **MD Emulators** (MD-Emus), which learn to *emulate* MD-sampling by modeling the transition density associated with a fixed *lag time* τ (Klein et al., 2023; Schreiner et al., 2023; Nam et al., 2024; Diez et al., 2024; Costa et al., 2024). That is, MD-Emus are models that generate future conformations $x(t+\tau)$ conditioned on some input frame $x(t)$. At inference, these methods are applied autoregressively to produce surrogate MD trajectories. Jing et al. (2024b) advanced this approach by training a model to generate multiple frames jointly, all separated by a fixed interval τ. However, learning dynamics at a fixed lag time introduces key challenges. Short lag times limit the achievable speed-up, while long lag times can skip over important meta-stable states. More fundamentally, the training signal is dominated by frequent but uninformative transitions observed during the simulation, while high-barrier transitions that drive exploration remain underrepresented. As a result, MD-Emus may struggle to capture rare, large conformational changes (Figure 1, left).

Limitations of MD-Emus stem from the data imbalance in MD, whose temporal dynamic contains irrelevant high-frequency information. A common strategy to extract meaningful signal from MD

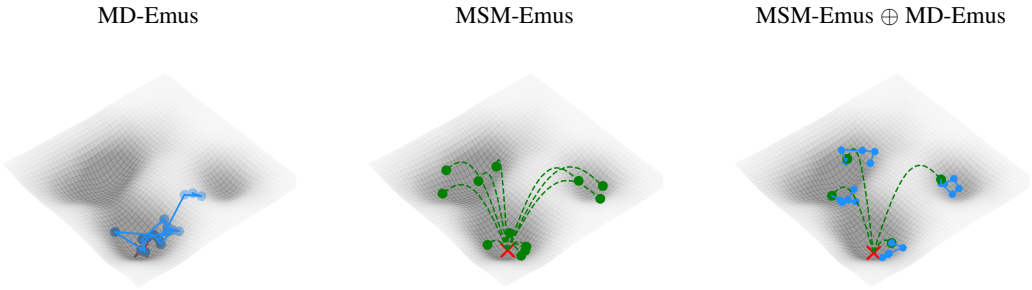

Figure 1: Comparison between existing approaches (MD-Emus) and our proposed novel class (MSM-Emus). MD-Emus learn transitions within a state (i.e. energy minima) well but could fail to generate transitions across different states (minima) since they are constrained by the data imbalance intrinsic to MD. Conversely, our framework learns to sample from the distribution induced by a Markov State Model (MSM). This modeling shift means that generative models are decoupled from temporal dynamics and can better learn to sample inter-state transitions. During sampling, MSM-Emus can generate conformations in parallel or be combined with existing MD-Emus to capture both large conformational changes as well as local dynamics within states.

through coarse-grained representations, is offered by **Markov State Models (MSMs)** (Noé et al., 2009; Prinz et al., 2011; Bowman et al., 2013). MSMs cluster frames into discrete *states* and describe dynamics via a Markov chain matrix T. While compressing the dynamical content, MSMs guarantee estimation of long-time statistics through the equilibrium distribution induced by T.

Main Contributions. We present **MSM Emulators (MSM-Emus)**, a new class of generative models that learn to sample from the Markov chain transition induced by an MSM, rather than emulating the noisy temporal dynamics induced by MD. This *modeling shift* has important implications as illustrated in Figure 1. Since target transitions now depend on state connectivity rather than specific observed paths, MSM-Emus benefit from much higher training diversity for learning rare conformational changes across high-energy barriers. In fact, MSM-Emus interpolate across discrete macroscopic states associated with large conformational changes, such as folding or unfolding, rather than among specific frames, which is a more robust and generalizable signal across different proteins. During sampling, frames are more easily decorrelated and autoregressive calls are greatly reduced, mitigating compounding error effects. We showcase this class with MARKOV SPACE FLOW MATCHING (MARS-FM), a novel framework optimized using Flow Matching (Lipman et al., 2023).

Evaluation with chemical and structural diversity. MD-Emus have usually been evaluated on small peptides or datasets with limited chemical (Lindorff-Larsen et al., 2011; Majewski et al., 2023) or structural (Vander Meersche et al., 2024) diversity. To address this gap, we evaluate MARS-FM on MD-CATH (Mirarchi et al., 2024), a large-scale dataset of thousands of protein domains with sequence length up to 500 residues. In particular, by leveraging the highest-temperature replica, we are able to test MARS-FM's ability to capture large conformational changes, such as *unfolding*. To assess *generalization*, we adopt the protocol of Lewis et al. (2024) and use MMSEQS2 (Steinegger & Söding, 2017) with maximum sensitivity to ensure that **test proteins share no more than** 20% **sequence similarity with any protein in the training set**. We benchmark MARS-FM using structural observables reported in Jing et al. (2024b); Mirarchi et al. (2024), including RMSD, radius of gyration, and secondary structure content, and show that it consistently outperforms MD-Emus—often by a large margin. Crucially, MARS-FM explores the target distribution much **more efficiently than both MD and MD-Emus**, generating conformations across different states even in low-sample regimes.

2 PRELIMINARIES AND SETTING

A molecular system of N atoms, in thermal equilibrium at temperature T, explores conformations according to the Boltzmann distribution μ, i.e. $q \sim \mu(q) \propto \exp(-\mathcal{H}(q)/(k_B T))$ with $q = (x, \xi) \in \mathcal{X}$ denoting the system's position and velocity, $\mathcal{H}$ the Hamiltonian, and k_B the Boltzmann constant. We focus on proteins, whose *motions*—such as local unfolding or cryptic pocket formation—arise from structural fluctuations. To *identify* these conformational changes and *quantify* their frequencies (free energy), we aim to sample structures from the Boltzmann distribution μ. This enables the measurement of **observables** ϕ—functions defined on the phase space $\mathcal{X}$—via expectations $\mathbb{E}_{x\sim\mu}[\phi(x)]$.

Since direct sampling from μ is intractable, a widespread alternative is to use **Molecular Dynamics (MD)**. MD simulates a continuous-time Markov process, designed to be *ergodic* w.r.t. μ. That is, for any observable, time averages over a long trajectory approximate expectations under μ (Schütte et al., 2023). A common formulation of MD is via **Langevin dynamic**: given a system with potential energy $\mathcal{U}$, position and velocity of an atom i with mass m_i, are updated as follows

$$\dot{x}_i = \xi_i, \quad m_i \dot{\xi}_i = -\nabla_x \mathcal{U}(x) - \gamma m_i \xi_i + \sigma \dot{W}_t, \tag{1}$$

with γ and σ controlling friction and thermal noise. In practice, we discard the kinetic component and focus on the trajectories of positions $t \mapsto x(t)$. While MD is a powerful tool, its computational cost is a major bottleneck. Many biologically relevant events occur on millisecond timescales, while Langevin dynamics requires integration with time steps of the order of femtoseconds, making long simulations prohibitively expensive. Crucially, meta-stable states of a protein are often separated by high-energy barriers, leading to MD simulations wasting time trapped in deep local minima (Wales, 2005).

Problem Formulation. Given an initial protein conformation $x(0)$, MD produces an ensemble of structures by integrating eq. (1) and saving frames at regular intervals. Since MD is expensive, we aim to use generative models to sample surrogate distributions at a fraction of the cost. To enable generalization across proteins, a *single* model is trained using available MD trajectories from multiple sequences. At inference, we provide sequence and input conformation from an unseen protein, and the goal is to generate samples that are statistically indistinguishable from MD trajectories with respect to a set of physical observables ϕ—that is, the model should match the distribution over structural or thermodynamic quantities that practitioners care about. Next, we review existing approaches that attempt to address this problem by sampling from a fixed transition density and discuss their limitations.

3 MD Emulators and the Challenges of Fixed Lag Time Transitions

A key object describing MD sampling is the **transition density** $y \sim p_\tau(y|x)$ associated with a *lag time* τ, which represents the probability of a state x evolving to state y within time τ. Concretely, this is estimated from MD trajectories by examining future transitions $x(t+\tau)$ given an input frame $x(t)$. **MD Emulators (MD-Emus)** are a class of conditional generative models trained to approximate $p_\tau(\cdot|x(t))$ from MD data (Fu et al., 2023; Klein et al., 2023; Schreiner et al., 2023; Nam et al., 2024; Diez et al., 2024; Costa et al., 2024). Specifically, MD-Emus learn to generate future conformations $y \sim p^\theta_\tau(y|x(t))$ given $x(t)$. Common training approaches include Normalizing Flows (Rezende & Mohamed, 2015; Chen et al., 2018), Flow Matching (Lipman et al., 2022; Tong et al., 2023; Albergo et al., 2023), or Score Matching (Ho et al., 2020; Song et al., 2021).

These models train a neural network v_θ—*conditioned* on sequence and an input conformation $x(t)$—to match the empirical distribution of transitions observed in the MD data. Jing et al. (2024b) refined such an approach in MDGen, by generating the next K future conformations at once. Explicitly, MDGen learns to sample from the joint density $\mathbf{y} = (y_1, \dots y_K) \sim p_{\tau,K}(\mathbf{y}|x) = \prod_i p_\tau(y_{i+1}|y_i)$, conditioned on $y_0 = x$. In the following, we assume we are given an MD-Emu that is trained to approximate $p_{\tau,K}$.

3.1 Limitations of MD-Emus

Training. One of the key challenges in learning from MD trajectories is the *data imbalance* between frequent but uninformative transitions in an energy minima, and rare but key transitions across different minima (Wales, 2005). To make things more concrete, consider the case where a protein MD trajectory visits two (or more) macroscopic states S_A and S_B, such as folded and unfolded conformations, separated by high-energy barriers. Transitions $S_A \to S_B$ are rare but critical as they reveal conformational flexibility and functional dynamics. However in any given trajectory interval of length $K\tau$, the system is far more likely to remain in single state, than to cross an energy barrier. As a result, training batches for MD-Emus are typically dominated by high-frequency, low-information intra-state transitions. This data imbalance limits the model's ability to learn conditional distributions describing meaningful state changes. For example, to learn how to transition from folded to unfolded states, the model must observe actual transitions $S_A \ni x(t) \to x(t+K\tau) \in S_B$ in the data—but such events are rare. This sparsity constrains the diversity and efficiency of training data and affects generalization, as MD-Emus learn to replicate fine-grained temporal resolution rather than large conformational changes.

