LEARNING TOPOLOGICAL REPRESENTATIONS OF PROTEIN STRUCTURE AND DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern protein representation models support tasks such as enzyme design and drug discovery, but their reliance on static data such as sequence and native structure limits their ability to capture the conformational dynamics that drive protein function. We investigate whether persistent homology (PH) can provide descriptors shared across diverse proteins that retain global structure, fine-grained conformational variability, and kinetically relevant information without large-scale pretraining. We introduce the *masked Flood complex*, i.e., an adaptation of a recently proposed simplicial complex construction, that incorporates domain knowledge to emphasize inter-residue structure at low computational cost. We then use it to compute PH on molecular dynamics (MD) sampled structures, vectorize the persistence diagrams into a shared coordinate system, and probe the capacity of these representations in terms of the aforementioned aspects. To assess the amount of kinetic information, we learn low-dimensional embeddings from time-lagged observations and evaluate Markov state models (MSMs) estimated from them. Using these MSMs to guide training of the recent `MarS-FM` generative framework improves several ensemble statistics relative to the original model. After finetuning on lower-temperature MD data and adapting the sampling procedure, the resulting model also shows promising transfer to fast folding proteins.

1 INTRODUCTION

Machine learning has driven major advances in protein structure prediction (Jumper et al., 2021; Aranganathan et al., 2025). Complementing these advances, representations learned from sequence (Lin et al., 2023), structure (Hsu et al., 2022), and their integration with functional annotations (Hayes et al., 2025) support the study of protein function, with applications to screening candidate drugs, antibodies, and enzymes (Nguyen & Hy, 2025; Heinzinger & Rost, 2025).

While protein structure is a primary determinant of function, it is inherently *dynamic*: even the native state of a protein comprises an ensemble of rapidly fluctuating conformations with comparable stability (Voet et al., 2016; McCammon et al., 1977), while intrinsically disordered proteins explore heterogeneous ensembles (Trivedi & Nagarajaram, 2022). These dynamics underlie processes such as protein folding (Jäger et al., 2006), transient cryptic-pocket formation (Zhang & Bowman, 2026), and ligand binding (Ostermann et al., 2000).

A standard computational approach to study these processes is *molecular dynamics (MD)*, which simulates molecular trajectories by numerically integrating equations of motion. The resulting trajectories record both the variation within physical states and the transitions between them. Extracting the corresponding kinetic information requires *molecular descriptors* that resolve relevant structural and dynamical differences. Such descriptors enable analyses such as Markov state models (MSMs) (Zwanzig, 1983; Singhal et al., 2004), which describe kinetics via transitions between discrete states.

Notably, conventional descriptors such as backbone torsions, pairwise distances, or contact maps are tied to a single system as their coordinates refer to specific residues or atom pairs and therefore do not carry the same meaning across different proteins. However, recent MD datasets such as `mdCATH` (Mirarchi et al., 2024), `ATLAS` (Vander Meersche et al., 2024) or `GPCRmd` (Rodríguez-Espigares et al., 2020) span simulations of thousands of proteins and thereby invite learning a *shared* representation space. Such shared embeddings are available from protein language models (PLMs) trained as foundation models on a large and diverse corpus of data. Many popular PLMs (e.g., Elnaggar et al.,

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

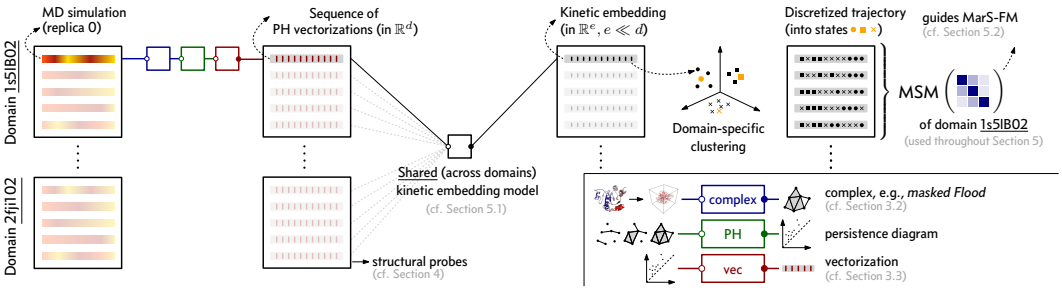


Figure 1: For each frame of an MD trajectory (here, one replica of the mdCATH domain 1s5lB02), we compute PH, e.g., via our masked Flood complex (Section 3.2), and vectorize the persistence diagrams (Section 3.3), turning the trajectory into a sequence of PH vectorizations. A kinetic embedding model shared across domains maps these to a low-dimensional space, in which domain-specific clustering yields discretized trajectories and an MSM per domain (Section 5). We probe the PH vectorizations for structural information (Section 4) and evaluate the MSMs directly (Section 5.1) and as guidance for generative modeling (Section 5.2). The legend (bottom right) details the per-frame steps: the complex construction yields a nested family of complexes over increasing radius r, i.e., a filtration, and PH summarizes how their topology changes along it in a persistence diagram, which is then vectorized.

2022; Lin et al., 2023) operate directly on the level of amino acid sequences and therefore cannot distinguish between different conformations of the same protein. Structure-aware extensions (e.g., Hsu et al., 2022; Su et al., 2024) can discern them, but are typically trained on a single near-native state, rather than on the many conformations a protein samples in MD simulations. Closer to our setting, Aboelsoud et al. (2026) add MD trajectories as a modality to the protein foundation model OneProt (Flöge et al., 2025), but pool each trajectory into a single embedding, summarizing a protein's variability rather than representing individual conformations. Still, all these approaches rely on learning representations via large-scale pretraining, and their massive compute requirements motivate the question: *How well can principled, conformation-specific descriptors represent protein structure and dynamics without large-scale pretraining?*

To obtain descriptors suitable for learning across protein domains, we build on *persistent homology (PH)* (Edelsbrunner et al., 2002; Zomorodian & Carlsson, 2005), i.e., a tool from topological data analysis that summarizes shape across spatial scales. PH has proven informative for analyzing, classifying, and predicting properties of proteins (Xia & Wei, 2014; Cang et al., 2015; Cang & Wei, 2017b), and for distinguishing conformations along folding trajectories (Ichinomiya et al., 2020; Ichinomiya, 2022). Upon vectorization, these topological summaries provide a common representation space without requiring feature correspondence between proteins.

Motivated by these developments as well as prior work on incorporating biological information into PH-based analyses (Cang et al., 2018; Dey & Mandal, 2018), and the availability of large-scale MD datasets, we seek to extend this computational pipeline to large collections of proteins and their trajectories by computing descriptors that are compatible across domains and informed by biological knowledge that emphasizes relevant aspects of molecular geometry. Effectively, we ask: *How well can biologically informed PH descriptors represent protein structure and dynamics while supporting learning across proteins and generalization to unseen domains?*

Contribution To investigate the aforementioned questions and to address the challenge of designing topologically grounded representations of protein structure and dynamics, we first introduce the *masked Flood complex* as an adaptation of a recent method for efficient PH computation tailored to the characteristics of proteins. We then probe representation quality of the topological summaries by empirically showing that they carry important information about global shape and capture instantaneous and local variations. Further, we assess the quality of topologically informed MSMs to probe representations for their kinetic content. Finally, when used within the recent MarS-FM model of Kapuśniak et al. (2026), where these MSMs guide the training of a generative model of molecular structures, we show (1) consistently strong performance across a range of ensemble statistics on the mdCATH dataset and (2) promising transfer to the regime of fast folding proteins (Lindorff-Larsen et al., 2011) which fundamentally differs from the mdCATH training data.

2 Related Work

Computational topology for molecular structure and dynamics Prior work has used PH-based descriptors to characterize protein structure and flexibility (Xia & Wei, 2014), classify proteins (Cang et al., 2015), and predict molecular properties (Cang & Wei, 2017a;b). More recent approaches combine *local* atomic PH with graph learning for interface-quality assessment (Han et al., 2025), or analyze persistence generators across large structural databases (Madsen et al., 2025). Biological information already enters these pipelines through element selection, interaction-dependent dissimilarities (Cang et al., 2018), and constructions informed by atomic radii and bonds (Dey & Mandal, 2018). In Definition 2, we will see that our approach builds on this principle through a different mechanism: residue-dependent masks restrict the atoms permitted to cover each simplex in the landmark-based simplicial complex construction of Graf et al. (2025).

In the context of MD simulations, PH has been used to study folding via representative cycles and residue-pair features, followed by dimensionality reduction (Ichinomiya et al., 2020; Ichinomiya, 2022). While these features aid structural interpretation, their coordinates depend on protein-specific residue indexing. Our topological summary vectorization enables learning across proteins of different sizes without residue correspondence, while kinetic coordinates are learned from time-lagged statistics. A complementary line of work analyzes conformational state-space topology: persistent clustering uses kinetic distances to identify metastable states (Chang et al., 2013), and Mapper-based methods characterize pathways (Yao et al., 2009; Donati et al., 2025). In contrast, we apply PH to individual conformations and use the resulting descriptors downstream, e.g., to learn a shared kinetic representation across domains, followed by domain-specific MSM estimation.

Learning from protein structure Pretrained encoders map sequences (Lin et al., 2023) or structures (Hsu et al., 2022; Su et al., 2024) to embeddings reused across tasks; ESM-IF1, for instance, is a transformer with GVP input layers (Jing et al., 2021), trained for inverse folding on millions of predicted structures. Equivariant networks such as EGNN (Satorras et al., 2021) and GotenNet (Aykent & Xia, 2025) instead usually learn from coordinates end-to-end for each task. Our descriptors are fixed before any learning, with biological priors entering through the simplicial complex, and are read out by simple probes.

Generative modeling of molecular ensembles While learning-based methods have a long history in MD (Blank et al., 1995), most prominently through learned interatomic potentials that amortize the cost of evaluating atomic interactions (Unke et al., 2021; Friederich et al., 2021), our work primarily connects to generative approaches for sampling molecular geometries (Noé et al., 2019; Lewis et al., 2025; Plainer et al., 2025; Costa et al., 2025; Bigi et al., 2025; Daigavane et al., 2025). Temporal information can enter through models of conformational transitions (Jing et al., 2024) or through MSM-guided selection of training pairs (Kapuśniak et al., 2026). Our downstream evaluation builds on the latter approach where we replace the MSMs used in `MarS-FM` with those learned from topological descriptors and assess the resulting generated ensembles (Section 5.2). This tests whether the kinetic information captured by our representations benefits generative-model training.

3 Method

The blueprint of our approach is to map atomic configurations to vectorized topological summaries from PH. Furthermore, we introduce a protein-structure–biased PH variant by constructing a simplicial complex that highlights relevant residue interactions. Details are in Sections 3.2 and 3.3.

3.1 Persistent homology primer

One way to extract topological features from a point cloud $X \subset \mathbb{R}^3$ is persistent homology. When computing PH, one represents the point cloud as a family of combinatorial objects $\{\Sigma_r : r \geq 0\}$ called *simplicial complexes*, each corresponding to a radius or scale $r > 0$ at which the point cloud is studied. PH then tracks the changes in the structure of Σ_r, and therefore in the underlying point cloud X as the scale varies, and summarizes them in persistence diagrams. More precisely, a simplicial complex Σ over the set X is a set of subsets $\sigma \subset X$, with the property that if $\sigma \in \Sigma$ then also $\tau \in \Sigma$ for all $\tau \subset \sigma$. Moreover, computing PH requires that the family $\{\Sigma_r\}_r$ of simplicial complexes satisfies $\Sigma_s \subset \Sigma_r$ whenever $s \leq r$, i.e., it needs to be a *filtration*.

Typically, for each simplex $\sigma \in \Sigma$ there is a minimal r such that $\sigma \in \Sigma_r$, i.e., the moment when it appears. This value r is called the filtration value of σ and a function $f : \Sigma \to \mathbb{R}$ that assigns to each simplex a filtration value is called a filtration function. Conversely, any function $f : \Sigma \to \mathbb{R}$ that satisfies $f(\tau) \leq f(\sigma)$ for every $\tau, \sigma \in \Sigma$ with $\tau \subset \sigma$ induces a well-defined filtration on Σ in terms of the sublevel sets $\Sigma_r = \{\sigma \in \Sigma \colon f(\sigma) \leq r\}$. Topological information about the filtration can be summarized in multisets of tuples $(b, d) \in \mathbb{R}^2$ called persistence diagrams. Each tuple corresponds to a topological feature; its entries specify at which filtration values r the feature appears (is born) and vanishes (dies), resp. For $X \subset \mathbb{R}^3$, there are typically three persistence diagrams of interest; H_0 corresponding to connected components, H_1 to loops, and H_2 to voids.

For commonly used filtrations, the appearance of a simplex in a complex Σ_r means that its vertices are close relative to r. For example, the Vietoris-Rips complex consists of all subsets of X with diameter at most r, and the Alpha complex consists of simplices in the Delaunay triangulation of X whose vertices lie on the boundary of a ball of radius at most r containing no other points of X.

For protein data with atoms represented as points in Euclidean space, however, not all points are equivalent: the residues' α-carbons (C_α) form a natural set of distinguished atoms. Moreover, atoms within the same residue remain in close spatial proximity, and their pairwise distances are typically less affected by non-trivial transitions such as protein folding or the opening of cryptic pockets. Incorporating this structural prior motivates the *masked Flood complex*.

3.2 Masked Flood complex

Recently, Graf et al. (2025) introduced Flood complexes, a class of simplicial complexes that allow for efficient computation of PH of large-scale point clouds. We restate their definition below.