Inference. MD-Emus need to be used autoregressively to generate *long* trajectories. This may lead to error accumulation, causing samples to progressively drift away from the data manifold. Nam

et al. (2024) introduced a refinement step to address this, yet this comes at the expense of training a second network. Besides, using a larger K does not fully resolve the problem, as the number of required samples often exceeds the maximum window feasible during training for larger proteins.

The challenges faced by MD-Emus stem from fixed lag time transitions often being uninformative and failing to capture meaningful conformational changes across different energy minima. To overcome these limitations, we first need to apply a *coarse-grained representation* to MD data so to discard high-frequency information. To this aim, we rely on Markov State Models (MSMs) and introduce a novel class of generative flows to learn transitions directly among states (rather than frames).

4 The new class of MSM Emulators

Markov State Models (MSMs) provide a coarse-grained representation that removes high-frequency time signal while still capturing long-timescale statistics (Noé et al., 2009; Pande et al., 2010; Prinz et al., 2011; Husic & Pande, 2018). In MSMs, frames $x(t)$ are assigned to discrete states $S_1, \dots, S_M$ and the dynamics is modeled as a Markov chain over these states. Given an interval τ, one estimates a **transition matrix** T, where T_{ij} is the probability of transitioning from S_i to S_j within time τ:

$$\mathsf{T}_{ij} = \frac{C_{ij}}{\sum_k C_{ik}}, \quad C_{ij} = |\{x(t) \in S_i : x(t+\tau) \in S_j\}|. \tag{2}$$

Explicitly, that means we consider a probability density $p_{\mathsf{T}}(\cdot|x(t))$ satisfying

$$\int_{S_j} p_{\mathsf{T}}(y|x(t))dy = \mathsf{T}_{ij}, \quad \forall x(t) \in S_i,\ i,j \in \{1,\dots,M\}. \tag{3}$$

In its simplest form, MSMs assumes a uniform density within each state. That is, given $x(t) \in S_i$ and $y \in S_j$, $p_{\mathsf{T}}(y|x(t))$ depends only on the identity of the clusters, but not on the specific conformations. *This coarse-graining enables robust estimation of long-timescale dynamics, even from limited transition data*. In principle, one could initialize *short* MD simulations (of length τ) across discrete states $\{S_i\}$, and propagate their dynamics via T. However, sampling from the states of an underlying (unseen) MSM remains a challenge. We propose a class of generative models that addresses this issue.

Definition 1 MSM Emulators *(**MSM-Emus**) are generative models that are trained to sample conformations from $p_{\mathsf{T}}(\cdot|x(t))$, where T is the Markov chain transition matrix of a given MSM.*

By matching the distribution $p_{\mathsf{T}}(\cdot|x(t))$—that depends only on the MSM states—MSM-Emus bypass many of the data scarcity and imbalance issues that affect MD-Emus §3.

Increased sample diversity. MSM-Emus *decouple* the learning objective from the specific transitions observed in the simulation. Rather than learning frame-to-frame transitions $x(t) \to x(t+\tau)$, they learn *state-to-state transitions* $S_i \to S_j$, enabling broader generalization (as validated in §5). Unlike MD-Emus, which attempt to replicate the exact temporal dynamics at fixed lag time τ, MSM-Emus focus on capturing *macroscopic* transitions identified by the MSM. In the scenario described in §3, MSM-Emus learn to generate transitions $S_A \ni x \to y \in S_B$ from diverse starting points $x \in S_A$ beyond those explicitly visited by the MD simulation. Crucially, the kinetics defined by the underlying MSM remains intact: during training, transitions are sampled from the distribution T, ensuring that the global dynamical behavior is faithfully preserved.

Fast exploration. At inference, MSM-Emus can explore the energy landscape by generating samples $y \sim p_{\mathsf{T}}(y|x(0))$, without being bound by fine-grained temporal dynamics. This significantly improves *sampling efficiency* (as visualized in Figure 1 and validated in §5). Additionally, as many conformations can now be generated in *parallel*, MSM-Emus reduce compounding errors due to autoregressive calls.

4.1 A representative framework: Markov Space Flow Matching

We introduce Markov Space Flow Matching (MarS-FM), a novel representative framework of the MSM-Emu class. First, we describe concrete MSM instantiations defined over MD trajectories.

Constructing MSMs. Specific versions of MarS-FM are instantiated by first building an MSM. To this aim, we use standards tools (Scherer et al., 2015; Hoffmann et al., 2021).

The states $\{S_i\}$ are defined in a lower-dimensional space of collective variables. A common approach for dimensionality reduction is *Time-lagged Independent Component Analysis* (TICA) (Pérez-Hernández et al., 2013; Wu et al., 2017), which identifies directions $\mathbf{w}_j$ that maximize the autocorrelation of $\mathbf{w}_j^\top x(t)$ at lag time τ. Alternatively, clustering can be applied directly to observables such as radius of gyration and fraction of secondary structures (Figure 2). Given states, the Markov chain transition matrix T is estimated at a lag time τ. Two key observations are worth emphasizing. First, the MSM construction is performed **once per dataset** and remains *fixed* across all training trajectories, ensuring minimal computational overhead. Second, since MARS-FM learns to interpolate between MSM states beyond the transitions directly observed in MD, we can afford to select a larger τ than MD-Emus without sacrificing the data availability. We emphasize that MSMs are defined as pre-processing of training data; importantly, **no MSM information is provided during inference**. Additional details on how we construct MSMs are reported in Appendix B.

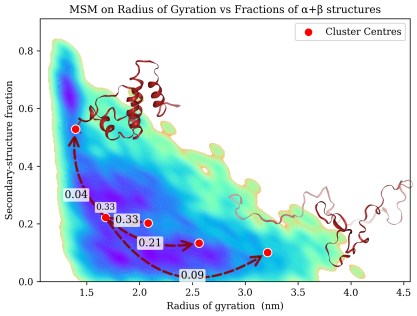

Figure 2: Clustering of MD conformations of protein 3ma5A00 from MD-Cath. We report cluster centres and Markov chain transitions from one representative state. We note how MSM states capture large structural differences (folded vs unfolded).

Representation. To generate protein conformations, MARS-FM adopts the same SE(3) representation as (Jumper et al., 2021; Jing et al., 2024b). Each residue ℓ of frame $x(t)$ is represented as

$$\chi^\ell(t) = (q^\ell(t), r^\ell(t), (\cos(\theta_k^\ell(t)), \sin(\theta_k^\ell(t))_{k=1}^7), \tag{4}$$

where $q^\ell(t) \in \mathbb{R}^4$ is a unit quaternion describing a rotation and $r^\ell(t) \in \mathbb{R}^3$ is a translation, while $\theta_k^\ell(t)$ represent the k torsion angles. Given an input $x(t)$, the target conformation y is represented as an offset in roto-translational space. To model a vector field v_θ transporting a source distribution to the target distribution $p_\mathsf{T}(\cdot|x(t))$, we build on the MDGen architecture (Jing et al., 2024b). Specifically, v_θ leverages DiT-style blocks (Peebles & Xie, 2023) and is *conditioned* on protein sequence a and the current conformation $x(t)$ via IPA layers (Jumper et al., 2021)—see Appendix A for a complete description. For notational simplicity, we omit the explicit conditioning on the protein sequence a.

Training objective. We train MARS-FM using **Flow Matching**, which learns a time-dependent vector field v_θ to transport a source distribution $p_0 = \mathcal{N}(0,1)$ to the target distribution $p_1(\cdot|x(t)) = p_\mathsf{T}(\cdot|x(t))$. Concretely, given an input frame $x(t) \in S_i$, we first draw a state S_j according to the probability distribution $j \mapsto \mathsf{T}_{ij}$ and then sample $x_1 \in S_j$ uniformly within such a state. Given sample pairs $x_0 \sim p_0$ and $x_1 \sim p_1$, we define interpolations $[0,1] \ni s \mapsto x_s$ connecting x_0 to x_1. We optimize v_θ by minimizing velocity mismatch along conditional paths $s \mapsto p_s(\cdot|x_0, x_1)$ based on these interpolations (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022). That is, we consider the training loss

$$\mathcal{L}_{\text{MARS-FM}}(\theta) = \mathbb{E}_{i\sim[M],x(t)\sim S_i,s\sim[0,1],x_0\sim p_0(x_0),x_1\sim p_\mathsf{T}(x_1|x(t))} \|v_\theta(s, x_s; x(t)) - \dot{x}_s\|^2, \tag{5}$$

where $\dot{x}_s$ is the velocity of the conditional path. Crucially, we have emphasized how we select each state S_i uniformly, before conditioning on $x(t)$. This state-based sampling ensures that rare states are encountered more frequently during training. In contrast, standard MD-Emus draw frames uniformly from the trajectory, which often biases training toward common intra-state transitions. A similar strategy was used by Costa et al. (2024), though their model still approximates a fixed lag time transition. Further implementation and batching details are provided in Appendix A.

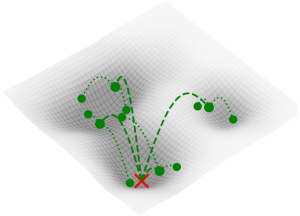

Figure 3: Hierarchical sampling used for MARS-FM.

Sampling. Following standard evaluation for MD-Emus, we assume that at inference, we have sequence a and an input conformation $x(0)$ from an unseen protein. We explore two sampling strategies.

1. Tree Sampling. We apply MARS-FM following a hierarchical sampling scheme (Figure 3). We first generate n frames $\{y_1, \ldots, y_n\} \sim p_\mathsf{T}(\cdot|x(0))$ in *parallel*. Next, we generate n frames from $p_\mathsf{T}(\cdot|y_i)$ for each y_i. We continue extending the depth of the tree-sampling scheme based on our sample budget.