Definition 1. *Let $X, L \subset \mathbb{R}^d$ be finite sets. The Flood complex $\mathrm{Flood}_r(X, L)$ at radius $r \geq 0$ is*

$$\mathrm{Flood}_r(X, L) = \left\{ \sigma \in \mathrm{Del}(L) \colon \mathrm{conv}(\sigma) \subset \bigcup_{x \in X} B_r(x) \right\} \ . \tag{1}$$

The set X is the point cloud of interest whereas L is typically a small subset of X. As $\mathrm{Flood}_r(X, L)$ is a subcomplex of the Delaunay triangulation $\mathrm{Del}(L)$, its number of simplices is comparably small, allowing for fast PH computation, while the filtration values are still informed by the full point cloud X due to the requirement that a simplex σ is in $\mathrm{Flood}_r(X, L)$ only if its convex hull $\mathrm{conv}(\sigma)$ is covered, or "flooded" by the union of closed balls $\bigcup_{x \in X} B_r(x)$. Its filtration values can be computed efficiently in parallel on a GPU, with further speedups from precomputing, for each simplex σ, a candidate mask of points $x \in X$ on which the simplex's filtration value must be realized.

While originally introduced as a means for efficient PH computation, both landmark selection and masking can be used to *emphasize specific properties of the point cloud* in the resulting filtration and its persistent homology. The definition of the masked Flood complex below captures this very idea.

Definition 2. *Let $L \subset \mathbb{R}^d$ be a finite set, and for each $\sigma \in \mathrm{Del}(L)$ let $X_\sigma \subset \mathbb{R}^d$ be finite such that $X_\sigma \subset X_\tau$ for all $\tau \subset \sigma$. The masked Flood complex at radius $r \geq 0$ is the simplicial complex*

$$\mathrm{mFlood}_r(\{X_\sigma\}_{\sigma \in \mathrm{Del}(L)}, L) = \left\{ \sigma \in \mathrm{Del}(L) \colon \mathrm{conv}\,(\sigma) \subset \bigcup_{x \in X_\sigma} B_r(x) \right\} \ . \tag{2}$$

In practice, we have some point cloud X and a distinguished set of landmarks, and we assign to each simplex $\sigma \in \mathrm{Del}(L)$ a set $X_\sigma \subset X$, i.e., a mask that specifies which points of X contribute to the filtration value of σ. By construction, the masked Flood complex is then a subcomplex of the original Flood complex from Graf et al. (2025) on X, and they agree if $X_\sigma = X$ for all $\sigma \in \mathrm{Del}(L)$. We include the requirement that $\tau \subset \sigma$ implies $X_\tau \supset X_\sigma$ in Definition 2 as it ensures that at each radius r the masked Flood complex is a well-defined simplicial complex and therefore induces a filtration of $\mathrm{Del}(L)$. Indeed, denoting by d the Euclidean distance, the filtration value $f(\sigma)$ of each simplex $\sigma \in \mathrm{Del}(L)$ is $f(\sigma) = \max_{p \in \mathrm{conv}(\sigma)} \min_{x \in X_\sigma} d(p, x)$. Hence $\tau \subset \sigma$ implies $f(\tau) \leq f(\sigma)$, since

$$f(\tau) = \max_{p \in \mathrm{conv}(\tau)} \min_{x \in X_\tau} d(p, x) \leq \max_{p \in \mathrm{conv}(\sigma)} \min_{x \in X_\tau} d(p, x) \leq \max_{p \in \mathrm{conv}(\sigma)} \min_{x \in X_\sigma} d(p, x) = f(\sigma) \ . \tag{3}$$

Following Graf et al. (2025), we approximate filtration values using a grid on the simplices' convex hulls. In Appendix E we verify that this still results in a well-defined filtration in the presence of masking, and, for fixed landmark positions, bottleneck stability with respect to the masks.

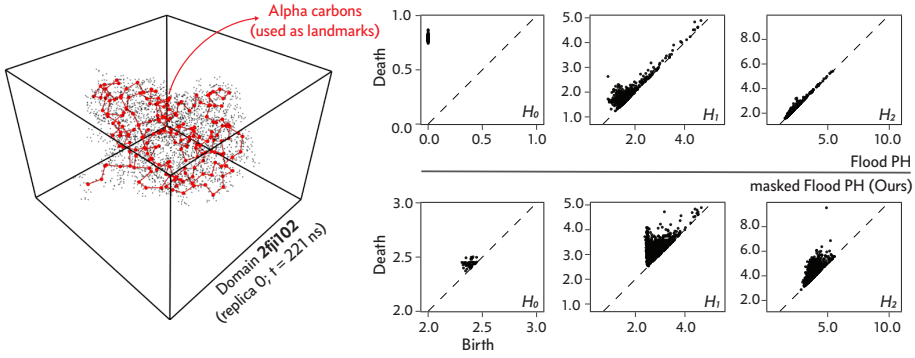


Figure 2: Left: Point cloud corresponding to a folded protein. Highlighted in red are the C_α atoms. **Right**: Persistence diagrams produced by Flood PH (top) and masked Flood PH (bottom).

For proteins, we use their C_α atoms as landmarks, i.e., simplices are sets of C_α atoms, and therefore correspond to sets of residues. To emphasize spatial relationships across residues, we design the masks as follows: For each vertex $l \in L$, its mask X_l consists only of atoms that are not part of the same residue. For higher-dim. simplices, we take the intersection of their vertex masks, i.e., the largest possible set of atoms that is compatible with the nesting condition for the masks from Definition 2. Thus, a simplex is flooded only by atoms from residues not represented by its vertices, e.g., an edge is flooded only by atoms from outside the two residues whose C_α atoms it connects.

Fig. 2 compares the resulting persistence diagrams to those from the Flood complex. The differences are most transparent in H_0 where, in contrast to most simplicial complexes, the features are *not* all born at radius $r = 0$, but at differing positive radii, because landmarks l are born at $\min_{x \in X_l} d(l, x)$, as $l \notin X_l$. Moreover, there are fewer points in H_0 than there are landmarks, as a vertex can appear at the same time as an edge connecting it to an existing connected component, and thus, without creating a new one (see Fig. I.6 in the appendix for an illustration). We also observe that more points in the H_1 and H_2 diagrams are persistent, i.e., well-separated from the diagonal. Using masks to encode structural information rather than solely for efficiency doubles the runtime relative to Flood PH. Still, masked Flood PH is more than three times faster than all-atom `Alpha PH`, since the complex is built only on the C_α landmarks and filtration values are computed in parallel on the GPU (see Table 1 and Section F).

Table 1: Wall time of PH computation (H_0–H_2) per persistence diagram (mean ± std over 500 `mdCATH` domains) and estimated total for all `mdCATH` conformations; cf. Table F.20.

	Per diagram (ms)	Total (h)
`Flood PH`	14.4 ± 7.1	46
`mFlood PH`	28.4 ± 21.1	91
`Alpha PH`	94.1 ± 63.5	301

3.3 Vectorizing persistence diagrams

To use persistence diagrams in learning tasks, we first need to represent them as vectors. In particular, to enable learning across molecular configurations, we require a map into a coordinate system consistent across frames and MD trajectories. To this end, we adopt a vectorization based on exponential structure elements (Hofer et al., 2019), which maps persistence diagrams (up to dimension 2) into a vector by using the same structure elements (fitted on training data) for all diagrams.

In the following, we (1) probe the (PH + vectorization) representations for their structural information content (Section 4), (2) assess whether they support learning low-dimensional kinetic embeddings shared across protein domains for MSM estimation (Section 5.1), and (3) use the MSMs to guide sampling of training pairs for a generative model (Section 5.2).

4 Probing structural representation quality

We evaluate the extent to which vectorized persistence diagrams capture *structurally relevant* information in molecular systems. Evaluation in terms of *dynamic information* is done in Section 5. For comparison to masked Flood PH (`mFlood PH`), we evaluate PH (up to dim. 2) computed from Alpha complexes on the full point cloud X (`Alpha PH`), Alpha & Rips complexes on the C_α atoms (`Alpha PH` (C_α), `Rips PH` (C_α)), and Flood PH (Graf et al., 2025) with C_α atoms as landmarks (`Flood PH`).

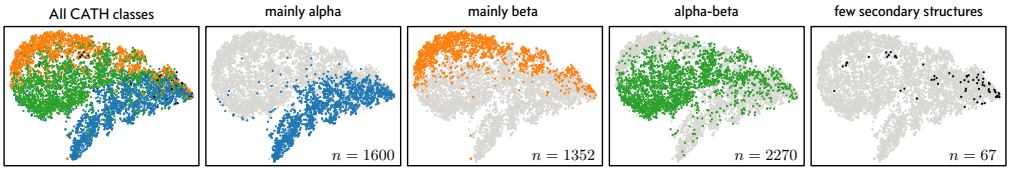


Figure 3: t-SNE embeddings of mean-pooled `mFlood PH` vectorizations of the 450 K `mdCATH` trajectories.

Datasets We rely on the `mdCATH` (Mirarchi et al., 2024) dataset, inheriting a train-val-test split from Kapuśniak et al. (2026) that enforces strict sequence dissimilarity between train and test domains. `mdCATH` consists of MD trajectories from 5,398 domains. For each domain, we use the five simulation replicas at 450 K; most trajectories are 500 ns long, with coordinates recorded every 1 ns.

Implementation To isolate representational quality, we probe the vectorized persistence diagrams with simple models: a single linear layer, and 2-layer MLPs with ELU activations. Persistence diagrams are vectorized using 128 exponential structure elements (Hofer et al., 2019) per H_0, H_1 and H_2 diagram, yielding a 384-dim. representation. We refer to Appendix C.1 for a description of the vectorization method and to Appendix C.2 for architectural and training details.

We probe two complementary aspects of representation quality: *First*, we evaluate protein class prediction at the replica level, assessing whether time-aggregated representations retain global structural discriminability. *Second*, we regress physical observables, testing whether the representations capture instantaneous conformational variability. Specifically, we seek to predict the radius of gyration (`RG`), the secondary structure fraction (`SSF`), the fraction of native contacts (`FNC`), and the root mean-square deviation (`RMSD`) to the replica's starting frame (see Appendix B for their definition).

Baselines As a *physical baseline*, we encode protein domains as (num. residues, 7)-shaped sequences, with residues represented by the sines and cosines of their ϕ- and ψ-torsion angles, and the 3D coordinates of their α-carbons. On these sequences, we train a 1D convolutional network with windowing along the residue axis. This respects the sequential nature of proteins and allows handling different domain sizes. For architectural details of the baseline and our approach see Appendix C.2.

As *foundation model baselines*, we include `ESM2` (Lin et al., 2023) and the inverse folding model `ESM-IF1` (Hsu et al., 2022). Their pretraining data, UniRef sequences and AlphaFold2 predictions of such sequences, may include our test domains, which cannot be verified. It can for the CATH training split of `ESM-IF1`: 53.5% of our test domains are identical or near-identical to one of its structures (Section C.4), so we gray out all results involving `ESM-IF1`.

As *learned end-to-end baselines* providing strong task-specific performance references, we use `EGNN` (Satorras et al., 2021) and `GotenNet` (Aykent & Xia, 2025) trained directly on the downstream tasks.

Classification of top-level protein class We consider protein classification at the trajectory level, by assigning each MD simulation replica of a protein domain to one of four CATH classes (mainly alpha, mainly beta, alpha-beta and few secondary structures). We obtain fixed-length features by averaging vectorized persistence diagrams over time. The 2D t-SNE embeddings in Fig. 3 already suggest good class separability for the `mFlood PH` representations.

For a quantitative analysis, we train a linear classifier and a small MLP and report balanced accuracy in Table 2. In addition to the aforementioned baselines, we also include a comparison with a feature set based on four physical observables (`RG`, `SSF`, `FNC`, `RMSD`).

All PH-based representations retain more class information than the physical references and, for most, at least as much as the end-to-end models. Compared with foundation-model representations, we see that

Table 2: Per-replica classification of CATH class. Balanced accuracy (mean ± std over 3 runs) in %. **Best**/second-best result excluding `ESM-IF1`.

	Linear	MLP
`mFlood PH`	79.1 ± 0.7	**81.2** ± 2.1
`Flood PH`	78.4 ± 2.0	80.0 ± 1.6
`Alpha PH` (C_α)	78.4 ± 1.0	78.2 ± 5.8
`Alpha PH`	71.3 ± 3.4	74.5 ± 3.1
`Rips PH` (C_α)	**79.5** ± 0.8	77.4 ± 5.3
(`RG`, `SSF`, `FNC`, `RMSD`)	54.3 ± 0.6	60.5 ± 0.4
(ϕ, ψ) + C_α coords.	–	70.8 ± 0.1
`ESM2_35m`	**79.5** ± 0.2	77.3 ± 0.6
`ESM-IF1`	82.7 ± 1.8	78.7 ± 3.6
(`ESM2_35m`, `ESM-IF1`)	81.5 ± 1.6	78.8 ± 1.8
`EGNN`	–	71.8 ± 5.1
`GotenNet`	–	75.5 ± 7.9

linear probes of Rips PH (C_α) and mFlood PH match those of ESM2_35m, the best of four ESM2 sizes (Table D.8), and that mFlood PH exceeds them when probed by the same MLP. The highest accuracy is obtained from ESM-IF1 representations with a linear probe, within one standard deviation of MLP probes of mFlood PH. *Overall*, topological summaries capture global conformational information beyond simple physical descriptors, on par with end-to-end trained models and learned representations (despite their advantage on the test split).

Regression of physical observables We probe how well common structural descriptors (RG, SSF, RMSD, FNC) can be read out from a representation. These observables are standard in the analysis of MD simulations and describe structure at complementary scales: SSF reflects local secondary structure, FNC the preservation of native tertiary contacts (Best et al., 2013), and RG and RMSD global compactness and overall deviation from the starting conformation.

For each target, we train a separate scalar regression model on all time points and all replicas of the training domains and evaluate across all time points of all replicas of the test domains. Table 3 lists the coefficient of determination (R^2) across all targets.

Table 3: Regression of physical observables per time point across all testing domains of mdCATH. Reported is the mean coefficient of determination, $R^2 \uparrow$, computed over training runs of the linear and MLP model from different random seeds. Standard deviations are at most 0.005 for fixed representations and up to 0.026 for end-to-end models (Table D.9). **Best**/second-best PH-based result.