2. MARS-FM $\oplus$ *MD-Emu.* We combine MARS-FM with MDGen (Jing et al., 2024b). We first sample conformations using MARS-FM and then use MDGen to generate shorter trajectories from each of these points, separately. This approach *mirrors the MSM paradigm*, where statistics are inferred by initiating short simulations from different states. We introduce such a hybrid scheme, as in certain workflows temporal fidelity may also be relevant (De Vivo et al., 2016).

For both cases, most conformations can be generated in parallel hence reducing autoregressive sampling. Crucially, as samples of MARS-FM are decoupled from temporal dynamics, they can explore the target distribution more efficiently (see Figure 5).

5 EXPERIMENTS

A key condition for a generative model to replace MD-sampling, is that measurements of observables under the generated samples match those under MD. For this reason, in our experiments, we compare distribution of observables computed along MD trajectories and those computed along conformations generated by the underlying model. We rely on observables used in previous generative works (Jing et al., 2024a;b) as well as additional ones reported for analyzing the MD-Cath dataset (Mirarchi et al., 2024).

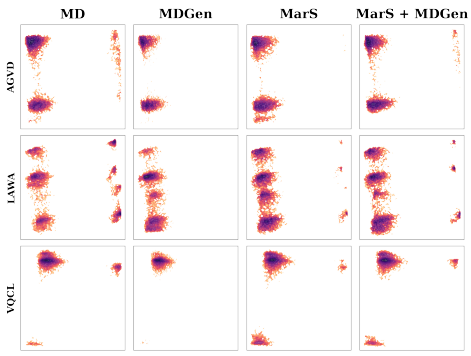

Figure 4: TICA plot for 3 random peptides in the test set, comparing MD ground-truth, MDGen, MARS (ours) and MARS + MDGen (ours). Our frameworks explore modes that are otherwise entirely ignored by MDGen. Similar plots are reported in Appendix C.

Variants of MARS-FM. We first construct an MSM—dependent on the dataset but invariant across trajectories, as described below—and then train MARS-FM to sample from $p_{\mathsf{T}}(\cdot|x(t))$. Yet *no information about MSMs is provided during inference*. As explained in §4, during sampling, we consider two variants: MARS-FM $\oplus$ MDGen and MARS-FM.

Baselines. In our experiments, we take MDGen (Jing et al., 2024b) as the main representative of the MD-Emu class. Primarily, this is due to MDGen being the most versatile variant of MD-Emus due to the choice of the window K of consecutive transitions at resolution τ. For larger systems, we additionally compare against BioEmu (Lewis et al., 2024), noting specific caveats discussed in Appendix B.7. Finally, we point out that other methods are either *not* scalable for MD-Cath domain (Klein et al., 2023; Schreiner et al., 2023) or have only been validated within the training distribution (Costa et al., 2024) (see Appendix B.7 for further discussion).

5.1 PRELIMINARY INVESTIGATION: TETRAPEPTIDES

We first adopt the dataset of tetrapeptides that provided the main evaluation for MDGen (Jing et al., 2024b). This consists of ∼3000 trajectories in training, 100 in validation and 100 in test. Each trajectory is simulated in explicit solvent for 100 ns and conformations are saved every 10 ps. That is, for any tetrapeptide in the test set, we have a held-out MD distribution of 10^4 frames. Accordingly, we generate distributions of 10^4 frames using both MD-Emus (MDGen) and MSM-Emus (MARS-FM).

MSM construction. When training MARS-FM, we need to define MSMs over the MD trajectories. As this dataset contains exhaustive simulations relative to the size of the systems, we rely on TICA (Pérez-Hernández et al., 2013), selecting the minimal number of TICA coordinates whose cumulative kinetic variance exceeds 95%. We then apply k-means clustering to these coordinates, obtaining 100 microstates, which we further group into 10 metastable states using the PCCA+ spectral clustering method (Röblitz & Weber, 2013). This procedure yields the final MSMs constructed with a lag time of 100 ps. Further details on MSM construction are provided in Appendix B.

Evaluation and metrics. During sampling, we report the Jensen-Shannon Divergence (JSD) over different observables. Namely, we follow the same metrics as in MDGen Jing et al. (2024b): (i) Torsional angles (of both backbone and sidechains); (ii) Projection onto the slowest modes as identified by TICA; (iii) Equilibrium distributions induced by MSMs built over the test peptides.

Table 1: JSD ↓ between sampled and ground-truth distributions for Tetrapeptides. Results based on sampled trajectories of 10^4 conformations compared to ground-truth distributions.

	Torsions (bb)	Torsions (sc)	Torsions (all)	TICA-0	TICA-0,1 joint	MSM states	Macrostate MAE
MD (Oracle)	0.10	0.06	0.08	0.20	0.27	0.21	—
MDGen-1000	0.13	**0.09**	0.11	0.23	0.32	0.23	1.13
MDGen-200	0.14	0.10	0.12	0.24	0.33	0.27	1.12
MARS-FM ⊕ MDGen-200 (ours)	0.12	**0.09**	**0.10**	**0.21**	0.30	0.23	0.83
MARS-FM (ours)	**0.11**	**0.09**	**0.10**	**0.21**	**0.29**	**0.22**	**0.63**

Inspired by Lewis et al. (2024), we also report the Macrostate MAE (mMAE), which measures the mean absolute error between model-generated and ground-truth MD free energies across metastable states. The mMAE provides a direct measure of the model's accuracy in reproducing metastable state distributions. Further computational details and definitions are presented in Appendix B.

Results. For the baselines, we train MDGen with window size $K = 200$ and $K = 1000$, and use the former combined with MARS-FM as per our hybrid-scheme described in §4. Results in Table 1 illustrate that MARS-FM achieves same or better performance than baselines. In particular, errors are often comparable with those encountered by different replicas of MD trajectories (oracle performance). Noticeably, a large improvement is achieved on the Macrostate MAE, highlighting that MARS-FM can better sample from rare meta-stable states of the underlying TICA space of unseen peptides. This is also confirmed in Figure 4—additional TICA plots are reported in Appendix C. In general though, we expect MD-Emus and MSM-Emus to be comparable for this dataset, as we have very limited chemical diversity and no large domain motion due to the size of the systems.

5.2 MD-CATH: PROTEINS WITH SEQUENCE DISSIMILARITY AND LARGE DOMAIN MOTIONS

Our main evaluation is based on the MD-Cath dataset from Mirarchi et al. (2024). This includes 5398 domains up to 500 residues (*average number of residues* 137), simulated at 5 different temperatures, spanning 5 random replicas per setting. The average length of each trajectory is 464 ns and frames are saved at 1 ns intervals. The dataset is already built to ensure protein diversity by considering non-homologous domains at the $S20$ (20%) homology level—further details are reported in Appendix B.

MSM construction. MSMs are constructed independently for each domain using k-means clustering directly applied to normalized observables identified as critical by the original dataset analysis (Mirarchi et al., 2024). Specifically, we consider two features: (i) the radius of gyration and (ii) the fraction of residues in α-helical and β-sheet secondary structures—additional details are reported in Appendix B. We cluster these normalized features into 10 metastable states, providing a physically meaningful partitioning of the conformational space. This yields the final MSMs constructed with a lag time of 50 ns.

Large domain motions. To validate the ability of MARS-FM to generate large conformational changes, our main evaluation is performed over the highest-temperature replica (450 K), in which proteins exhibit ***unfolding***. In fact, proteins are quite stable at lower temperatures over the simulated timescales, meaning that often no significant conformational change has occurred—as reported in Mirarchi et al. (2024)—which would result in an easier generative task. Nonetheless, for a more complete evaluation, we also train MARS-FM on lowest-temperature replica and sample conformations in this regime–results are reported in Appendix C.

Assessing generalization. First, we consider a random training-validation-test split partitioning the total number of domains according to 80%-10%-10%. We then follow the strategy in Lewis et al. (2024) and use MMSEQS2 (Steinegger & Söding, 2017) with *highest sensitivity setting* to **filter out any test protein sharing more than** 20% **sequence similarity with any training protein** (note that Lewis et al. (2024) use a more forgiving 40% threshold). As a result, our test set consists of 495 domains that are meaningfully diverse from those whose trajectories have been used during training.

Metrics and Observables. For evaluation, we focus on observables studied in the original dataset analysis of Mirarchi et al. (2024), namely the radius of gyration and fractions of residues in secondary structures (α-helices and β-sheets). We measure the forward KL divergence to quantify the model's exploration capabilities and the Jensen-Shannon Divergence (JSD) for general distribution alignment. Inspired by Jing et al. (2024a), we also quantify ensemble flexibility using the Pearson correlation (r) computed on pairwise backbone Root Mean Square Deviation (RMSD), global Root Mean Square Fluctuation (RMSF), and per-target RMSF. We additionally evaluate folding free energies via the mean absolute error (ΔG_{fold} MAE). Lastly, we calculate the JSD between reconstructed MSMs and

Table 2: Pearson $r \uparrow$ for RMSD and RMSF, forward KL divergence $\downarrow$, JSD $\downarrow$ of gyration radius, fraction of secondary structures, and MSMs distribution, and folding free energy MAE (kcal/mol) $\downarrow$. Results based on sampled trajectories of 500 conformations compared to ground-truth distributions and averaged over 5 inference runs.

	Pairwise RMSD r	Global RMSF r	Per target RMSF r	Gyration Radius KL	Gyration Radius JSD	Secondary Structures KL	Secondary Structures JSD	MSM JSD	ΔG_{fold} MAE
MD (Oracle)	0.82 ± .014	0.82 ± .012	0.90 ± .005	0.58 ± .020	0.07 ± .001	0.61 ± .048	0.07 ± .003	0.19 ± .006	0.90 ± .046
MDGen-100	0.42 ± .047	0.49 ± .018	0.64 ± .007	1.13 ± .025	0.15 ± .003	1.17 ± .045	0.18 ± .005	0.29 ± .005	1.21 ± .010
MDGen-20	0.40 ± .016	0.38 ± .013	0.38 ± .013	0.89 ± .010	0.20 ± .002	1.85 ± .024	0.30 ± .002	0.43 ± .009	1.44 ± .026
MDGen-100 (in parallel)	0.42 ± .011	0.54 ± .003	0.69 ± .009	2.15 ± .019	0.22 ± .001	1.83 ± .015	0.20 ± .002	0.39 ± .014	2.14 ± .015
MDGen-20 (in parallel)	0.43 ± .002	0.55 ± .001	0.70 ± .005	2.71 ± .005	0.26 ± .001	2.96 ± .008	0.28 ± .001	0.46 ± .002	3.68 ± .007
BioEmu	0.25 ± .002	0.41 ± .004	0.66 ± .001	3.83 ± .011	0.40 ± .001	4.17 ± .015	0.41 ± .001	0.51 ± .001	4.67 ± .004
MarS-FM ⊕ MDGen-20	0.63 ± .004	0.69 ± .002	0.83 ± .001	0.98 ± .017	0.13 ± .002	**0.73** ± .005	**0.11** ± .001	0.24 ± .002	**1.02** ± .002
MarS-FM	**0.65** ± .004	**0.71** ± .003	**0.89** ± .001	**0.55** ± .002	**0.10** ± .001	0.93 ± .010	0.13 ± .001	**0.19** ± .004	1.05 ± .003

reference MSMs, providing a direct measure of the ability to explore states of domains dissimilar from those observed in training. Additional details on metrics are provided in Appendix B.

Extended baselines. We benchmark different variants of MDGen to highlight how advantages of our framework *cannot be simply replicated by changing the lag time or altering the sampling approach*. As such, we train MDGen with window size $K = 20$ and $K = 100$, corresponding to a total lag time of 20 ns and 100 ns, respectively. Additionally, to demonstrate that the improvements of MarS-FM cannot be simply ascribed to a reduction of autoregressive calls, we also evaluate MDGen in *parallel*, meaning that we sample conformations only conditioned on the input frame. Finally, we also report the performance of one MD replica against 4 held-out ones (and average over all possible 5 combinations) to be treated as *oracle* performance.

Results. We report our evaluation in Table 2. Both hybrid sampling (MarS-FM ⊕ MDGen) and hierarchical sampling (MarS-FM) significantly improve over all extended MDGen baselines, often by a large margin. This confirms that our framework can extrapolate over unseen large conformational changes better than MD-Emus. Crucially, MarS-FM can even match the oracle performance in the reconstruction of the unseen underlying MSM. This validates how the framework is capable to generalize across unseen MSMs and points to such a signal being easier and more robust than the one derived from fixed-lag time transitions. As all test proteins share no more than 20 % sequence similarity to any training protein, we again emphasize that the results in Table 2 provide a stringent test for the generative models to sample conformations over meaningfully different domains.

Ablation: Changing sample budgets. Finally, we report results across different sample budget in Table 3. Namely, we consider the same metrics as above but compare distributions obtained by generating 100 and 1000 samples, respectively. In the low-sample regime, MarS-FM also **surpasses the oracle performance** as given by the first 100 frames (at 1 ns resolution) sampled by MD. This confirms that by decoupling the training objective from temporal dynamics, MarS-FM can sample large conformational changes much more efficiently as the model interpolates directly across states rather

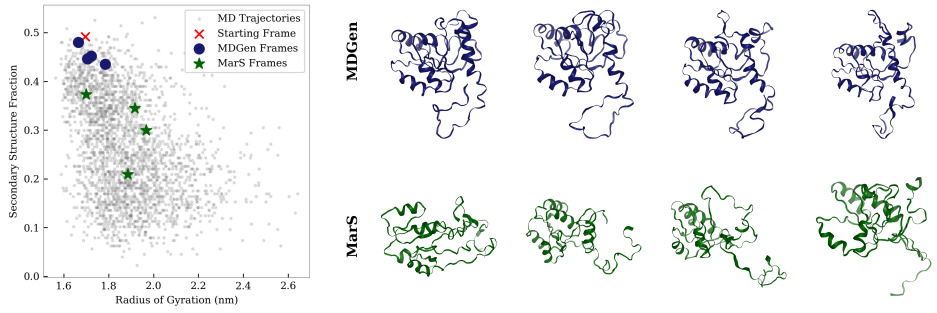

Figure 5: First 4 samples generated by MDGen and MarS-FM for the domain 2ynmD03 in the test set. As MarS-FM interpolates among states independently of temporal dynamics, it can explore the energy landscape more efficiently. In fact, the secondary structure content varies significantly among these 4 samples (note that there is no ordering as they are generated in parallel). Conversely, MDGen samples all belong to the same energy minimum which reduced sampling efficiency and exploration.

Table 3: Pearson $r \uparrow$ for RMSD and RMSF, forward KL divergence $\downarrow$ and JSD $\downarrow$ of gyration radius, fraction of secondary structures, MSMs distribution, and folding free energy MAE (kcal/mol) $\downarrow$. Results based on sampled trajectories of 100 / 1000 conformations compared to ground-truth distributions and averaged over 5 inference runs.

	Pairwise RMSD r	Global RMSF r	Per target RMSF r	Gyration Radius KL	Gyration Radius JSD	Secondary Structures KL	Secondary Structures JSD	MSM JSD	ΔG_{fold} MAE
MD (Oracle)	0.65 / 0.89	0.70 / 0.87	0.77 / 0.92	2.19 / 0.32	0.18 / 0.05	2.75 / 0.26	0.22 / 0.05	0.49 / 0.12	2.40 / 0.80
MDGen-100	0.34 / 0.28	0.46 / 0.32	0.60 / 0.46	3.66 / 0.78	0.31 / 0.14	3.60 / 1.49	0.29 / 0.26	0.51 / 0.27	2.58 / 1.52
MDGen-20	0.57 / 0.28	0.62 / 0.23	0.61 / 0.20	2.48 / 1.37	0.22 / 0.33	2.08 / 2.34	0.19 / 0.39	0.48 / 0.51	1.52 / 2.01
MDGen-100 (in parallel)	0.37 / 0.43	0.46 / 0.56	0.62 / 0.71	3.62 / 1.84	0.31 / 0.21	3.51 / 1.54	0.28 / 0.19	0.52 / 0.38	2.68 / 2.12
MDGen-20 (in parallel)	0.41 / 0.43	0.53 / 0.55	0.67 / 0.71	3.56 / 2.50	0.30 / 0.26	3.87 / 2.75	0.31 / 0.27	0.53 / 0.46	3.77 / 3.65
BioEmu	0.23 / 0.26	0.40 / 0.42	0.64 / 0.67	4.75 / 3.55	0.43 / 0.39	5.08 / 3.91	0.44 / 0.41	0.55 / 0.41	4.82 / 4.62
MARS-FM ⊕ MDGen-20	0.59 / 0.64	0.66 / 0.71	0.79 / 0.84	1.99 / 0.77	0.20 / 0.12	**1.85 / 0.61**	**0.18 / 0.11**	0.43 / 0.23	1.38 / **1.02**
MARS-FM	**0.60 / 0.65**	**0.68 / 0.71**	**0.84 / 0.90**	**1.74 / 0.42**	**0.18 / 0.09**	1.92 / 0.95	**0.18** / 0.14	**0.42 / 0.17**	**1.25** / 1.20

than across frames separated by a time interval. We illustrate this phenomenon in Figure 5. Our framework also offers optimal performances at higher sapling regimes, as it significantly reduces autoregressive calls hence mitigating compounding error effects. Overall, the results in Table 3 confirm that MARS-FM improves upon the baselines across different sample budgets due to (i) Its ability to sample large-domain transitions independent of temporal ordering; (ii) Reduced autoregressive sampling.

Sampling speed. A key requirement for generative models to serve as practical replacements for MD is that they offer substantial computational speed-ups. Table 4 summarizes the wall-clock time required to generate the equivalent of 500 ns of trajectory (500 conformations) for a 159-residue protein (domain 4dhkB00) on an NVIDIA H100. For a fair comparison, we report MD in *implicit solvent*, which is substantially more efficient than explicit-solvent simulations. For such a system, implicit MD runs at $\approx$ 2400 ns/day on the same hardware for a 159-residue protein (OpenMM Team, 2025). Under these conditions, MARS-FM provides a computational speed-up of $600\times$.

Table 4: Wall-clock time (in seconds, $\downarrow$) to generate the equivalent of 500 ns trajectory (500 conformations) for a 159-residue protein.

Method	Time (s) $\downarrow$
MD (implicit solvent)	$\approx$ 18,000 (= 5h)
MDGen-100	31.70 ± 0.21
MDGen-20	100.87 ± 2.24
MARS-FM ⊕ MDGen-20	19.79 ± 0.04
MARS-FM	30.34 ± 0.08

6 CONCLUSIONS

Related works. Beyond MD-Emus, our work is related to the class of Boltzmann Generators (BGs), normalizing flows that are trained via the potential energy (Noé et al., 2019; Wirnsberger et al., 2020; Köhler et al., 2021; Rizzi et al., 2021; Garcia Satorras et al., 2021; Rizzi et al., 2023; Midgley et al., 2023; Klein & Noe, 2024; Tan et al., 2025). However, BGs have shown limited scalability beyond small peptides. Alternatively, Jing et al. (2024a); Lewis et al. (2024) proposed sequence-to-structure generative flows trained on MD data, after ignoring any temporal ordering. These methods though suffer from the same data inbalance issues affecting MD-Emus. Learning to sample diverse protein conformations was also investigated in (Jing et al., 2023; Zheng et al., 2024; Lu et al., 2024) and is related to works that perturbed AlphaFold at the MSA level (Del Alamo et al., 2022; Wayment-Steele et al., 2024) or refined its predictions via experiments (Maddipatla et al., 2025). However, these works may fail to attain *quantitative* distributional matching. Finally, MSMs have been used, for a single peptide dataset, to reweigh samples when fine-tuning a generative model in Lewis et al. (2024).