	Linear				MLP			
	FNC	RG	RMSD	SSF	FNC	RG	RMSD	SSF
mFlood PH	**0.691**	0.728	0.683	0.652	0.860	0.861	0.781	0.842
Flood PH	0.686	0.719	0.675	**0.658**	**0.863**	0.864	0.782	0.853
Alpha PH	0.668	**0.831**	**0.708**	0.623	0.858	**0.921**	**0.798**	0.815
Alpha PH (C_α)	0.669	0.675	0.656	0.650	0.859	0.849	0.772	**0.854**
Rips PH (C_α)	0.630	0.559	0.597	0.599	0.840	0.772	0.732	0.828
(ϕ, ψ) + C_α coords.	–	–	–	–	0.762	0.332	0.578	0.841
ESM-IF1	0.918	0.904	0.822	0.933	0.929	0.946	0.849	0.934
(ESM2_35m, ESM-IF1)	0.929	0.905	0.829	0.932	0.942	0.944	0.853	0.934
EGNN	–	–	–	–	0.878	0.955	0.799	0.863
GotenNet	–	–	–	–	0.879	0.953	0.808	0.868

Most targets are recovered better from PH-based representations than from the physical reference. End-to-end GNNs recover RG best, and pretrained ESM-IF1 and its fusion with ESM2_35m excel in recovering the remaining observables, in line with their task-specific training and scale.

5 Dynamics

We assess the utility of vectorized persistence diagrams in a series of experiments building upon each other. We first use them to fit MSMs on mdCATH, which we then use as a drop-in replacement for the ones originally used (based on physical observables) to guide the training of MarS-FM. Finally, we study the transferability of the obtained MarS-FM model to fast folding proteins (via finetuning).

5.1 Learning a shared kinetic embedding space and MSM estimation

To study the dynamic information captured by vectorized persistence diagrams methodically, we rely on MSMs to assess the capability of features to resolve kinetically relevant processes; typically the slowest modes of a dynamical system, often associated with rare but meaningful transitions (Prinz et al., 2011). To construct an MSM, conformations are usually described by physical observables, or projected onto the directions of slowest decorrelation via TICA (Molgedey & Schuster, 1994; Schwantes & Pande, 2013; Perez-Hernandez et al., 2013). The resulting representations are then discretized by clustering, optionally followed by coarse-graining (Röblitz & Weber, 2013). Usually, this workflow is performed on one or more MD trajectories of a *single system*, and the resulting MSM quality is tied to the amount and quality of the available data, as well as the chosen descriptors. The availability of datasets comprising trajectories from diverse protein domains motivates leveraging population-level information by learning a shared representation space *across domains*.

To this end, we adapt the VAMPNet framework (Mardt et al., 2018) to train a shared model across systems (cf. Appendix C.5). After training, we map all trajectory frames to the shared latent space and, for each domain, discretize the representations of each replica into k states using k-means++ (Arthur & Vassilvitskii, 2007). The resulting discrete trajectories are then used for MSM estimation.

Implementation As input for the kinetic embedding we use the PH vectorizations of Section 4 based on the 450 K simulations of `mdCATH`. As VAMPNet, we use a simple 5-layer MLP with a 7D output (for training details, see Appendix C.5). MSMs are estimated per domain by maximum likelihood with a reversible transition matrix from effective transition counts at lag 50 ns, restricted to the largest connected set.[1] The choice of the lag time and the temperature for MD simulations is determined by the subsequent generative modeling experiment for which we reuse the MSMs, as we want to preserve the setup from Kapuśniak et al. (2026) for a sound comparison.

Quality measures To assess the quality of the produced MSMs, we use three complementary quality measures: `vamp-2`, which measures how well the slow processes in the data are resolved, `CK RMSE` computed from Chapman–Kolmogorov (CK) tests which measure whether the resulting state decomposition yields Markovian dynamics at the chosen lag time, and π-`JS`, the Jensen–Shannon divergence between the MSM-implied stationary distribution and the empirical distribution. As we reserve the testing portion of `mdCATH` for evaluating the generative model in Section 5.2, we report scores on its validation split. For each domain, we fit MSMs on four of the five MD replicas and compute the `vamp-2` and π-`JS` on the held-out replica. The resulting five scores are averaged per domain, and Table 4 reports the mean and standard deviation across domains. Due to the short `mdCATH` trajectories, cross-validating the CK test is unstable (see Appendix B). In any case, it stands as a valid self-consistency check for Markovianity, rather than a generalization measure.

Baselines We compare against MSMs fitted on (i) representations from a convolutional VAMPNet trained on backbone torsion angles (ϕ, ψ) and C_α coordinates, (ii) the physical observables (`RG`, `SSF`), as in Kapuśniak et al. (2026), (iii) 7-dimensional TICA projections of commonly used descriptors: torsion angles (sine-cosine pairs of backbone and sidechain angles) and the solvent-accessible surface area (SASA) in the main part and other typically used descriptors in Appendix D.2.2.

Results `mFlood PH` is the only representation on par with or better than (`RG`, `SSF`) on all three measures, although the latter are tailored to unfolding, the dominant slow process at 450 K. Even with one embedding shared across domains, `mFlood PH` captures the slow processes as well as these hand-picked observables and yields more Markovian MSMs. Masking is decisive: relative to `Flood PH`, it lowers `CK RMSE` by a third and nearly halves π-`JS`. `mFlood PH`-based MSMs could thus replace (`RG`, `SSF`) in `MarS-FM` without loss in MSM quality.

Table 4: MSM quality measures at lag time $\tau = 50$ ns with 7-dimensional VAMPNet features and 10 micro states. `CK RMSE` is evaluated at lag times 50 and 100 ns on 3 macro states. **Best**/second-best overall.

	`vamp-2` ↑	`CK RMSE` ↓	π-`JS` ↓
`mFlood PH`	1.503 ± .010	**0.034** ± .001	0.120 ± .004
`Flood PH`	1.482 ± .013	0.051 ± .002	0.219 ± .006
`Alpha PH` (C_α)	1.505 ± .016	0.053 ± .002	0.246 ± .005
`Alpha PH`	1.463 ± .010	**0.034** ± .001	**0.109** ± .004
`Rips PH` (C_α)	1.477 ± .014	0.049 ± .001	0.254 ± .006
(ϕ, ψ) + C_α coords.	1.113 ± .004	0.046 ± .001	0.332 ± .004
(`RG`, `SSF`)	**1.513** ± .004	0.039 ± .001	0.124 ± .004
`TICA (Torsion)`	1.264 ± .002	0.076 ± .000	0.139 ± .001
`TICA (SASA)`	1.439 ± .002	**0.034** ± .000	0.258 ± .001

5.2 Generative Modeling

In `MarS-FM` (Kapuśniak et al., 2026), a generative model learns by flow matching to produce a target frame from a source frame, with pairs drawn from MSM state transitions of each domain. We replace its (`RG`, `SSF`)-based MSMs by PH-based ones to test whether the kinetic information of PH carries over to generation, on held-out `mdCATH` domains and, after finetuning, on fast folding proteins.

mdCATH With all other settings of Kapuśniak et al. (2026) unchanged, we compare generated and MD ensembles in conformational diversity (pairwise RMSD), residue flexibility (RMSF), compactness (`RG`), secondary structure (`SSF`) and folding free energy (ΔG), as defined in Appendix B. All PH variants improve on the original `MarS-FM` in every statistic of Table 5 except RG KL, where they are closer to the `MD oracle`. As the differences are often small, we also test their significance: five of the six significant differences favor `MarS-FM-mFlood` (Table D.16; full results in Appendix D.2.3).

[1] Although individual `mdCATH` trajectories are short ($\leq$ 500 ns), they suffice for valid MSMs (Appendix D.2.1).

Table 5: Ensemble statistics on the mdCATH test domains, mean ± 1 std. dev. over 5 evaluation runs; MarS-FM from its released checkpoint. MD oracle: one MD replica scored against all five; MarS-FM-random: MSMs fitted to 2D Gaussian random features. **Best** result, excluding oracle and random.

	Pairwise RMSD r ↑	Global RMSF r ↑	RG KL ↓	SSF KL ↓	SSF JSD ↓	ΔG MAE ↓
MD oracle	0.82 ± .014	0.82 ± .012	0.58 ± .020	0.61 ± .048	0.07 ± .003	0.90 ± .046
MarS-FM-random	0.57 ± .004	0.68 ± .003	0.51 ± .002	1.52 ± .009	0.18 ± .001	1.36 ± .007
MarS-FM	0.64 ± .002	0.71 ± .002	**0.55** ± .002	1.05 ± .007	0.14 ± .000	0.99 ± .009
MarS-FM-mFlood	**0.68** ± .003	**0.74** ± .001	0.58 ± .003	0.81 ± .005	**0.12** ± .000	0.96 ± .012
MarS-FM-Flood	0.66 ± .002	0.73 ± .001	0.58 ± .002	0.81 ± .007	**0.12** ± .000	0.97 ± .011
MarS-FM-Alpha (C_α)	0.67 ± .002	0.72 ± .001	0.60 ± .003	**0.78** ± .006	**0.12** ± .000	**0.91** ± .004
MarS-FM-Alpha	**0.68** ± .004	**0.74** ± .002	0.59 ± .002	0.85 ± .007	**0.12** ± .000	0.92 ± .005

Transfer to fast folding proteins While mdCATH covers a broad structural space, its trajectories are too short to resolve asymptotic behavior. To address this limitation, following Costa et al. (2025), we assess MarS-FM-mFlood on the long-timescale trajectories of Majewski et al. (2023), which provide ∼7 ms of aggregate simulation time for 12 fast folding proteins (Lindorff-Larsen et al., 2011). Compared to mdCATH, these proteins are much smaller (10–80 residues), simulated at 350 K, and exhibit strong folding propensity ("fast folders"), representing the opposite of the unfolding dynamics within the 450 K mdCATH simulations. To account for this disparity *without* training on the fast folders, we finetune (ft) our MarS-FM-mFlood model on the 348 K subset of mdCATH, reduce lag time to 10 ns, and modify the sampling procedure by drawing the starting-frame cluster as one of the target rather than the source conformation. We obtain native-state conformations for *all* 12 fast folding proteins (see Table D.18) and promising structural validation measures (Table 6): RMSZ deviations in covalent backbone geometry, comparing to Engh-Huber bond and angle parameters (Engh & Huber, 1991), and folding free energy ΔG_{fold}. Covalent geometry is of similar quality for all proteins, including the smallest ones, for which Costa et al. (2025) report chemically invalid states (Appendix D.4).

Table 6: Structural validation of "fast folders". Metrics are averaged over all reference (MD), resp. generated, frames.

	Bond RMSZ ↓	Angle RMSZ ↓	ΔG_{fold} (kcal/mol)
MD reference	1.96 ± 0.02	1.86 ± 0.04	1.45 ± 0.67
MarS-FM	5.78 ± 0.89	3.21 ± 0.17	4.90 ± 1.01
MarS-FM-mFlood	4.96 ± 0.74	3.10 ± 0.22	4.62 ± 0.81
MarS-FM$^{\text{ft}}$	3.72 ± 0.53	2.44 ± 0.15	2.95 ± 1.03
MarS-FM-mFlood$^{\text{ft}}$	3.16 ± 0.28	2.36 ± 0.16	2.79 ± 0.96

6 Discussion

We set out to assess whether principled topological descriptors can serve as general-purpose representations of protein structure and dynamics without large-scale pretraining. To this end, we introduced the masked Flood complex, a protein-tailored simplicial complex that emphasizes inter-residue structure. Its vectorized persistence diagrams then place conformations of different proteins in a shared space. Our probing results show that they recover both global fold and per-frame conformational variability at a level comparable to learned protein representations. MSMs built on our descriptors improve ensemble statistics when used within MarS-FM, and the resulting generative model transfers to fast folding proteins after finetuning. While our results also show that no PH variant is best across all probes and downstream tasks (e.g., Alpha PH on all atoms performing well in many cases), the masked Flood complex (mFlood) is the only construction among the strongest in every evaluation, which makes it the most consistent PH representation overall (Table D.17). Furthermore, computation time stays within a third of the time of all-atom Alpha PH (Table 1). Future work could design masks for higher-order simplices systematically and summarize trajectory segments topologically, so that kinetic information enters the representation directly rather than only through a learned embedding. Overall, masked Flood PH provides a computationally efficient means to incorporate prior knowledge into the construction of a simplicial complex whenever structure can be encoded as masks.

ETHICS STATEMENT

This work does not involve human subjects or personal data. All datasets we use, `mdCATH` (Mirarchi et al., 2024) and the fast folding protein trajectories of Majewski et al. (2023), are publicly available simulation data and are used as provided by their authors. We discuss potential societal impacts in Appendix H. We are not aware of conflicts of interest related to this work.

REPRODUCIBILITY STATEMENT

Appendix A describes data sources, splits and preprocessing for `mdCATH` and the fast folding proteins, and Appendix B defines all physical observables and evaluation metrics. Persistence diagram vectorization, the representation probes, all baselines, the cross-domain VAMPNet, and the training and generation protocols of `MarS-FM`, including finetuning for the fast folding proteins, are specified in Appendix C, together with the computing infrastructure. We provide our code, including configuration files for all experiments, as supplementary material.

AI USE STATEMENT

In this work, we used generative AI tools to help with implementation of experiments. Additionally, we used generative AI tools to search for related work, to format tables, to help draft and edit text, and to check the manuscript for consistency with the underlying results. All AI-assisted code was reviewed by the authors and tested, and all reported numbers were checked against the underlying result files. We have not used generative AI tools to generate synthetic data, to formulate mathematical claims or assist with proofs, or to propose research hypotheses. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Reem Aboelsoud, Javad Kasravi, Stefan Kesselheim, and Alina Bazarova. When protein dynamics matter: Integrating molecular dynamics into protein foundation models. In *ICLR 2026 Workshop on Foundation Models for Science: Real-World Impact and Science-First Design*, 2026. URL https://openreview.net/forum?id=Q6DuJPwH2U.