Limitations and Future Works. In general, public MD data of protein dynamics is limited, particularly for long unbiased simulations. This generally hinders the development of generative models—a key reason as to why to assess large domain motions with sufficient chemical diversity we had to focus on higher-temperature simulations. In terms of scope, in this work we have focused on protein representation. Natural next steps would entail extending MSM-Emus to complexes, for example protein-ligand ones, and inorganic systems. MARS-FM requires an input 3D structure at inference. To further accelerate sampling and increase applicability of the framework, we will study how to leverage recent sequence-to-structure models to be able to generate protein conformations starting from sequence only. Finally, it would be interesting to explore whether MARS-FM can be combined with MD simulations, for example by sampling different input conformations across an underlying MSM, and then initializing shorter MD simulations in parallel.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024. (Cited on page 1)

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. (Cited on page 5)

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. (Cited on page 3)

Berni J Alder and Thomas Everett Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959. (Cited on page 1)

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. (Cited on page 1)

Gregory R Bowman, Vijay S Pande, and Frank Noé. *An introduction to Markov state models and their application to long timescale molecular simulation*, volume 797. Springer Science & Business Media, 2013. (Cited on page 2)

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. (Cited on page 3)

Allan dos Santos Costa, Ilan Mitnikov, Franco Pellegrini, Ameya Daigavane, Mario Geiger, Zhonglin Cao, Karsten Kreis, Tess Smidt, Emine Kucukbenli, and Joseph Jacobson. Equijump: Protein dynamics simulation via so (3)-equivariant stochastic interpolants. *arXiv preprint arXiv:2410.09667*, 2024. (Cited on pages 1, 3, 5, 6, and 19)

Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9):4035–4061, 2016. (Cited on pages 1 and 6)

Diego Del Alamo, Davide Sala, Hassane S Mchaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *Elife*, 11:e75751, 2022. (Cited on page 9)

Juan Viguera Diez, Mathias Schreiner, Ola Engkvist, and Simon Olsson. Boltzmann priors for implicit transfer operators. *arXiv preprint arXiv:2410.10605*, 2024. (Cited on pages 1 and 3)

Xiang Fu, Tian Xie, Nathan J Rebello, Bradley Olsen, and Tommi S Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *Transactions on Machine Learning Research*, 2023. (Cited on page 3)

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021. (Cited on page 9)

Donald Hamelberg, John Mongan, and J Andrew McCammon. Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. *The Journal of chemical physics*, 120 (24):11919–11929, 2004. (Cited on page 1)

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020. (Cited on page 3)

Moritz Hoffmann, Martin Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke E Husic, Stefan Klus, Hao Wu, Nathan Kutz, Steven L Brunton, et al. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 3(1):015009, 2021. (Cited on pages 4 and 16)

Brooke E Husic and Vijay S Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, 2018. (Cited on page 4)

Wei Jiang and Benoît Roux. Free energy perturbation hamiltonian replica-exchange molecular dynamics (fep/h-remd) for absolute ligand binding free energy calculations. *Journal of chemical theory and computation*, 6(9):2559–2565, 2010. (Cited on page 1)

Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi S Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023. (Cited on page 9)

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 22277–22303, 2024a. (Cited on pages 6, 7, 9, and 17)

Bowen Jing, Hannes Stark, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*, 2024b. (Cited on pages 1, 2, 3, 5, 6, 15, 16, 18, and 19)

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021. (Cited on pages 1, 5, and 15)

Leon Klein and Frank Noe. Transferable boltzmann generators. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. (Cited on pages 1 and 9)

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36: 52863–52883, 2023. (Cited on pages 1, 3, 6, and 19)

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021. (Cited on page 9)

Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024. (Cited on page 1)

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002. (Cited on page 1)

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. *bioRxiv*, pp. 2024–12, 2024. (Cited on pages 1, 2, 6, 7, 9, 16, 17, and 18)

Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. (Cited on page 2)

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. (Cited on pages 3 and 5)

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 2)

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. (Cited on page 5)

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. (Cited on page 18)

Jiarui Lu, Xiaoyin Chen, Stephen Zhewen Lu, Chence Shi, Hongyu Guo, Yoshua Bengio, and Jian Tang. Structure language models for protein conformation generation. *arXiv preprint arXiv:2410.18403*, 2024. (Cited on page 9)

Advaith Maddipatla, Nadav Bojan Sellam, Meital Bojan, Sanketh Vedula, Paul Schanda, Ailie Marx, and Alex M Bronstein. Inverse problems with experiment-guided alphafold. *arXiv preprint arXiv:2502.09372*, 2025. (Cited on page 9)

Maciej Majewski, Adrià Pérez, Philipp Thölke, Stefan Doerr, Nicholas E Charron, Toni Giorgino, Brooke E Husic, Cecilia Clementi, Frank Noé, and Gianni De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature communications*, 14(1):5739, 2023. (Cited on page 2)

J Andrew McCammon, Bruce R Gelin, and Martin Karplus. Dynamics of folded proteins. *nature*, 267(5612):585–590, 1977. (Cited on page 1)

Robert T. McGibbon, Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528 – 1532, 2015. doi: 10.1016/j.bpj.2015.08.015. (Cited on page 17)

Laurence Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Se (3) equivariant augmented coupling flows. *Advances in Neural Information Processing Systems*, 36:79200–79225, 2023. (Cited on page 9)

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. (Cited on pages 2, 6, 7, 16, and 21)

Juno Nam, Sulin Liu, Gavin Winter, and Rafael Gomez-Bombarelli. Generative acceleration of molecular dynamics simulations for solid-state electrolytes. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*, 2024. (Cited on pages 1 and 3)

Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106(45):19011–19016, 2009. (Cited on pages 2 and 4)

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. (Cited on pages 1 and 9)

OpenMM Team. Openmm benchmarks. https://openmm.org/benchmarks, 2025. Accessed: 2025-05-08. (Cited on page 9)

Vijay S Pande, Kyle Beauchamp, and Gregory R Bowman. Everything you wanted to know about markov state models but were afraid to ask. *Methods*, 52(1):99–105, 2010. (Cited on page 4)

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023. (Cited on pages 5 and 15)

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1), 2013. (Cited on pages 5 and 6)

Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of chemical physics*, 134(17), 2011. (Cited on pages 2 and 4)

Aneesur Rahman. Correlations in the motion of atoms in liquid argon. *Physical review*, 136(2A): A405, 1964. (Cited on page 1)

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015. (Cited on page 3)

Hannes Risken and Hannes Risken. *Fokker-planck equation*. Springer, 1996. (Cited on page 1)

Andrea Rizzi, Paolo Carloni, and Michele Parrinello. Targeted free energy perturbation revisited: Accurate free energies from mapped reference potentials. *The journal of physical chemistry letters*, 12(39):9449–9454, 2021. (Cited on page 9)

Andrea Rizzi, Paolo Carloni, and Michele Parrinello. Multimap targeted free energy estimation. *arXiv preprint arXiv:2302.07683*, 2023. (Cited on page 9)

Susanna Röblitz and Marcus Weber. Fuzzy spectral clustering by pcca+: application to markov state models and data classification. *Advances in Data Analysis and Classification*, 7:147–179, 2013. (Cited on page 6)

Davide Sabbadin and Stefano Moro. Supervised molecular dynamics (sumd) as a helpful tool to depict gpcr–ligand recognition pathway in a nanosecond time scale. *Journal of chemical information and modeling*, 54(2):372–376, 2014. (Cited on page 1)

Martin K Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé. Pyemma 2: A software package for estimation, validation, and analysis of markov models. *Journal of chemical theory and computation*, 11(11):5525–5542, 2015. (Cited on page 4)

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: multiple time-resolution surrogates for molecular dynamics. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 36449–36462, 2023. (Cited on pages 1, 3, 6, and 19)

Christof Schütte, Stefan Klus, and Carsten Hartmann. Overcoming the timescale barrier in molecular dynamics: Transfer operators, variational principles and machine learning. *Acta Numerica*, 32: 517–673, 2023. (Cited on page 3)

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2021. (Cited on page 3)

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017. (Cited on pages 2, 7, and 16)

Charlie B Tan, Avishek Joey Bose, Chen Lin, Leon Klein, Michael M Bronstein, and Alexander Tong. Scalable equilibrium sampling with sequential boltzmann generators. *arXiv preprint arXiv:2502.18462*, 2025. (Cited on page 9)

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. (Cited on page 3)

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. (Cited on page 2)

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2. (Cited on page 17)

David J Wales. Energy landscapes and properties of biomolecules. *Physical biology*, 2(4):S86, 2005. (Cited on page 3)

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. (Cited on page 9)

Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racanière, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14), 2020. (Cited on page 9)

Hao Wu, Feliks Nüske, Fabian Paul, Stefan Klus, Péter Koltai, and Frank Noé. Variational koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of chemical physics*, 146(15), 2017. (Cited on page 5)

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. (Cited on pages 1 and 9)

Outline of Appendix

In Section A we describe the MARS-FM architecture, and our MSM-informed procedure for drawing training pairs; Section B reports dataset details, supplementary MSM construction choices, evaluation metrics, hyper-parameters, sampling settings, and compute resources; Section C gathers supplementary figures and tables—including tetrapeptide TICA plots, MD-Cath samples plots, expanded MD-Cath flexibility metrics, and the full 320K evaluation—that complement the main text; Section Section D summarizes usage of LLMs; finally, Section E reflects on the societal benefits and potential misuse risks of generative models such as MARS-FM and Section F contains reproducibility statement.

A Implementation Details

A.1 Architecture

Inspired by MDGEN (Jing et al., 2024b), the velocity network $v_\theta : (\mathbb{R}^{21})^L \to (\mathbb{R}^{21})^L$, with L being the overall number of residues, employs modified DiT blocks (Jing et al., 2024b; Peebles & Xie, 2023) and Invariant Point Attention (IPA) layers (Jumper et al., 2021). A pseudocode is given in Algorithm 1.

Algorithm 1 Velocity Network

Require: conditioning tokens $\chi^{(0)} \in \mathbb{R}^{21L}$, target tokens $\chi_s^{(1)} \in \mathbb{R}^{21L}$, conditioning roto-translations $g^{(0)} \in SE(3)^L$, amino acid identities a, flow matching time $s \sim \mathcal{U}(0,1)$
Ensure: velocity $v \in \mathbb{R}^{21L}$

$s \leftarrow \text{TIMEEMBED}(s)$
$x \leftarrow \text{AMINOACIDEMBED}(a)$
for $\ell = 1$ **to** n_{IPA} **do**
  $x \leftarrow \text{INVARIANTPOINTATTENTIONLAYER}\left(x, g^{(0)}, s\right)$
$x \leftarrow x + \text{LINEAR}(\chi^{(0)}) + \text{LINEAR}(\chi_s^{(1)})$
for $\ell = 1$ **to** n_{DTA} **do**
  $x \leftarrow \text{DIFFUSIONTRANSFORMERATTENTIONLAYER}(x, s)$
return $\text{DIFFUSIONTRANSFORMERFINALLAYER}(x, s)$

At flow-matching time $s \in [0,1]$, we construct the noisy representation as a linear combination of the ground-truth frame and i.i.d. Gaussian noise ε:

$$\chi_s^{(1)} \;=\; \sigma(s)\,\varepsilon + \alpha(s)\,\chi^{(1)}, \qquad \alpha(s) = \sin\!\left(\tfrac{\pi}{2}s\right), \quad \sigma(s) = \cos\!\left(\tfrac{\pi}{2}s\right).$$

The derivatives $\dot\alpha(s) = \frac{\pi}{2}\cos\left(\frac{\pi}{2}s\right)$ and $\dot\sigma(s) = -\frac{\pi}{2}\sin\left(\frac{\pi}{2}s\right)$ enter the target velocity $\dot\chi_s^{(1)}$ during training.

A.2 Drawing trainings pairs

Drawing Procedure. Given a per-protein Markov State Model (MSM), for each training example we draw a pair (x_0, x_1), as follows,

1. **Select source states.** We always include the state containing the first frame of the trajectory. The remaining source states are sampled uniformly (with replacement) from all Markov states.
2. **Draw destination states.** For each source state S_i, we sample a fixed number of destination states S_j independently using the MSM transition probabilities T_{ij} as a categorical distribution.
3. **Select frames.** For each source–destination pair (S_i, S_j), we uniformly sample one frame $x(t) \in S_i$ as the conditioning frame and one frame $x_1 \in S_j$ as the target frame. These form the training pair (x_0, x_1). This step can be repeated a few times

This strategy balances sampling between rare and frequent metastable states while preserving the transition structure encoded in the MSM. Our sampling method allows to sample across different replicas.

B Experimental Details

B.1 Datasets

Tetrapeptides We use the tetrapeptides dataset introduced in Jing et al. (2024b), where trajectories are trained at a resolution of 10 ps over a total duration of 100 ns. We adopt the exact same split as introduced by the authors, comprising 3,109 tetrapeptides for training, 100 for validation, and 100 for testing.

MD-Cath We use the MD-Cath dataset introduced in Mirarchi et al. (2024), which provides molecular dynamics simulations for 5,398 protein domains. Frames are trained at a resolution of 1 ns, with an average trajectory length of 464 ns per replica and the majority of trajectories being 500 ns long. Each domain (ranging from 50 to 500 residues, average 137) is simulated at five temperatures from 320 K to 450 K, with five replicas per temperature, yielding 25 trajectories per domain. For the train/validation/test split, we follow the similar strategy to Lewis et al. (2024) (but more strict) and use MMseqs2 (Steinegger & Söding, 2017) with default parameters and maximum sensitivity to exclude from the test set any domain that shares more than 20% sequence identity with any training or validation domain. This results in a final split of 4,304 domains for training, 538 for validation, and 495 for testing. We primarily evaluate our models on the highest-temperature subset (450 K) of MD-Cath, where most proteins undergo (partial) unfolding within 100–500 ns. In contrast, lower temperature simulations (e.g., at 320 K) often remain near the native structure and exhibit limited conformational diversity—for a quantitative comparison see Appendix C. We use all available replicas for the given temperature.

B.2 Additional details on Markov State Model construction

We follow the MSM construction as described in Section 5. All MSMs are implemented using the `deeptime` library (Hoffmann et al., 2021). The resulting transition probability matrices are symmetrized post hoc as $\mathbf{P} \leftarrow (\mathbf{P} + \mathbf{P}^\top)/2$.

For clustering of MD-Cath dataset we use standardized Radius of Gyration and Secondary Structures Fractions as defined in the Appendix B.3.

B.3 Evaluations and Metrics

B.3.1 Tetrapeptides

Evaluation metrics. We follow the evaluation protocol of Jing et al. (2024b) and compute the Jensen-Shannon Divergence (JSD) between model-generated and reference MD trajectories across several observables: (i) distributions torsional angles, (ii) TICAs, and (iii) equilibrium distributions derived from Markov State Models.

Macrostate Mean Absolute Error (mMAE). To complement the evaluation and JSD-based observables, we additionally report the mMAE. The mMAE measures the discrepancy in metastable-state populations between the generated and reference ensembles, after both are coarse-grained into macrostates using the MSM calculated as described in Section 5. It is computed as the mean absolute difference in free energy across macrostates,

$$\text{mMAE} = \frac{1}{n}\sum_{i=1}^{n} \left| G_i^{\text{model}} - G_i^{\text{ref}} \right|, \quad \text{where} \quad G_i = -k_B T \log \pi_i.$$

Here, G_i denotes the free energy of macrostate i, k_B is the Boltzmann constant, T is the temperature, and π_i is the stationary distribution (normalized histogram count) of macrostate i. We use a small floor value (e.g., 10^{-4}) to avoid numerical instability in the logarithm. Since we do not have access to additional replicas, we did not report the mMAE of the Oracle.

B.3.2 MD-CATH

Predicting Flexibility. We follow the exact procedure of Jing et al. (2024a) to compute pairwise Root Mean Square Deviation (RMSD), global Root Mean Square Fluctuation (RMSF), and per-target RMSF. These metrics capture the flexibility and structural variability of conformational ensembles.

Distribution Alignment Metrics. To assess the similarity between model-generated and reference ensembles, we compute both the forward Kullback–Leibler (KL) divergence and the Jensen–Shannon Divergence (JSD). Both distributions are estimated by binning values into histograms with 100 bins and normalizing the counts to obtain discrete probability distributions. To avoid numerical issues, we apply a small floor value ($\epsilon = 10^{-5}$) to the model probabilities before computing KL. Specifically:

- The forward KL divergence, $D_{\mathrm{KL}}(P \,\|\, Q)$, is computed with smoothed model probabilities: $Q = \max(Q, \epsilon)$.
- The Jensen–Shannon Divergence is computed as the squared JSD distance — `scipy`'s `jensenshannon` function (Virtanen et al., 2020).

Radius of Gyration. We compute the radius of gyration for each frame in a generated trajectory and compare its distribution against that of generated trajectories across all replicas at a given temperature. It is defined as,

$$R_g = \sqrt{\frac{\sum_{i=1}^{N} m_i \left\|\mathbf{x}_i - \mathbf{x}_{\text{center}}\right\|^2}{\sum_{i=1}^{N} m_i}}, \tag{6}$$

where m_i and $\mathbf{x}_i$ denote the mass and position of atom i, and $\mathbf{x}_{\text{center}}$ is the center of mass of the structure. Higher R_g values typically correspond to more extended or unfolded conformations.

Secondary Structure Fractions. We compute the fraction of residues in secondary structure per frame based on DSSP assignments. Specifically, we include canonical α-helix and β-strand states, using `mdtraj`'s `compute_dssp` function (McGibbon et al., 2015),

$$f_{\text{SS}} = \frac{1}{L} \sum_{j=1}^{L} \mathbb{I}\left[s_j \in \{\text{H, G, I, E, B}\}\right], \tag{7}$$

where L is the number of residues in a frame, s_j is the DSSP-assigned secondary structure of residue j, and $\mathbb{I}$ is the indicator function.

Markov State Model Recovery. We build MSMs using the same procedure as for training (Section 5), and then we compare the resulting stationary distributions via Jensen–Shannon Divergence.

Folding Free Energies. To estimate folding free energies, we follow the BioEmu protocol (Lewis et al., 2024). For each protein, we first compute a native–contact (FNC) score using all heavy–atom pairs with sequence separation $|i - j| > 3$ and reference distance $d_{ij}^{\text{ref}} < 10$ Å. For a given trajectory frame at time t, each pair contributes

$$q_{ij}(t) = \left[1 + \exp\left(-\beta\left[d_{ij}(t) - \lambda d_{ij}^{\text{ref}}\right]\right)\right]^{-1}, \tag{8}$$

with $\beta = 5$ and $\lambda = 1.2$. The overall contact score is then defined as $Q(t) = \langle q_{ij}(t) \rangle$, averaged over all native contacts. To separate folded and unfolded ensembles, we determine a midpoint threshold $Q_{1/2}$ from the kernel density estimate of the 320 K MD reference distribution. Specifically, we locate the deepest minimum in the range 0.45–0.90; if no minimum is found, we set $Q_{1/2} = 0.70$. This same value is then reused to evaluate both 450 K MD and all generative trajectories.

Given this threshold, we compute the folded probability of each frame as

$$p_{\text{fold}}(t) = \left[1 + \exp\left(-2s\left[Q(t) - Q_{1/2}\right]\right)\right]^{-1}, \qquad s = 10, \tag{9}$$

and report its ensemble average $\bar{p}_{\text{fold}}$. Finally, the folding free energy at temperature T is defined as

$$\Delta G = -k_{\text{B}} T \, \ln\left[\frac{\bar{p}_{\text{fold}}}{1 - \bar{p}_{\text{fold}}}\right]. \tag{10}$$

MD Oracle performance To quantify MD Oracle performance for 100 and 500 samples, we hold out one replica and compare it against a reference set containing the other four replicas. For 1000 samples, we hold out two replicas and use the remaining three as the reference.