Akashnathan Aranganathan, Xinyu Gu, Dedi Wang, Bodhi P. Vani, and Pratyush Tiwary. Modeling Boltzmann-weighted structural ensembles of proteins using artificial intelligence–based methods. *Current Opinion in Structural Biology*, 91:103000, 2025. doi:10.1016/j.sbi.2025.103000.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, 2007.

Sarp Aykent and Tian Xia. GotenNet: Rethinking Efficient 3D Equivariant Graph Neural Networks. *ICLR*, 2025.

Robert B. Best, Gerhard Hummer, and William A. Eaton. Native contacts determine protein folding mechanisms in atomistic simulations. *PNAS*, 110(44):17874–17879, 2013. doi:10.1073/pnas.1311599110.

Filippo Bigi, Sanggyu Chong, Agustinus Kristiadi, and Michele Ceriotti. FlashMD: long-stride, universal prediction of molecular dynamics. In *NeurIPS*, 2025.

Thomas B. Blank, Steven D. Brown, August W. Calhoun, and Douglas J. Doren. Neural network models of potential energy surfaces. *The Journal of Chemical Physics*, 103(10):4129–4137, 1995. doi:10.1063/1.469597.

Zixuan Cang and Guo-Wei Wei. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics*, 33(22):3549–3557, 2017a. doi:10.1093/bioinformatics/btx460.

Zixuan Cang and Guo-Wei Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7): e1005690, 2017b. doi:10.1371/journal.pcbi.1005690.

Zixuan Cang, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Molecular Based Mathematical Biology*, 3(1):140–162, 2015. doi:10.1515/mlbmb-2015-0009.

Zixuan Cang, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLOS Computational Biology*, 14(1): e1005929, 2018. doi:10.1371/journal.pcbi.1005929.

Huang-Wei Chang, Sergio Bacallado, Vijay S. Pande, and Gunnar E. Carlsson. Persistent topology and metastable state in conformational dynamics. *PLOS ONE*, 8(4):e58699, 2013. doi:10.1371/journal.pone.0058699.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007.

Allan Dos Santos Costa, Manvitha Ponnapati, Dana Rubin, Tess Smidt, and Joseph Jacobson. Accelerating protein molecular dynamics simulation with DeepJump. In *NeurIPS 2025 AI for Science Workshop*, 2025.

Ameya Daigavane, Bodhi P. Vani, Darcy Davidson, Saeed Saremi, Joshua A Rackers, and Joseph Kleinhenz. JAMUN: Bridging smoothed molecular dynamics and score-based learning for conformational ensemble generation. In *NeurIPS*, 2025.

Tamal K. Dey and Sayan Mandal. Protein classification with improved topological data analysis. In *18th International Workshop on Algorithms in Bioinformatics (WABI 2018)*, volume 113 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 6:1–6:13, 2018. doi:10.4230/LIPIcs.WABI.2018.6.

Luca Donati, Surahit Chewle, Dominik St Pierre, Vijay Natarajan, and Marcus Weber. Topological Analysis Reveals Multiple Pathways in Molecular Dynamics. *Journal of Chemical Theory and Computation*, 21(20):10385–10397, 2025. doi:10.1021/acs.jctc.5c00819.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. arXiv:2401.08281, 2024.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE TPAMI*, 44(10):7112–7127, October 2022. doi:10.1109/TPAMI.2021.3095381.

Richard A. Engh and Robert Huber. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallographica Section A*, 47(4):392–400, 1991. doi:10.1107/S0108767391001071.

Klemens Flöge, Srisruthi Udayakumar, Johanna Sommer, Marie Piraud, Stefan Kesselheim, Vincent Fortuin, Stephan Günnemann, Karel J. van der Weg, Holger Gohlke, Erinc Merdivan, and Alina Bazarova. OneProt: Towards multi-modal protein foundation models via latent space alignment of sequence, structure, binding sites and text encoders. *PLOS Computational Biology*, 21(11): e1013679, November 2025. ISSN 1553-7358. doi:10.1371/journal.pcbi.1013679.

Pascal Friederich, Florian Häse, Jonny Proppe, and Alán Aspuru-Guzik. Machine-learned potentials for next-generation matter simulations. *Nature Materials*, 20(6):750–761, 2021. doi:10.1038/s41563-020-0777-6.

Florian Graf, Paolo Pellizzoni, Martin Uray, Stefan Huber, and Roland Kwitt. The Flood Complex: Large-Scale Persistent Homology on Millions of Points. In *NeurIPS*, 2025.

Bingqing Han, Yipeng Zhang, Longlong Li, Xinqi Gong, and Kelin Xia. TopoQA: a topological deep learning-based approach for protein complex structure interface quality assessment. *Briefings in Bioinformatics*, 26(2):bbaf083, 2025. doi:10.1093/bib/bbaf083.

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

Michael Heinzinger and Burkhard Rost. Teaching AI to speak protein. *Current Opinion in Structural Biology*, 91:102986, April 2025. doi:10.1016/j.sbi.2025.102986.

C. Hofer, R. Kwitt, and M. Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

Moritz Hoffmann, Martin Konrad Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke Elena Husic, Stefan Klus, Hao Wu, J Nathan Kutz, Steven Brunton, and Frank Noé. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 2021.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *ICML*, 2022.

Takashi Ichinomiya. Topological data analysis gives two folding paths in HP35(Nle-Nle), double mutant of villin headpiece subdomain. *Scientific Reports*, 12(1):2719, 2022. doi:10.1038/s41598-022-06682-x.

Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. Protein-Folding Analysis Using Features Obtained by Persistent Homology. *Biophysical Journal*, 118(12):2926–2937, 2020. doi:10.1016/j.bpj.2020.04.032.

Marcus Jäger, Yan Zhang, Jan Bieschke, Houbi Nguyen, Maria Dendle, Marianne E. Bowman, Joseph P. Noel, Martin Gruebele, and Jeffery W. Kelly. Structure–function–folding relationship in a WW domain. *Proceedings of the National Academy of Sciences*, 103(28):10648–10653, 2006. doi:10.1073/pnas.0600511103.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations*, 2021.

Bowen Jing, Hannes Stark, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. In *NeurIPS*, 2024.

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

Kacper Kapuśniak, Cristian Gabellini, Michael M. Bronstein, Prudencio Tossou, and Francesco Di Giovanni. MarS-FM: Generative modeling of molecular dynamics via Markov state models. In *ICLR*, 2026.

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

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. doi:10.1126/science.ade2574.

Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How Fast-Folding Proteins Fold. *Science*, 334(6055):517–520, 2011. doi:10.1126/science.1208351.

Christian D. Madsen, Agnese Barbensi, Stephen Y. Zhang, Lucy Ham, Alessia David, Douglas E. V. Pires, and Michael P. H. Stumpf. The topological properties of the protein universe. *Nature Communications*, 16:7503, 2025. doi:10.1038/s41467-025-61108-2.

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. doi:10.1038/s41467-023-41343-1.

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5, 2018. doi:10.1038/s41467-017-02388-1.

J. Andrew McCammon, Bruce R. Gelin, and Martin Karplus. Dynamics of folded proteins. *Nature*, 267(5612):585–590, 1977. doi:10.1038/267585a0.

Robert T. McGibbon and Vijay S. Pande. Variational cross-validation of slow dynamical modes in molecular kinetics. *The Journal of Chemical Physics*, 142(12):124105, March 2015. doi:10.1063/1.4916292.

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. doi:10.1038/s41597-024-04140-z.

Lutz Molgedey and Heinz G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994. doi:10.1103/PhysRevLett.72.3634.

Viet Thanh Duy Nguyen and Truong-Son Hy. Advances in Protein Representation Learning: Methods, Applications, and Future Directions. arXiv:2503.16659, May 2025.

Frank Noé, Simon Olsson, Jonas Kähler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi:10.1126/science.aaw1147.

Feliks Nüske, Hao Wu, Jan-Hendrik Prinz, Christoph Wehmeyer, Cecilia Clementi, and Frank Noé. Markov state models from short non-equilibrium simulations—Analysis and correction of estimation bias. *The Journal of Chemical Physics*, 146(9):094104, March 2017. doi:10.1063/1.4976518.

Andreas Ostermann, Robert Waschipky, Fritz G. Parak, and G. Ulrich Nienhaus. Ligand binding and conformational motions in myoglobin. *Nature*, 404(6774):205–208, 2000. doi:10.1038/35004622.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023.

Guillermo Perez-Hernandez, 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):015102, 2013. doi:10.1063/1.4811489.

Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noe. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models. In *NeurIPS*, 2025.

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):174105, 2011. doi:10.1063/1.3565032.

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(2):147–179, 2013. doi:10.1007/s11634-013-0134-6.

Ismael Rodríguez-Espigares, Mariona Torrens-Fontanals, Johanna K. S. Tiemann, David Aranda-García, Juan Manuel Ramírez-Anguita, Tomasz Maciej Stepniewski, Nathalie Worp, Alejandro Varela-Rial, Adrián Morales-Pastor, Brian Medel-Lacruz, Gáspár Pándy-Szekeres, Eduardo Mayol, Toni Giorgino, Jens Carlsson, Xavier Deupi, Slawomir Filipek, Marta Filizola, José Carlos Gómez-Tamayo, Angel Gonzalez, Hugo Gutiérrez-de-Terán, Mireia Jiménez-Rosés, Willem Jespers, Jon Kapla, George Khelashvili, Peter Kolb, Dorota Latek, Maria Marti-Solano, Pierre Matricon, Minos-Timotheos Matsoukas, Przemyslaw Miszta, Mireia Olivella, Laura Perez-Benito, Davide Provasi, Santiago Ríos, Iván R. Torrecillas, Jessica Sallander, Agnieszka Sztyler, Silvana Vasile, Harel Weinstein, Ulrich Zachariae, Peter W. Hildebrand, Gianni De Fabritiis, Ferran

Sanz, David E. Gloriam, Arnau Cordomi, Ramon Guixà-González, and Jana Selent. GPCRmd uncovers the dynamics of the 3D-GPCRome. *Nature Methods*, 17(8):777–787, August 2020. doi:10.1038/s41592-020-0884-y.

M. Royer, F. Chazal, C. Levrard, Y. Umeda, and Y. Ike. ATOL: Measure vectorization for automatic topologically-oriented learning. In *AISTATS*, 2021.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. $E(n)$ Equivariant Graph Neural Networks. In *ICML*, 2021.

Martin K. Scherer, Brooke E. Husic, Moritz Hoffmann, Fabian Paul, Hao Wu, and Frank Noé. Variational Selection of Features for Molecular Kinetics. *The Journal of Chemical Physics*, 150 (19):194108, 2019. doi:10.1063/1.5083040.

Ch Schütte, A Fischer, W Huisinga, and P Deuflhard. A Direct Approach to Conformational Dynamics Based on Hybrid Monte Carlo. *Journal of Computational Physics*, 151(1):146–168, May 1999. doi:10.1006/jcph.1999.6231.

Christian R. Schwantes and Vijay S. Pande. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. *Journal of Chemical Theory and Computation*, 9(4):2000–2009, 2013. doi:10.1021/ct300878a.

Nina Singhal, Christopher D. Snow, and Vijay S. Pande. Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *The Journal of Chemical Physics*, 121(1):415–425, 2004. doi:10.1063/1.1738647.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. doi:10.1038/nbt.3988.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. In *ICLR*, 2024.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and the UniProt Consortium. UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015. doi:10.1093/bioinformatics/btu739.

Rakesh Trivedi and Hampapathalu Adimurthy Nagarajaram. Intrinsically Disordered Proteins: An Overview. *International Journal of Molecular Sciences*, 23(22):14050, 2022. doi:10.3390/ijms232214050.

Oliver T. Unke, Stefan Chmiela, Huziel E. Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T. Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine Learning Force Fields. *Chemical Reviews*, 121(16):10142–10186, 2021. doi:10.1021/acs.chemrev.0c01111.

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. doi:10.1093/nar/gkad1084.

Donald Voet, Judith G. Voet, and Charlotte W. Pratt. *Fundamentals of Biochemistry: Life at the Molecular Level*. Wiley, Hoboken, 5th edition, 2016. ISBN 978-1-118-91840-1.

Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International Journal for Numerical Methods in Biomedical Engineering*, 30(8):814–844, 2014. doi:10.1002/cnm.2655.

Yuan Yao, Jian Sun, Xuhui Huang, Gregory R. Bowman, Gurjeet Singh, Michael Lesnick, Leonidas J. Guibas, Vijay S. Pande, and Gunnar Carlsson. Topological methods for exploring low-density states in biomolecular folding pathways. *The Journal of Chemical Physics*, 130:144115–144115, 2009. doi:10.1063/1.3103496.

Adam Zemla. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374, July 2003. doi:10.1093/nar/gkg571.

Si Zhang and Gregory R. Bowman. Decrypting cryptic pockets with physics-based simulations and artificial intelligence. *Current Opinion in Structural Biology*, 96:103215, 2026. doi:10.1016/j.sbi.2025.103215.

A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete Computational Geometry*, 33(2):249–274, 2005.

Robert Zwanzig. From classical dynamics to continuous time random walks. *Journal of Statistical Physics*, 30(2):255–262, 1983. doi:10.1007/BF01012300.

Supplementary Material

In the following supplementary parts to the main manuscript, we provide dataset descriptions in Appendix A, definitions of the physical observables and evaluation metrics in Appendix B, discuss implementation details for training and finetuning in Appendix C, and provide supplementary experiments in Appendix D.