B.4 Training Details

We follow the hyper-parameters of Jing et al. (2024b) as closely as possible. Throughout all experiments we use an exponential moving average with decay 0.999, and AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of 1×10^{-4}. We use an architecture described in Appendix A.1 with 5 transformer layers, 384-dimensional token embeddings, 16-head multi-head attention, and an IPA layers with 4 heads, 32-dimensional head size, and 8 query–key as well as 8 value points.

For the tetrapeptide dataset we recreate the setup of Jing et al. (2024b), sampling MDGen-1000 from the released checkpoint and training our MarS-FM model for 1000 epochs and MDGen-200 for 2500 epochs, both with a batch size of 8.

On MD-Cath dataset, MarS-FM is run for 1000 epochs with batch size 8 while sampling training pairs from two source clusters (x_0) and two corresponding destination clusters (x_1) with 12 sequences per cluster; MDGen-100 is trained for the same epoch budget with batch size 1, and MDGen-20 for 2500 epochs with batch size 4.

B.5 Sampling Details

All trajectories are sampled from the first MD frame in the test set; for the MD-Cath data, we use the first frame of the first replica. MDGen is then applied autoregressively, where the last of generated frames at each rollout serves as the conditional input for the next roullout. For MDGen in parallel mode, all calls are conditioned on the initial MD frame; we simply call it multiple times independently.

For MarS-FM $\oplus$ MDGen, we first draw a set of $\frac{\text{total number of frames}}{\text{MDGen window size}}$ frames using MarS-FM in parallel. Each output of MarS-FM is then used as the conditional frame for an independent MDGen rollout.

For hierarchical MarS-FM sampling, we begin from the initial frame and sample 200 first-layer children in parallel. Each of these nodes is then expanded once, and the process continues recursively until the required number of frames is reached.

B.6 Compute Resources

All preprocessing tasks (including TICA projection, Radius of Gyration calculation, Secondary Structures Fractions calculation, clustering), and evaluation metric computation (such as divergence measures, RMSD, and RMSF), are performed on CPU nodes of the compute cluster.

Training and sampling are conducted on a single NVIDIA H100 GPU. Training on the tetrapeptide dataset takes approximately 1–2 days, while training on the MD-Cath dataset requires around 4–6 days. For sampling, generating 10,000 conformations for the full tetrapeptides test set takes approximately 1.5 hours; for MD-CATH, generating 500 conformations for all test domains takes around 5 hours.

B.7 Baseline Caveats

BioEmu For larger systems, we also benchmark against BioEmu (Lewis et al., 2024). Several caveats should be noted when interpreting these results. First, BioEmu uses a backbone-only protein representation, whereas MarS-FM explicitly models both backbone and side-chain torsion angles. Second, BioEmu was trained on substantially more data, including proprietary MD simulations, which makes direct comparisons less controlled. Third, BioEmu currently provides only an inference pipeline, with no publicly available training code. Despite these limitations, we performed inference evaluations on MD-Cath using the pre-trained BioEmu model, generating 500 conformations for comparison with reference MD simulations.

Other Methods Several other works have aimed, similar to MDGen (Jing et al., 2024b), at speeding up molecular dynamics, but they exhibit important limitations. The Implicit Transfer Operator (ITO) (Schreiner et al., 2023) relies on a coarse-grained Cα representation and assumes a fully connected graph with $\mathcal{O}(M^2)$ scaling, making application to systems with thousands of atoms infeasible. TimeWarp (Klein et al., 2023), based on RealNVP and a Transformer stack, incurs prohibitive training costs even for short peptides and has only been demonstrated on tetrapeptides. EquiJump (Costa et al., 2024) does not provide publicly available code and has so far only reported performance on the same fast-folding proteins used for training, without evidence of generalization.

C Additional Results

C.1 Supplementary Tetrapeptides TICA plots

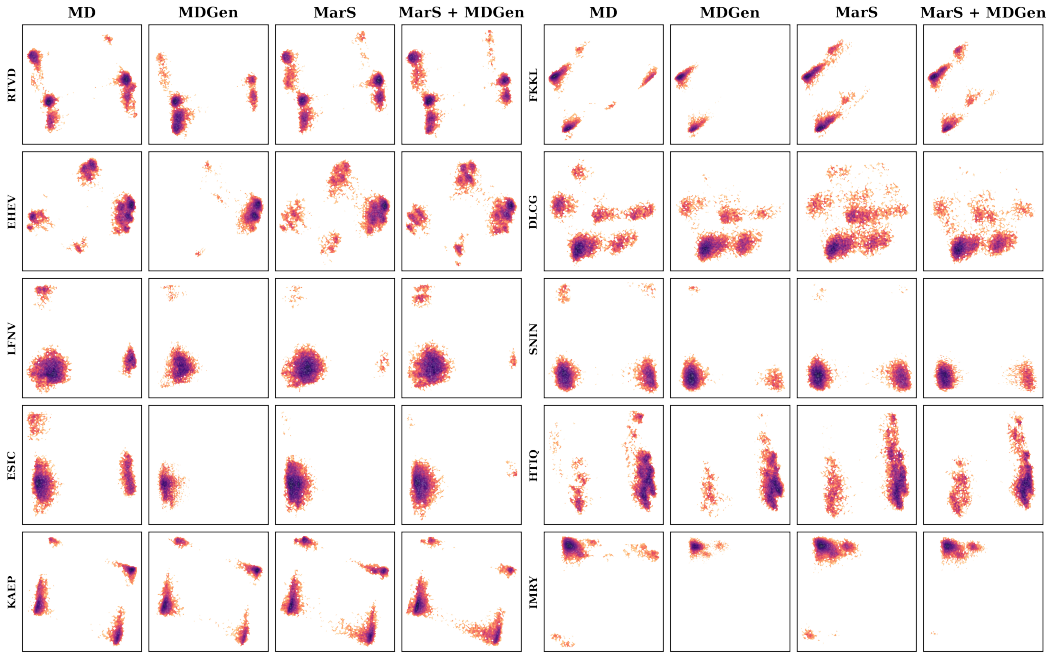

Figure 6: TICA plot for 10 random peptides in the test set, comparing MD ground-truth, MDGen, MarS (ours) and MarS + MDGen (ours). Our frameworks explore modes that are otherwise entirely ignored by MDGen.

C.2 SUPPLEMENTARY PLOTS OF GENERATED SAMPLES FROM MD-CATH DOMAINS

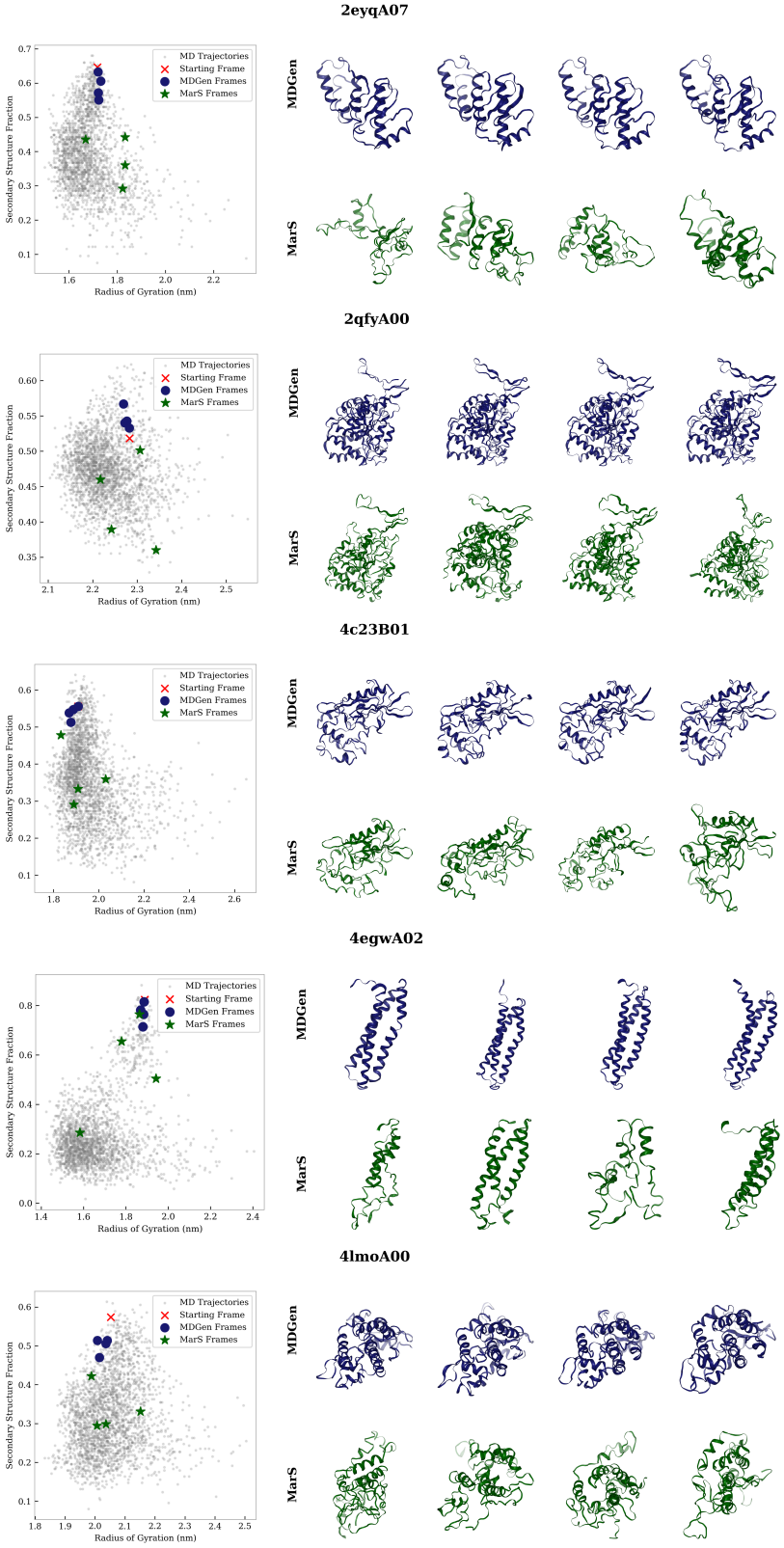

Figure 7: First 4 samples generated by MDGen and MARS-FM for 5 random domains in the MD-Cath test set.

C.3 JSD FOR PAIRWISE RMSD AND GLOBAL RMSF, AND PEARSON r FOR FOLDING FREE ENERGIES

To further quantify structural flexibility and thermodynamic accuracy, we report the mean JSD on the test set between reference MD samples and 500 samples generated by our models and baselines, along with the Pearson correlation (r) for folding free energies (ΔG_{fold}). The results are presented in Table 5. For each protein, trajectories are aligned with respect to the first MD frame of the first replica before calculating the metrics. Both hybrid sampling (MARS-FM $\oplus$ MDGen) and hierarchical sampling (MARS-FM) more effectively capture the structural distribution of the trajectories and recover folding free energy correlations than the extended MDGEN baselines, often by a large margin.

Table 5: JSD $\downarrow$ for pairwise RMSD and global RMSF, and Pearson r $\uparrow$ for Folding Free Energies (ΔG_{fold}). Results based on sampled trajectories of 500 conformations compared to ground-truth distributions and averaged over 5 inference runs.

	Pairwise RMSD JSD	Global RMSF JSD	ΔG_{fold} r
MD (Oracle)	0.16 ± .005	0.09 ± .003	0.90 ± .002
MDGen-100	0.22 ± .008	0.31 ± .009	0.82 ± .006
MDGen-20	0.38 ± .005	0.52 ± .010	0.85 ± .005
MDGen-100 (in parallel)	0.54 ± .004	0.58 ± .004	0.83 ± .004
MDGen-20 (in parallel)	0.73 ± .002	0.71 ± .001	0.86 ± .001
BioEmu	0.70 ± .001	0.72 ± .001	0.71 ± .002
MARS-FM $\oplus$ MDGen-20	**0.15** ± .001	**0.24** ± .001	**0.88** ± .001
MARS-FM	<u>0.18</u> ± .001	<u>0.28</u> ± .001	<u>0.87</u> ± .003

C.4 MD-CATH AT 320K

In contrast to the high-temperature replicas, trajectories at 320 K display no large-scale domain motions. Both the radius of gyration and secondary structure fractions exhibit nearly an order-of-magnitude lower variability (Table 6). Similarly, all-atom RMSF and secondary structure fluctuations remain tightly clustered around their native values. These statistics confirm that proteins largely remain stable at lower temperatures, with no significant conformational changes—consistent with the findings reported by Mirarchi et al. (2024).

Table 6: Structural variability at 320K versus 450K in the MD-CATH dataset: standard deviations of radius of gyration and secondary-structure fractions, plus mean pairwise RMSD and mean all-atom RMSF

	MD-Cath 320K	**MD-Cath 450K**
Radius of Gyration std (nm)	0.045 ± .058	0.265 ± .164
Secondary Structures Fractions std	0.046 ± .024	0.104 ± .040
Pairwise RMSD (Å)	4.44 ± 2.93	14.54 ± 4.71
All Atom RMSF (Å)	2.81 ± 2.10	10.34 ± 3.49

For methodological consistency, we built the Markov State Model (MSM) with the same hyper-parameters at 320 K as for the 450 K regime (10 clusters and 50 ns lag time). However, MSM could be temperature-specific, for instance, by shortening lag times or changing the number of clusters to capture subtler motions. As shown in Tables 7 and Table 8, MARS-FM continues to outperform all baselines in reproducing structural flexibility, radius of gyration and MSM state reconstruction. All methods show only marginal differences on secondary-structure KL/JSD which could be potentially improved by hyper-tuning MSM settings.

D LLM USAGE

We used large language models (GPT-5) to assist in improving the grammar of the manuscript, as well as facilitate writing code.

Table 7: Pearson $r \uparrow$ for RMSD and RMSF, forward KL divergence $\downarrow$ and JSD $\downarrow$ of gyration radius, fraction of secondary structures, MSMs distribution, and folding free energy MAE (kcal/mol) $\downarrow$ **at 320K**. Results based on sampled trajectories of 500 conformations compared to ground-truth distributions and averaged over 5 inference runs.

	Pairwise RMSD r	Global RMSF r	Per target RMSF r	Gyration Radius KL	Gyration Radius JSD	Secondary Structures KL	Secondary Structures JSD	MSM JSD	ΔG_{fold} MAE
MD (Oracle)	0.90 ± .009	0.88 ± .005	0.91 ± .003	0.82 ± .028	0.10 ± .003	0.37 ± .017	0.05 ± .002	0.13 ± .007	0.32 ± .003
MDGen-100	0.82 ± .011	0.78 ± .004	0.81 ± .007	1.50 ± .022	0.20 ± .003	0.76 ± .014	0.13 ± .001	0.18 ± .002	0.77 ± .009
MDGen-20	0.79 ± .018	0.75 ± .010	0.79 ± .012	1.30 ± .065	0.19 ± .007	0.78 ± .009	0.15 ± .001	0.19 ± .002	1.10 ± .013
MDGen-100 (in parallel)	0.87 ± .004	0.83 ± .002	0.87 ± .003	1.60 ± .027	0.18 ± .002	**0.44** ± .004	**0.07** ± .001	0.14 ± .002	0.64 ± .012
MDGen-20 (in parallel)	0.83 ± .004	0.81 ± .002	0.85 ± .003	2.10 ± .018	0.21 ± .001	0.71 ± .005	0.09 ± .001	0.18 ± .002	1.10 ± .004
BioEmu	0.58 ± 001	0.63 ± 004	0.84 ± 002	2.67 ± 042	0.36 ± 001	0.94 ± 011	0.15 ± 001	0.25 ± 003	0.83 ± .003
MARS-FM ⊕ MDGen-20	**0.91** ± .001	**0.87** ± .001	**0.89** ± .001	0.74 ± .001	**0.13** ± .001	0.46 ± .001	0.09 ± .001	**0.10** ± .001	0.62 ± .001
MARS-FM	0.90 ± .001	**0.87** ± .001	**0.90** ± .003	**0.72** ± .004	0.14 ± .001	0.68 ± .009	0.12 ± .001	0.14 ± .001	**0.58** ± .001

Table 8: Pearson $r \uparrow$ for RMSD and RMSF, forward KL divergence $\downarrow$ and JSD $\downarrow$ of gyration radius, fraction of secondary structures, MSMs distribution, and folding free energy MAE (kcal/mol) $\downarrow$ **at 320K**. Results based on sampled trajectories of 100 / 1000 conformations compared to ground-truth distributions and averaged over 5 inference runs.

	Pairwise RMSD r	Global RMSF r	Per target RMSF r	Gyration Radius KL	Gyration Radius JSD	Secondary Structures KL	Secondary Structures JSD	MSM JSD	ΔG_{fold} MAE
MD (Oracle)	0.87 / 0.93	0.85 / 0.91	0.88 / 0.93	2.21 / 0.44	0.18 / 0.07	0.96 / 0.19	0.09 / 0.04	0.43 / 0.08	0.58 / 0.32
MDGen-100	0.82 / 0.75	0.77 / 0.73	0.81 / 0.77	3.43 / 1.15	0.29 / 0.20	1.23 / 1.00	0.12 / 0.20	0.20 / 0.22	0.70 / 1.15
MDGen-20	0.82 / 0.74	0.79 / 0.68	0.83 / 0.73	2.96 / 1.25	0.25 / 0.22	1.08 / 1.05	**0.11** / 0.22	0.18 / 0.24	0.62 / 1.08
MDGen-100 (in parallel)	0.81 / 0.88	0.77 / 0.85	0.81 / 0.88	3.52 / 1.24	0.30 / 0.16	1.19 / **0.33**	**0.11** / **0.06**	0.21 / 0.13	0.71 / 0.63
MDGen-20 (in parallel)	0.81 / 0.83	0.78 / 0.81	0.83 / 0.86	3.26 / 1.86	0.27 / 0.20	1.28 / 0.62	0.12 / 0.09	0.20 / 0.18	1.10 / 1.09
BioEmu	0.58 / 0.58	0.62 / 0.63	0.83 / 0.84	4.07 / 2.39	0.40 / 0.35	1.57 / 0.86	0.18 / 0.14	0.42 / 0.24	0.84 / 0.82
MARS-FM ⊕ MDGen-20	**0.89** / **0.90**	**0.86** / **0.87**	0.87 / **0.90**	2.23 / **0.57**	0.22 / **0.12**	**0.89** / 0.40	0.12 / 0.09	**0.14** / **0.11**	0.60 / 0.63
MARS-FM	**0.89** / **0.90**	**0.86** / **0.87**	**0.89** / **0.90**	**1.99** / 0.61	**0.21** / 0.13	1.25 / 0.60	0.15 / 0.12	0.15 / 0.13	**0.59** / **0.58**

E BROADER IMPACTS

The MARS-FM model's ability to generate realistic protein conformations over two orders of magnitude faster than conventional MD can accelerate drug discovery and other socially beneficial molecular design tasks, while also reducing the energy footprint of large-scale simulations. Conversely, the same speed-ups and accessibility could facilitate malicious protein engineering or foster overconfidence when the model is applied outside its intended scope.

F REPRODUCIBILITY STATEMENT

All details necessary to reproduce the experiments are present in the paper. All code necessary to reproduce our results has been included in the supplementary material as part of the submission and will be released publicly upon acceptance.