A Data

A.1 mdCATH

`mdCATH` (Mirarchi et al., 2024) provides all-atom MD trajectories for 5,398 CATH domains (50–500 residues, mean 137), each simulated in five replicas at five temperatures (320, 348, 379, 413, 450 K) for mainly 500 ns with coordinates recorded at 1 ns frame interval. We adopt the split of Kapuśniak et al. (2026): an 80/10/10 partition of domains into train, validation, and test, with test domains sharing no more than 20% sequence similarity with any training domain (MMseqs2 (Steinegger & Söding, 2017), maximum sensitivity). We additionally exclude 48 domains with faulty trajectories as documented in `https://github.com/compsciencelab/mdCATH/issues/7`. We train on the 450 K replicas, where (partial) unfolding produces conformational diversity within the simulated window; lower-temperature replicas remain close to the native basin.

A.2 Fast Folders

While `mdCATH` covers a large structural space, its individual trajectory lengths on the order of 500 ns are too short to resolve asymptotic behavior. We therefore, additionally and as Costa et al. (2025), evaluate our model on trajectories from Majewski et al. (2023) who, through adaptive sampling, achieved thorough exploration of phase space. The twelve proteins are the fast folders of Lindorff-Larsen et al. (2011), small domains with α-helical, β-sheet and mixed folds that fold and unfold reversibly on timescales accessible to simulation. Data is downloaded as described at `https://github.com/torchmd/torchmd-protein-thermodynamics/tree/main/Datasets`. As seen in Table A.7, we match Table 1 of Majewski et al. (2023) for 9 of the 12 proteins.

For reproducibility, we assemble each per-protein data from the `*filtered.xtc` trajectories: the curated subset already prepared for downstream analysis paired with their `filtered.pdb` topologies. We excluded part of the protein G trajectories (`proteing_3_trajectories`), a sub-campaign still containing solvent; For NTL9 and λ-repressor conservatively using only trajectories which contained the "`filtered`" suffix in the file name, does not match the aggregated time of Majewski et al. (2023). Nonetheless, we estimate the data as more than sufficient as its aggregated time is on the same order of magnitude than those of the remaining 9 proteins ($\sim$0.2-1.4 ms). For Villin, norleucine residues are mapped to methionine, the closest standard residue, as `MarS-FM` supports only standard amino acids.

Following Majewski et al. (2023), the first 10% of frames of all trajectories are discarded to remove bias from the starting sample, the remainder is featurized by C_α-C_α distances and projected via TICA ($\tau = 10$ ns) onto 4 dimensions. In this low-dimensional space, per-protein trajectories are clustered into 32 states via k-means++ as in Costa et al. (2025). The native state is identified as the cluster with the minimal centroid distance to the respective crystal structure. For every protein, we fit an MSM ($\tau = 10$ ns) on all of its trajectory data (TIC space) to obtain the protein's stationary distribution with which we weigh the MD frames in our evaluation metrics.

Table A.7: Fast folder data. For each protein, we report the sequence length, the aggregated simulation time, and the minimum RMSD to the experimental crystal structure observed across all used trajectories of Majewski et al. (2023).

Protein	Number of residues	Aggregated time (μs)	Min. RMSD (Å)
Chignolin	10	186.9	0.15
Trp-Cage	20	196.9	0.45
BBA	28	364.5	1.13
WW-Domain	34	1374.6	0.73
Villin	35	234.7	0.47
NTL9	39	512.9	0.32
BBL	47	676.5	1.55
Protein B	47	609.4	1.21
Homeodomain	54	199.1	0.63
Protein G	56	1161.7	0.55
α3D	73	768.0	1.81
λ-repressor	80	784.0	1.52

B Definition of physical observables and evaluation metrics

RMSD The root mean-squared deviation (RMSD) is typically computed with respect to a reference conformation, e.g., a native crystal structure as in Table D.18. In case of the pairwise RMSD, we calculate it between a random subset of all pairs of structures (once for generated, once for reference MD trajectory) and compute the Pearson correlation coefficient between their distributions.

RMSF The root mean-squared fluctuation (RMSF) quantifies atomic flexibility and computes, for every C_α atom, the deviation of its position along a trajectory. That is for residue l,

$$\mathtt{RMSF}_l = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\mathbf{r}_{\alpha,l} - \bar{\mathbf{r}}_{\alpha,l}\|_2^2} \tag{B.4}$$

with $\mathbf{r}_{\alpha,l}$ the coordinate of residue l's α-carbon and $\bar{\mathbf{r}}_{\alpha,l}$ its mean over time T. The RMSF is related to the B factor,

$$\mathrm{B}_l = \frac{8\pi^2}{3}\mathtt{RMSF}_l^2 \ , \tag{B.5}$$

which can be measured experimentally through X-ray crystallography.

For the RMSF, we compute per-target and global Pearson correlations. In the per-target evaluation, the RMSF for a specific protein is compared between reference MD and generated trajectory. This metric discovers if generated and reference molecules differ in the relation of fluctuation *between* residues, e.g., if both reference and generated proteins have floppy ends and comparatively rigid interiors. It would not, however, capture if the generative model prescribes a greater mobility (or rigidity) to *all* residues. The *global* correlation will pick up exactly such trends by comparing the distribution of all reference protein simulations with the distribution of all generated proteins.

RMSZ To assess the backbone covalent geometry of generated structures, we compute root mean-square Z-scores (RMSZ) of bond lengths and bond angles against the empirical crystallographic targets of Engh & Huber (1991). For an observed backbone bond length ℓ_k and bond angle θ_k, with target means ℓ_k^*, θ_k^* and standard deviations $\sigma_k^{\ell*}$, $\sigma_k^{\theta*}$,

$$z_k^\ell = \frac{\ell_k - \ell_k^*}{\sigma_k^{\ell*}}, \qquad z_k^\theta = \frac{\theta_k - \theta_k^*}{\sigma_k^{\theta*}}, \tag{B.6}$$

and we report $\mathrm{RMSZ}_\gamma = \sqrt{\frac{1}{N_\gamma}\sum_{k=1}^{N_\gamma}(z_k^\gamma)^2}$ aggregated separately over all backbone bonds ($\gamma = \ell$) and bond angles ($\gamma = \theta$). Values close to those of the reference MD ensemble indicate physically realistic covalent geometry, whereas large RMSZ exposes systematic bond or angle distortions in generated samples.

We obtain crystallographic reference values `stereo_chemical_props.txt` from AlphaFold's GitHub repository (see https://github.com/google-deepmind/alphafold/issues/32 in case of retrieval issues).

Radius of Gyration The radius of gyration (RG) is defined as the root mean-squared distance to the protein's center of mass, i.e.,

$$\mathsf{RG} = \sqrt{\frac{\sum_{i=1}^{N} m_i \|\mathbf{r}_i - \bar{\mathbf{r}}\|_2^2}{\sum_{i=1}^{N} m_i}} \,. \tag{B.7}$$

SSF We compute the secondary structure fraction (SSF) as the fraction of residues that are part of

- G: 3-turn helix
- H: 4-turn helix (α helix)
- I: 5-turn helix (π helix)
- E: Extended strand in parallel and/or anti-parallel beta-sheet conformation
- B: Residue in isolated beta-bridge

such that for a protein with L residues,

$$\mathsf{SSF} = \frac{1}{L} \sum_{l}^{L} \mathbb{I}[\mathrm{DSSP}(l) \in \{\mathrm{G}, \mathrm{H}, \mathrm{I}, \mathrm{E}, \mathrm{B}\}] \tag{B.8}$$

with the indicator function $\mathbb{I}$, and the DSSP (dictionary of protein secondary structure) denoting the standard algorithm for secondary structure assignment.

Distribution comparison For `RG` and `SSF`, we compute the value of every frame of the reference MD (all replicas pooled) and of the generated trajectory, and histogram both with 100 bins over their joint range. `RG KL` and `SSF KL` report the Kullback–Leibler divergence $D_{\mathrm{KL}}(p_{\mathrm{MD}} \,\|\, p_{\mathrm{gen}})$, with empty bins of p_{gen} set to 10^{-5} before renormalization, and `SSF JSD` the Jensen–Shannon divergence; both in nats.

Stationary distribution π The stationary distribution π is the long-time equilibrium population over the MSM states: the unique probability vector left-invariant under the transition matrix $T(\tau)$

$$\pi^\top T(\tau) = \pi^\top, \qquad \sum_i \pi_i = 1, \tag{B.9}$$

i.e., the left eigenvector of $T(\tau)$ at eigenvalue $\lambda_1 = 1$, independent of the lag time τ. In practice, we obtain this quantity via `deeptime`'s (Hoffmann et al., 2021) `stationary_distribution` method on the computed MSM.

FNC We compute the fraction of native contacts (FNC) with the protocol of Lewis et al. (2025), based on Best et al. (2013): For all C_α pairs $(C_{\alpha,i}, C_{\alpha,j})$ with sequence distance $|l_i - l_j| > 3$ and distance $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| < 10\text{Å}$, the FNC for structure x is

$$\mathsf{FNC}(x) = \frac{1}{P} \sum_{(i,j)} \frac{1}{1 + \exp\left(\beta(r_{ij}(x) - \lambda r_{ij}^0)\right)} \tag{B.10}$$

with $r_{ij}(x)$ the distance between i and j in configuration x, r_{ij}^0 the corresponding distance in the native state, $\beta = 5\text{Å}^{-1}$ a smoothing parameter and $\lambda = 1.2$ accounting for fluctuations. Note that this metric corresponds to a coarse-grained version of the FNC, whereas the all-atom one in Best et al. (2013) uses cutoff distance 4.5Å and $\lambda = 1.8$.

GDT–TS The Global Distance Test–Total Score (GDT–TS) (Zemla, 2003) quantifies C_α-level similarity to a reference structure across a range of length scales $\{1, 2, 4, 8\}$ Å. For a conformation x with C_α coordinates $\{\mathbf{r}_i(x)\}_{i=1}^L$, rigidly superimposed onto a reference $\{\mathbf{r}_i^*\}_{i=1}^L$,

$$\texttt{GDT-TS}(x) \;=\; \frac{1}{4}\sum_{d\in\mathcal{D}} \frac{1}{L}\left|\left\{\, i : \|\mathbf{r}_i(x) - \mathbf{r}_i^*\| \le d \,\right\}\right|, \qquad \mathcal{D} = \{1, 2, 4, 8\}\,\text{Å}, \tag{B.11}$$

yielding a value in $[0, 1]$ that grows toward 1 as x approaches the reference.

Folding Free Energy For mdCATH, we again follow Kapuśniak et al. (2026) and classify each frame as folded or unfolded using a single threshold on the FNC defined in Eq. (B.10). A threshold $\texttt{FNC}_{1/2}$ is determined once from the 320 K reference MD trajectory: we fit a kernel density estimate to the distribution of FNC values and take $\texttt{FNC}_{1/2}$ to be the location of the deepest minimum within $[0.45, 0.90]$. If the KDE has no minimum in this range, we fall back to $\texttt{FNC}_{1/2} = 0.70$. The same $\texttt{FNC}_{1/2}$ is then applied unchanged to the 450 K MD trajectory and to all generative trajectories for that protein, so that all ensembles are compared against an identical folded/unfolded boundary.

Rather than a hard cutoff, we use a sigmoid soft assignment to obtain the probability of foldedness of each frame as

$$p_{\text{fold}} = \left(1 + \exp\{-2s\,[\texttt{FNC} - \texttt{FNC}_{1/2}]\}\right)^{-1} \tag{B.12}$$

and report its trajectory average $\bar{p}_{\text{fold}}$. The steepness parameter s controls how sharply the assignment transitions across $\texttt{FNC}_{1/2}$: we use $s = 10$, as in Kapuśniak et al. (2026). The folding free energy at temperature T then follows from the two-state population ratio:

$$\Delta G = -k_B T\, \ln\!\left(\frac{\bar{p}_{\text{fold}}}{1 - \bar{p}_{\text{fold}}}\right). \tag{B.13}$$

For the fast folders, we do have sufficiently more data per protein, such that we define the probability to be folded as the stationary distribution value of the native cluster, $\bar{p}_{\text{fold}} = \pi_{\text{native}}$. Otherwise, the evaluation in Eq. (B.13) remains the same. We obtain the underlying stationary distributions by fitting an MSM (lag time 10 ns) once on all available MD trajectories, and on all generated trajectories.

CK The Chapman–Kolmogorov (CK) property requires that the transition matrix of a Markovian system satisfies

$$T(\tau)^k = T(k\tau). \tag{B.14}$$

Following Prinz et al. (2011), we test this by coarse-graining a 10-state MSM into 3 macrostates via PCCA+ (Röblitz & Weber, 2013) and, for each macrostate A, comparing the probability to remain in the macrostate, $A \to A$, at lag $k\tau$ predicted by $T(\tau)^k$ with the empirical estimate from MD counts.

For convenience, we restate important equations from Prinz et al. (2011). For a Markov Model transition matrix $T(\tau)$, let $\boldsymbol{\pi}$ be its stationary distribution. Its restriction to macro state A is defined as

$$w_i^A := \begin{cases} \frac{\pi_i}{\sum_{j\in A}\pi_j} & i \in A \\ 0 & i \notin A \end{cases}. \tag{B.15}$$

The probability to remain in A in time step $k\tau$ estimated from the reference MD trajectory reads

$$p_{MD}(A, A; k\tau) = \sum_{i\in A} w_i^A\, p_{MD}(i, A; k\tau) \tag{B.16}$$

where the estimated probability to transition from frame i into state A is

$$p_{MD}(i, A; k\tau) = \frac{\sum_{j\in A} c_{ij}^{obs}(k\tau)}{\sum_{j=1}^{n} c_{ij}^{obs}(k\tau)} \tag{B.17}$$

with transition counts $c_{ij}^{obs}(k\tau)$ from state i to state j in time $k\tau$. The corresponding probability according to the MSM would be

$$p_{MSM}(A, A; k\tau) = \sum_{i\in A} (w_i^A)^\top T^k(\tau) \tag{B.18}$$

For the actual computation of the RMSE in Table 4, we compute the root mean-squared error of the probability estimate per replica, averaged over all macro states $A \in \{1, 2, 3\}$ and $k \in \{1, 2\}$

$$\texttt{CK RMSE} = \sqrt{\frac{1}{\#\text{ macrostates } \#\text{k}} \sum_{A,k} (p_{MD}(A, A; k\tau) - p_{MSM}(A, A; k\tau))^2} \tag{B.19}$$

For that domain, the result is then averaged over all 5 replicas.

As can be seen from Eq. (B.17), estimating the required probability to remain in state A within time $k\tau$, $p_{MD}(A, A; k\tau)$, from a single held-out replica yields noisy estimates for small c_{ij}^{obs} that become ill-defined whenever c_{ij}^{obs} becomes 0. That is, if A is not visited frequently or at all within the first $T - k\tau$ frames, with T the length of the replica trajectory. This can happen easily for a single held-out short 500 frame trajectory.

C Implementation Details

C.1 Vectorizing persistence diagrams

In order to use persistence diagrams as inputs to standard machine learning models one first needs to represent them as vectors. To this end, we employ the vectorization method from Hofer et al. (2019) based on exponential structure elements. Given a (center) location parameter $\mu \in \mathbb{R}^2$ and a sharpness parameter $\sigma \in \mathbb{R}^2$, Hofer et al. (2019) define

$$t_{\mu,\sigma} : \mathbb{R}^2 \to \mathbb{R} \ , \quad x \mapsto \exp\left(- \sum_{i \in \{1,2\}} \sigma_i^2 (\mu_i - x_i)^2 \right) \tag{C.20}$$

This map is applied to persistence diagrams, or rather all points (b, d) in a persistence diagram with finite death-time $d < \infty$, i.e., a multiset of points in the plane $\mathbb{R}^2$ with $d > b$, after transforming them to (birth,lifetime)-coordinates $(b, d - b)$ and the coordinate stretch

$$\tau_\nu : \mathbb{R} \times (0, \infty) \to \mathbb{R}^2 \ , \quad (b, d-b) \mapsto \begin{cases} (b, d-b) & d - b \geq \nu \\ (b, 2\nu - \frac{\nu^2}{d-b}) & d - b < \nu \end{cases} . \tag{C.21}$$

The map $(b, d) \mapsto t_{\mu,\sigma} \circ \tau_\nu(b, d - b)$ is then called an *exponential structure element*, and as τ_ν stretches lifetime coordinates lower than ν towards $-\infty$, its composition with $t_{\mu,\sigma}$ maps those points close to zero. Summation over all points in the persistence diagram produces a map to $\mathbb{R}$ via

$$\text{dgm} \mapsto \sum_{(b,d) \in \text{dgm}} t_{\mu,\sigma} \circ \tau_\nu(b, d - b) \ , \tag{C.22}$$

and by construction, the contribution of points tends to zero as the lifetime decreases to zero. Repeating this for multiple centers $\mu_1, \ldots, \mu_d$ and sharpness parameters $\sigma_1, \ldots, \sigma_d$, i.e., using d different structure elements, and concatenating the output then gives a vectorization map to $\mathbb{R}^d$ parametrized by $2d + 2d + 1 = 4d + 1$ parameters.

While in principle, these parameters can be learned within a neural network, see Hofer et al. (2019), we directly fit them to the data so that the vectorized persistence diagrams are fixed features for learning. To do so, we use a strategy as in ATOL (Royer et al., 2021). We first stack all persistence diagrams in the training split of a dataset, producing a multiset of points in $\mathbb{R}^2$ of large cardinality and apply the (birth,lifetime)-transform. We then fit the centers μ_j using k-means with k-means++ initialization (Arthur & Vassilvitskii, 2007). As the respective sharpness parameter σ_j we use $\sigma_j = (\frac{2}{\delta}_j, \frac{2}{\delta}_j)$ with $\delta_j = \min_{k \neq j} \|\mu_j - \mu_k\|_2$. This leaves two hyperparameters that need to be selected: the number d of centers, and the quantile q (on the training split) that is used to determine the stretch parameter ν. In all applications we set $d = 128$ and $q = 0.01$.

C.2 Setup for Representation Quality Probes

The inputs to the probing models are vectorized persistence diagrams of dimension 384, i.e., 128 dimensions per H_0, H_1 and H_2 diagram, computed with the stretch quantile $q = 0.01$ for every

simplicial complex (Appendix C.1). Each probe is either a linear map or an MLP with ELU activations and hidden dimensions (128, 64). For CATH classification the probe maps to four outputs; for observable regression it maps to a scalar, and one probe is trained per observable. Features are standardized with statistics of the training split. The VAMPNet uses an MLP with hidden dimensions (256, 128, 64, 32, 16) as backbone and learns 7-dimensional kinetic embeddings. For its domain-wise loss, a minibatch is constructed from 4 domains, 4 replicas per domain, and 400 time-lagged pairs (all sampled uniformly at random).

CATH classification The frame vectors of each replica are averaged over time, so that each replica contributes one sample (about 21k / 2.7k / 2.4k samples for training / validation / testing). Probes minimize a class-balanced cross-entropy with batch size 128 for at most 100 epochs, with early stopping on the validation balanced accuracy (patience 20) and restoration of the best weights. We report the balanced accuracy on the test split, averaged over three seeds.

Observable regression Every frame of all five 450 K replicas is used, i.e., about 9.9M / 1.2M / 1.1M frames for training / validation / testing. Probes minimize the mean squared error on the observable in its physical units, with batch size 16,384. Training runs for at most 200 epochs and stops early once the validation R^2 has not improved by at least 10^{-3} for 20 consecutive epochs; the weights of the epoch with the best validation R^2 are restored. All runs stopped early, after 21–67 epochs. We report the test R^2 over all test frames, averaged over three seeds.

The setup for the $((\phi, \psi) + \mathrm{C}_\alpha)$-baseline is as follows. For each MD simulation of a domain, the 7 per-residue features, i.e., angles and C_α coordinates are stacked along the residues and time steps, resulting in a tensor of shape $(\#\text{timesteps}, 7, \#\text{residues})$. These are then input to a 1D convolutional network, where the time steps serve as a batch and the 7 per-residues features as input channels; the convolution kernel operates along the residue axis. The convolutional network consists of three 1D convolutional layers with kernel size 7 and stride 1, ReLU activations, and output channels of size 16, 64 and 256. This is followed by an average pooling layer operating along the residue axis, producing embeddings of shape $(\#\text{timesteps}, 256)$ and a final linear layer projecting the embeddings either to $\mathbb{R}$ for regression or to $\mathbb{R}^4$ for classification. Gradients are accumulated over multiple domains to mimic batching.

Optimization All probes are trained using ADAM with a cosine annealing schedule stepped per epoch, an initial learning rate of 10^{-3} and weight decay 10^{-3}. The VAMPNet is trained using ADAM with an initial learning rate of $5 \cdot 10^{-4}$ and weight decay 10^{-3} for at most 200 epochs of 50 minibatches. Its learning rate is halved whenever the `vamp-2` score on the validation domains has not improved by a relative 10^{-3} for 5 epochs, down to 10^{-6}; training stops after 5 epochs without improvement, and the weights of the best epoch are restored. Early stopping uses the `vamp-2` score of the continuous embedding, not the MSM measures we report. Gradients are clipped at a global norm of 5, and the loss adds an L_2 penalty of 10^{-3} times the mean squared embedding. The end-to-end trained graph encoders deviate from this as described in Appendix C.3.

C.3 ML Baselines

We compare the persistence-based representations against two kinds of learned baselines. *Learned end-to-end baseline models* (`EGNN`, `GotenNet`) are trained jointly with the probe head on each task. *Learned representations baselines* (`ESM2`, `ESM-IF1` and their fusion) provide a fixed vector per domain or per frame, computed once, on top of which only a probe head is trained, exactly as for the vectorized persistence diagrams.

C.3.1 Learned end-to-end baseline models

Input graphs Both graph encoders operate on the same residue-level graph, with one node per residue, positioned at its C_α atom. As node features the residue identity is mapped to 20 classes. Edges are determined by implementing a radius graph with cutoff 1.0 nm (10 Å) and at most 32 neighbors per node, without self-loops. The original models were designed for small molecules (`EGNN` on fully connected QM9 graphs), which is infeasible for domains of 50–500 residues.

For CATH classification each domain contributes the first frame of each of its five replicas ($\sim$ 21k training, 2.6k validation and 2.4 k testing frames). For observable regression we use every 50th frame of every replica ($\sim$ 199k training, 25k validation and 22.7k testing frames); end-to-end training on every frame was outside of our compute budget.

EGNN We use the reference implementation of Satorras et al. (2021) with the configuration of the original QM9 experiments: 7 layers, hidden and output width 128, attention and residual connections enabled, no coordinate normalization and no $\tanh$ on the coordinate update. The node-input is the one-hot residue identity; the only edge input is the squared distance. We adapt the model to the larger sized graphs of `mdCATH` domains in three ways: the radius graph above replaces the fully connected graph; connectivity is computed once from the input coordinates and not rebuilt as the layers update the coordinates; and the updated coordinates are discarded, only the invariant node features h_i being pooled into one vector per graph.

GotenNet We use `GotenNetWrapper` from the authors' package (Aykent & Xia, 2025) (version 1.1.3), with scalar width 128, 8 interaction layers, $l_{\max} = 1$, 32 radial basis functions, a cosine cutoff at 1.0 nm and at most 32 neighbors per node. Residue identities enter through `GotenNet`'s atom-type embedding. The radius graph is built inside `GotenNet` from the C_α coordinates, and the scalar node features after the last layer are pooled.

Training of the graph encoders The encoders are mean-pooled over nodes and read out by the same two-layer MLP head (128, 64; ELU). Graphs are packed into batches of at most 8,000 nodes to bound memory across domain sizes. Otherwise, training follows the probes (Appendix C.2), with three deviations: a weight decay of 10^{-4} instead of 10^{-3} (at 10^{-3} both encoders stall, e.g., at a validation R^2 of 0.22–0.52 on `RG`, whereas 10^{-4} reaches 0.83 after one epoch), gradient clipping at a global norm of 1.0 as `EGNN` would otherwise diverge, and a budget of at most 200 epochs also for classification.

On one NVIDIA H100 GPU, one regression epoch takes about 7.5 min for `GotenNet` and 1.8 min for `EGNN`.

C.3.2 Learned Representations

ESM2 We use the pretrained `ESM2` models (Lin et al., 2023) `esm2_t6_8M_UR50D` (6 layers, 320 dimensions), `esm2_t12_35M_UR50D` (12 layers, 480 dimensions), `esm2_t30_150M_UR50D` (30 layers, 640 dimensions) and `esm2_t33_650M_UR50D` (33 layers, 1,280 dimensions). Each domain's residue sequence is tokenized with the model's alphabet; the final-layer residue representations are averaged over residues, yielding one vector per domain. As the sequence is identical in every frame, `ESM2` enters CATH classification as a per-domain rather than a per-replica vector and observable regression only through the fusion model below.

ESM-IF1 We use the encoder of the inverse folding model `esm_if1_gvp4_t16_142M_UR50` (Hsu et al., 2022). Its input is the backbone (N, C_α, C) of a single frame with its 512-dimensional per-residue outputs averaged over residues to give one vector per frame. For CATH classification, as for PH vectorizations, each replica is represented by the trajectory-averaged vector.

Fusion (`ESM2_35m`, `ESM-IF1`) The domain's `ESM2_35m` vector (480 dimensions) is repeated for every frame and concatenated with that frame's `ESM-IF1` vector (512 dimensions), giving 992 dimensions of which only the `ESM-IF1` part varies within the trajectory. `ESM2_35m` was chosen as the sequence representation since it is performing best in the ESM2 family on CATH classification (see Table D.8).

C.4 ESM Train-Test overlap

The ESM2 family was trained on $\sim$65 million unique sequences from UniRef50 (Suzek et al., 2015), v2021_04, which makes an overlap with our test set likely but cannot be verified as training IDs are not published.

ESM-IF1 (`esm_if1_gvp4_t16_142M_UR50`) was pretrained on $\sim$16k structures from CATH v4.3 and 12 million AlphaFold2 predicted ones from UniRef50 v2021_04. An overlap with the latter 12

million structures is possible, but can again not be verified as these training IDs are not public. The PDB ids of the CATH portion can be retrieved at https://dl.fbaipublicfiles.com/fair-esm/data/cath4.3_topologysplit_202206/splits.json. 43.9% (214) of our 488 test domains occur exactly in ESM-IF1's CATH training set with 207 as an exact match and 7 as re-depositions under another PDB id. In every case, the mdCATH sequence is an exact segment of the ESM-IF1 training chain. For these, crystal-structure alignment confirms structural identity (median qTM = 1.0). Of the 274 remaining test domains, a further 47 reach qTM ≥ 0.9 against an ESM-IF1 training structure, versus 3 when the same domains are measured against our own training split. In total, 53.5% of the test set is either identical or near-identical to something ESM-IF1 trained on. For context, our own train/test split has no detectable sequence homology (0/488 domains at $\geq$20% identity, as mdCATH is drawn from CATH's S20 non-redundant set). This entails substantial pretraining–evaluation overlap. We report the ESM-IF1 and (ESM2, ESM-IF1) results for completeness, but they should be digested with this caveat in mind.

C.5 VAMPNet

Formally, given coordinate trajectories $\{\mathbf{x}_t\}_t$ (e.g., vectorized persistence diagrams) of different protein domains d drawn from some distribution p, we want to optimize $f_{\boldsymbol{\theta}}$ such that the induced latent trajectories $\{f_{\boldsymbol{\theta}}(\mathbf{x}_t)\}_t$ maximize the expected vamp-2 score (for some fixed lag time τ). That is, we search for

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{d\sim p} \underbrace{\left[\left\| \left(\mathbf{C}_{00}^{(d)}\right)^{-1/2} \mathbf{C}_{01}^{(d)} \left(\mathbf{C}_{11}^{(d)}\right)^{-1/2} \right\|_F^2 \right]}_{\text{vamp-2 score}}, \tag{C.23}$$

where $\mathbf{C}_{00}^{(d)}$,$\mathbf{C}_{01}^{(d)}$ and $\mathbf{C}_{11}^{(d)}$ denote the domain-specific covariance matrices constructed from different combinations of time-lagged pairs $(\mathbf{z}_t, \mathbf{z}_{t+\tau}) = (f_{\boldsymbol{\theta}}(\mathbf{x}_t), f_{\boldsymbol{\theta}}(\mathbf{x}_{t+\tau}))$ of VAMPNet representations.

In practice, we rely on stochastic optimization of an empirical estimate of Eq. (C.23) based on multiple simulation replicas $\mathcal{R}_d$ of each domain $d \in \mathcal{D}$. Specifically, we first sample a mini-batch of domains $\mathcal{D}_b \subset \mathcal{D}$; then, for each sampled domain $d \in \mathcal{D}_b$, we select N_R replicas $\mathcal{R}_{d,b} \subseteq \mathcal{R}_d$ at random and, within each replica, a set of N_T random time points $\mathcal{T}_{d,r}$ (without replacement). Upon collecting all time-lagged pairs and letting $\mathbf{z}_{r,t} = f_{\boldsymbol{\theta}}(\mathbf{x}_{r,t})$ denote the VAMPNet representation of $\mathbf{x}_{r,t}$, we construct the empirical per-domain covariance matrices

$$\widehat{\mathbf{C}}_{00}^{(d)} = \frac{1}{N_d} \sum_{\substack{r\in\mathcal{R}_{d,b}\\ t\in\mathcal{T}_{d,r}}} \mathbf{z}_{r,t}\mathbf{z}_{r,t}^\top, \; \widehat{\mathbf{C}}_{01}^{(d)} = \frac{1}{N_d} \sum_{\substack{r\in\mathcal{R}_{d,b}\\ t\in\mathcal{T}_{d,r}}} \mathbf{z}_{r,t}\mathbf{z}_{r,t+\tau}^\top, \; \widehat{\mathbf{C}}_{11}^{(d)} = \frac{1}{N_d} \sum_{\substack{r\in\mathcal{R}_{d,b}\\ t\in\mathcal{T}_{d,r}}} \mathbf{z}_{r,t+\tau}\mathbf{z}_{r,t+\tau}^\top,$$

with $N_d := \sum_r |\mathcal{T}_{d,r}|$ and compute the empirical vamp-2 loss accordingly. Finally, using the trained network $f_{\boldsymbol{\theta}}$, we map all trajectory frames to the latent space and, for each domain, discretize the representations of each replica into k states using k-means++. The resulting discrete trajectories are then used to estimate MSM transition matrices via maximum likelihood at lag time τ.

C.6 MarS-FM

Training Training follows the setup of Kapuśniak et al. (2026) exactly. We use a transformer comprising a stack of Invariant Point Attention layers (Jumper et al., 2021) (4 heads, 32-dimensional head size, 8 query-key and 8 value points) followed by a stack of 5 modified DiT blocks (Peebles & Xie, 2023; Jing et al., 2024) with 384-dimensional token embeddings and 16-head multi-head attention.

Optimization uses ADAM with a constant learning rate of 10^{-4} and an exponential moving average with decay 0.999. Training pairs are drawn from the per-protein MSM with 2 clusters and 12 sequences per cluster. We train for 1,000 epochs at batch size 8. For generation, the learned velocity field is integrated with the adaptive dopri5 solver from torchdiffeq (Chen et al., 2018) using absolute tolerance 10^{-6}, relative tolerance 10^{-3} evaluated at 50 equally spaced time points.

Generation We follow the hierarchical tree-sampling scheme of Kapuśniak et al. (2026). Starting from the initial frame, we first generate 200 conformations in parallel. In a second pass, we generate

300 additional conformations, each conditioned on one of the first-step samples, yielding 500 samples in total. The first-layer branching factor of 200 corresponds to the default identified as near-optimal in Kapuśniak et al. (2026).

C.7 Fast folders

Finetuning To adapt to the dynamic regime of the fast folders (Appendix A.2) we finetune our network on the 348 K simulations of `mdCATH` by the following strategy: The default `MarS-FM` training batch consists of 2 clusters with 12 samples each for 8 protein domains. One of the two sampled clusters always corresponds to the initial frame cluster, usually a comparatively compact conformation, while the second is sampled randomly. Thereby, the model obtains a strong learning signal for generating *from* compact structures toward extended structures. The fast folding dataset, however, specifically seeks out folding events. To match this dynamic regime, we flip the role of the starting frame cluster, sampling clusters that transition *into* it.

To adapt to the faster dynamics, we reduce lag time τ with respect to which MSMs are fitted (for sampling training pairs), to 10 ns. Furthermore, VAMPNet training on the 348 K dataset (persistence diagram) vectorizations of `mdCATH` is also run at a lag time of 10 ns.

We finetune for 300 epochs using ADAM with a one-cycle schedule (`OneCycleLR` in PyTorch (Paszke et al., 2019)), i.e., a linear warm-up to a peak learning rate of 10^{-5} over the first 10% of training steps, followed by cosine annealing. The `MarS-FM` baseline is finetuned with the identical protocol and differs only in the features from which its MSMs are estimated.

Generation For generation, we sample each of the 32 clusters 32 times uniformly for source conformations as input for our generative model, generating 500 samples in a tree sampling manner as prescribed by Kapuśniak et al. (2026): We generate 200 conformations in the first step, and 300 more in the second with the initial 200 as input. This yields per protein $32 \times 32 \times 500 = 512{,}000$ generated conformations.

C.8 Computing infrastructure

All experiments were run on an Ubuntu Linux system (24.04), running kernel 6.8.0-83-generic, with two AMD EPYC 7313 16-Core Processors (64 threads), 512 GB of main memory, and two NVIDIA H100 NVL GPUs.

D Supplementary results

D.1 Probing Representation Quality

In Table D.8, we provide CATH classification results for additional sequence-based models, `ESM2_8m`, `ESM2_150m` and `ESM2_650m`, as well as the fusion model, i.e. concatenated representations (`ESM2_35m`, `ESM-IF1`).

Table D.9 lists the regression results of Table 3 with standard deviations over three seeds. All models are evaluated on every test frame, except `EGNN` and `GotenNet`, which are trained and evaluated on every 50th frame because of their training cost (Appendix C.3).

D.2 Dynamics

D.2.1 Validity of mdCATH MSM Evaluation

When applying VAMP naively to short trajectories, as they are present in `mdCATH`, one may run into problems overfitting. This is one reason for learning a common feature map across domains, increasing and diversifying the available data on which the embedding is estimated from $\sim 2.3\mu$s per domain to ~ 10ms using the whole training set. In general, to deal with finite data we follow McGibbon & Pande (2015), who argue that in the case of the cross-validation mean test set generalized matrix Rayleigh quotient (GMRQ) is the proper metric for model selection. The GMRQ $\mathcal{R}$ is an equivalent formulation of the VAMP score.

Table D.8: Additional Per-replica classification of CATH class. Balanced accuracy (mean ± std over 3 runs). **Best**/second-best PH-based result.

	Linear	MLP
mFlood PH	79.1 ± 0.7	**81.2** ± 2.1
Flood PH	78.4 ± 2.0	80.0 ± 1.6
Alpha PH (C_α)	78.4 ± 1.0	78.2 ± 5.8
Alpha PH	71.3 ± 3.4	74.5 ± 3.1
Rips PH (C_α)	**79.5** ± 0.8	77.4 ± 5.3
(RG, SSF, FNC, RMSD)	54.3 ± 0.6	60.5 ± 0.4
(ϕ, ψ) + C_α coords.	–	70.8 ± 0.1
ESM2_8m	76.9 ± 1.3	75.5 ± 2.6
ESM2_35m	79.5 ± 0.2	77.3 ± 0.6
ESM2_150m	75.4 ± 1.5	74.0 ± 0.7
ESM2_650m	73.5 ± 0.3	76.1 ± 1.4
ESM-IF1	82.7 ± 1.8	78.7 ± 3.6
(ESM2_35m, ESM-IF1)	81.5 ± 1.6	78.8 ± 1.8
EGNN	–	71.8 ± 5.1
GotenNet	–	75.5 ± 7.9

Table D.9: Regression of physical observables per time point across testing domains of mdCATH (Table 3). Coefficient of determination R^2 ↑ (mean ± std over 3 seeds). **Best**/second-best PH-based result.

	FNC	RG	RMSD	SSF
Linear				
mFlood PH	**0.691** ± .002	0.728 ± .001	0.683 ± .001	0.652 ± .003
Flood PH	0.686 ± .003	0.719 ± .002	0.675 ± .001	**0.658** ± .004
Alpha PH	0.668 ± .000	**0.831** ± .001	**0.708** ± .002	0.623 ± .000
Alpha PH (C_α)	0.669 ± .002	0.675 ± .001	0.656 ± .000	0.650 ± .003
Rips PH (C_α)	0.630 ± .001	0.559 ± .001	0.597 ± .001	0.599 ± .002
ESM-IF1	0.918 ± .001	0.904 ± .001	0.822 ± .000	0.933 ± .001
(ESM2_35m, ESM-IF1)	0.929 ± .001	0.905 ± .000	0.829 ± .001	0.932 ± .000
MLP				
mFlood PH	0.860 ± .002	0.861 ± .000	0.781 ± .001	0.842 ± .000
Flood PH	**0.863** ± .001	0.864 ± .000	0.782 ± .000	0.853 ± .001
Alpha PH	0.858 ± .001	**0.921** ± .000	**0.798** ± .000	0.815 ± .000
Alpha PH (C_α)	0.859 ± .001	0.849 ± .001	0.772 ± .000	**0.854** ± .000
Rips PH (C_α)	0.840 ± .000	0.772 ± .000	0.732 ± .002	0.828 ± .001
(ϕ, ψ) + C_α coords.	0.762 ± .015	0.332 ± .018	0.578 ± .003	0.841 ± .003
ESM-IF1	0.929 ± .000	0.946 ± .000	0.849 ± .001	0.934 ± .000
(ESM2_35m, ESM-IF1)	0.942 ± .000	0.944 ± .001	0.853 ± .005	0.934 ± .000
EGNN	0.878 ± .003	0.955 ± .011	0.799 ± .026	0.863 ± .005
GotenNet	0.879 ± .004	0.953 ± .025	0.808 ± .016	0.868 ± .002

Equation 24 in McGibbon & Pande (2015) proposes to calculate the mean over the training folds to gauge overfitting by the difference between train and test $\mathcal{R}$. We compare this difference for mFlood PH and the MarS-FM combination of (RG, SSF) in Table D.10, showing that mFlood PH does not display a gap between train and test data.

Table D.10: Test and train VAMP2. Large differences would indicate overfitting on the training trajectories.

Feature	test VAMP2	train VAMP2
mFlood PH	1.503 ± 0.010	1.503 ± 0.001
(RG, SSF)	1.513 ± 0.004	1.559 ± 0.001

It is a major strength of MSMs that the underlying simulations do *not* need to sample from equilibrium distributions (Nüske et al., 2017; Schütte et al., 1999). Nüske et al. (2017) consider trajectories which neither start nor end in the global equilibrium as the simulation time is too short, and state that short trajectories starting from local equilibria (mdCATH equilibrates every replica for 20 ns) suffice, if they last longer than the system's largest timescale t_2. We estimate t_2 implied by MSMs fitted on mFlood PH and (RG, SSF) features, resp., and report the t_2-mean per class $\pm$ its standard error in Table D.11.

Table D.11: Largest implied timescale per CATH class.

CATH class	t_2(mFlood PH)	t_2(RG, SSF)
mainly-alpha	67.2 ± 5.6	84.1 ± 9.1
mainly-beta	153.8 ± 15.3	165.3 ± 14.4
alpha-beta	122.6 ± 9.8	133.5 ± 9.7
few sec. struct.	34.7 ± 4.9	50.7 ± 4.9

Notably, the estimates per CATH class follow the trends reported by Mirarchi et al. (2024) when introducing mdCATH. For the 450 K simulations, the significant process is the unfolding of the protein domain (see Fig. D.4 for an example). According to Mirarchi et al. (2024), unfolding occurs for α rich structures abruptly within 100 ns, for α-β and β structures somewhat more slowly (cf. Mirarchi et al. (2024), Figure S1). Thus, according to Nüske et al. (2017), mdCATH trajectories suffice to estimate valid MSMs.

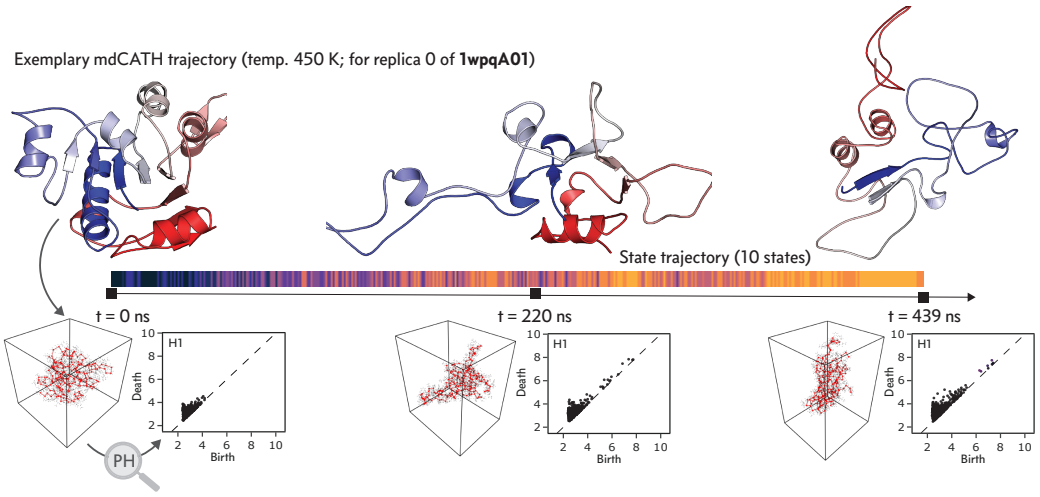


Figure D.4: Unfolding at 450 K, shown for replica 0 of the mdCATH domain **1wpqA01**. **Top**: conformations at three time points and the state trajectory obtained from the 10-state discretization of the shared mFlood PH VAMPNet embedding. **Bottom**: point clouds (C_α atoms in red) and H_1 persistence diagrams of the masked Flood complex at the same time points.

D.2.2 MSM Evaluation

TICA, as a special linear case of VAMP, allows for dimensionality reduction onto the components of maximal autocorrelation of a time series. It still needs some input features, and it needs to be applied to (the MD trajectories of) each domain separately. Importantly, it requires inverting the instantaneous covariance matrix; a matrix of size $(\text{num}_{\text{features}}, \text{num}_{\text{features}})$ and rank $\leq \text{num}_{\text{observed timepoints}}$. This renders the use of TICA challenging on mdCATH, where one typically has only 2k observations per domain, and the domains consist of up to 500 residues. In particular, the features used in Scherer et al. (2019) (for a different kind of data) are ill-suited. The distance-based features are quadratic in $\text{num}_{\text{residues}}$, and even the per-residue features (C_α, torsion), except SASA (which only uses 1 feature per residue), are already so many that TICA overfits, especially on large domains.

In Table D.12 we report MSM quality metrics on the validation domains based on TICA embeddings of three features: C_α coordinates, Torsion (sine-cosine pairs of all backbone and sidechain angles)

and $(\phi, \psi) + C_\alpha$ (again sine-cosine pairs of the angles). In correspondence with our VAMPNet implementation, we project to the 7 first TICA components. For comparison we also list the results of VAMPNet embeddings of mFlood.

Table D.12: MSM quality for physical featurizations, scored with the pipeline of Table 4 (per domain, leave-one-replica-out at lag $\tau = 50$, 10 micro states, TICA onto 7 components). `mFlood PH` is listed for comparison.

feature	vamp-2 ↑	CK RMSE ↓	π-JS ↓
C_α coordinates	1.392 ± .002	0.040 ± .000	0.254 ± .001
Torsion	1.264 ± .002	0.076 ± .000	0.139 ± .001
$(\phi, \psi) + C_\alpha$	1.335 ± .003	0.063 ± .000	0.185 ± .001
SASA	1.439 ± .002	0.034 ± .000	0.258 ± .001
`mFlood PH`	1.503 ± .010	0.034 ± .001	0.120 ± .004

When using distance based features (see Table D.12), the covariance matrices become singular on domains with more than 66 residues. On the remaining ∼13% of domains, performance is poor and no real dynamic information is extracted.

Table D.13: MSM quality for TICA projections of distance-based featurizations. These are computable only for the smallest domains, so *% ok* gives the share of the 532 validation domains that could be scored. Mean over those domains.

feature	% ok	vamp-2 ↑	CK RMSE ↓	π-JS ↓
resdist	12.8	1.227 ± .003	0.157 ± .002	0.046 ± .002
d^{-1}	12.8	1.225 ± .002	0.159 ± .001	0.057 ± .001
d^{-2}	12.8	1.222 ± .003	0.157 ± .001	0.062 ± .001
$\log d$	12.8	1.225 ± .001	0.159 ± .002	0.050 ± .001
e^{-d}	12.8	1.216 ± .001	0.158 ± .001	0.079 ± .001
h_4	12.8	1.226 ± .002	0.151 ± .002	0.080 ± .003
h_5	12.8	1.228 ± .003	0.151 ± .002	0.075 ± .002
h_6	12.8	1.222 ± .003	0.156 ± .002	0.065 ± .001
h_8	12.8	1.223 ± .002	0.160 ± .002	0.057 ± .001
h_{10}	12.8	1.221 ± .001	0.156 ± .002	0.049 ± .001
(e^{-d}, torsions)	4.7	1.207 ± .003	0.196 ± .003	0.047 ± .001

In Table D.13, d refers to the minimal distance between heavy atom pairs of different residues, excluding neighbors $i + 1$ and $i + 2$. The contact features are defined as

$$h_c(d) = \begin{cases} 1, & \text{if } d < c \\ 0, & \text{if } d \geq c \end{cases} \tag{D.24}$$

for $c \in \{4, 5, 6, 8, 10\}$ in units of Å.

D.2.3 MDCATH

In Table D.14 we extend Table 5 and report all ensemble metrics as were reported in Kapuśniak et al. (2026, Table 2 & Table 7). In Table D.15 we supply additional validity metrics (as in Kapuśniak et al. (2026, Table 8)) regarding the ability of the generated ensemble to recover (1) the reference distribution of native contacts (FNC JSD), (2) C_α-level similarity and (3) backbone geometry (GDT-TS JSD, Bond and Angle RMSZ). We report `MarS-FM` results from using the released checkpoint, as †`MarS-FM` our own re-training (using the same protocol and hyperparameters) and as ‡`MarS-FM` the values stated in Kapuśniak et al. (2026).

To establish statistical significance for domain-wise metrics[2], we compare the original `MarS-FM` with `MarS-FM-mFlood` using two-sided paired t-tests, followed by Benjamini–Hochberg correction across all 11 per-domain metrics (q=0.05) in Table D.16. The `MarS-FM-mFlood` variant performs

[2]For metrics comparing the whole generated ensemble (Pairwise RMSD r, Global RMSF r, ΔG r) no individual domain values exist, such that these cannot be compared in this manner.

Table D.14: Complete generative evaluation metrics. `MD oracle` represents the performance of *one* against the distribution of all 5 replicas, and, as baseline, `MarS-FM-random`, lists results of a `MarS-FM` model when the guiding MSMs are fitted to 2D Gaussian random features across training trajectories. For each model, we report mean ± 1 std. dev. over 5 evaluation runs.

	Pairwise RMSD r ↑	Global RMSF r ↑	Per-target RMSF r ↑	RG KL ↓	RG JSD ↓
`MD oracle`	0.82 ± .014	0.82 ± .012	0.90 ± .005	0.58 ± .020	0.07 ± .001
`MarS-FM-random`	0.57 ± .004	0.68 ± .003	0.89 ± .001	0.51 ± .002	0.10 ± .000
†`MarS-FM`	0.66 ± .003	0.72 ± .002	0.89 ± .003	0.55 ± .002	0.10 ± .000
‡`MarS-FM`	0.65 ± .004	0.71 ± .003	0.89 ± .001	0.55 ± .002	0.10 ± .001
`MarS-FM`	0.64 ± .002	0.71 ± .002	0.90 ± .002	0.55 ± .002	0.10 ± .000
`MarS-FM-mFlood`	0.68 ± .003	0.74 ± .001	0.90 ± .002	0.58 ± .003	0.10 ± .000
`MarS-FM-Flood`	0.66 ± .002	0.73 ± .001	0.89 ± .002	0.58 ± .002	0.10 ± .000
`MarS-FM-Alpha` (C_α)	0.67 ± .002	0.72 ± .001	0.89 ± .001	0.60 ± .003	0.10 ± .000
`MarS-FM-Alpha`	0.68 ± .004	0.74 ± .002	0.90 ± .001	0.59 ± .002	0.10 ± .000
	SSF KL ↓	**SSF JSD ↓**	**MSM JSD ↓**	**ΔG MAE ↓**	**ΔG r ↑**
`MD oracle`	0.61 ± .048	0.07 ± .003	0.19 ± .006	0.90 ± .046	0.90 ± .002
`MarS-FM-random`	1.52 ± .009	0.18 ± .001	0.16 ± .002	1.36 ± .007	0.86 ± .004
†`MarS-FM`	0.91 ± .006	0.13 ± .000	0.18 ± .001	1.05 ± .005	0.87 ± .000
‡`MarS-FM`	0.93 ± .010	0.13 ± .001	0.19 ± .004	1.05 ± .003	0.87 ± .003
`MarS-FM`	1.05 ± .007	0.14 ± .000	0.18 ± .001	0.99 ± .009	0.89 ± .002
`MarS-FM-mFlood`	0.81 ± .005	0.12 ± .000	0.18 ± .004	0.96 ± .012	0.89 ± .001
`MarS-FM-Flood`	0.81 ± .007	0.12 ± .000	0.18 ± .004	0.97 ± .011	0.88 ± .003
`MarS-FM-Alpha` (C_α)	0.78 ± .006	0.12 ± .000	0.18 ± .002	0.91 ± .004	0.89 ± .002
`MarS-FM-Alpha`	0.85 ± .007	0.12 ± .000	0.18 ± .003	0.92 ± .005	0.89 ± .002

Table D.15: Comparisons on foldedness distributions (FNC JSD), C_α level similarity (GDT-TS JSD) and backbone covalent geometry (Bond / Angle RMSZ).

	FNC JSD	GDT-TS JSD	Bond RMSZ	Angle RMSZ
`MD oracle`	0.13 ± .003	0.11 ± .004	2.23 ± .012	2.19 ± .009
`MarS-FM-random`	0.17 ± .001	0.14 ± .000	6.99 ± .048	3.14 ± .003
†`MarS-FM`	0.14 ± .001	0.11 ± .000	6.95 ± .071	3.12 ± .003
‡`MarS-FM`	0.15 ± .001	0.12 ± .001	12.62 ± .144	3.06 ± .001
`MarS-FM`	0.15 ± .000	0.12 ± .000	7.04 ± .055	3.16 ± .005
`MarS-FM-mFlood`	0.14 ± .001	0.11 ± .000	5.20 ± .018	2.94 ± .003
`MarS-FM-Flood`	0.14 ± .000	0.11 ± .000	5.07 ± .021	2.95 ± .005
`MarS-FM-Alpha` (C_α)	0.14 ± .000	0.11 ± .000	5.26 ± .057	2.90 ± .004
`MarS-FM-Alpha`	0.14 ± .001	0.11 ± .000	6.21 ± .050	3.12 ± .003

significantly better on five metrics, i.e., SSF KL, SSF JSD, bond RMSZ, angle RMSZ, and per-target RMSF correlation, whereas the original `MarS-FM` performs significantly better on RG KL. No statistically significant difference is detected for the remaining five metrics. The *Direction of mean difference* indicates which method yielded the better result in terms of absolute numbers.

Table D.16: Significance assessment of generated ensemble statistics between `MarS-FM` and `MarS-FM-mFlood`.

Metric	Direction of mean difference	FDR-adjusted p-value	Significant?
RG KL	`MarS-FM`	0.0002	Yes
RG JSD	`MarS-FM`	0.8607	No
SSF KL	`MarS-FM-mFlood`	1e-10	Yes
SSF JSD	`MarS-FM-mFlood`	1e-10	Yes
MSM JSD	`MarS-FM-mFlood`	0.8607	No
dG MAE	`MarS-FM-mFlood`	0.1577	No
FNC JSD	`MarS-FM-mFlood`	0.2959	No
GDT-TS JSD	`MarS-FM-mFlood`	0.9428	No
Bond RMSZ	`MarS-FM-mFlood`	1e-10	Yes
Angle RMSZ	`MarS-FM-mFlood`	1e-10	Yes
Per-target RMSF r	`MarS-FM-mFlood`	0.0435	Yes

D.3 Consistency across evaluations

Table D.17 summarizes how the PH-based representations rank against each other across the evaluations of the main text, using all generative metrics of Appendix D.2.3. For each metric, we rank them at the precision reported in the respective table, with ties sharing the better rank, and average the ranks per evaluation.

Table D.17: Mean rank of the PH-based representations per evaluation (1 = best): CATH classification (2 metrics, Table 2), observable regression (8 metrics, Table 3), MSM quality (3 metrics, Table 4) and generation (14 metrics, Tables D.14 and D.15; `Rips PH` (C_α) not evaluated). *Overall* and *Worst* give the mean and the worst rank over all metrics; *Overall w/o generation* averages over the 13 classification, regression and MSM metrics, on which all five representations are evaluated.

	Classification	Regression	MSM quality	Generation	Overall	Overall w/o generation	Worst
`mFlood PH`	**1.50**	2.25	**1.67**	**1.36**	**1.67**	**2.00**	**3**
`Flood PH`	2.50	**2.00**	3.33	2.14	2.26	2.38	4
`Alpha PH`	5.00	2.63	2.33	1.86	2.37	2.92	5
`Alpha PH` (C_α)	3.00	3.25	3.33	1.86	2.52	3.23	5
`Rips PH` (C_α)	2.50	4.88	4.00	–	4.31	4.31	5

D.4 Fast Folders

Table D.18 compares native-state conformations generated by `MarS-FM-mFlood` with finetuning to those of the MD reference, measuring RMSD, FNC and GDT-TS with respect to the experimental structure, with the native state defined as in Appendix A.2. For proteins with a tight native basin in the reference (reference RMSD below 2 Å, i.e., Chignolin, Trp-Cage, WW-Domain, Villin, NTL9 and Protein G), generated native conformations deviate from the crystal structure by 0.6 to 1.6 Å more than the reference. For proteins whose native basin is already broad in the reference, this gap narrows and nearly vanishes for BBA, BBL and Homeodomain, indicating that larger deviations there reflect the breadth of the native ensemble rather than a failure of the model.

Figure D.5 illustrates the effect of finetuning for Protein G. Without finetuning, most generated probability mass falls into a single basin distant from the crystal structure, and the native region is only sparsely populated. After finetuning, the generated landscape recovers the overall shape of the MD reference, including the basins near the native state, and the generated frame closest to the crystal structure improves from 1.02 Å to 0.89 Å, compared to 0.55 Å in the reference.

Table D.19 shows that covalent geometry is of comparable quality across all proteins. Bond RMSZ of generated conformations ranges from 2.6 (Homeodomain) to 3.7 (λ-repressor) and angle RMSZ from 2.1 to 2.7, compared to 1.9 to 2.0 for the MD reference. The mean is raised by a small fraction of strongly distorted frames: the median bond RMSZ lies between 2.3 and 2.8, and 95.9% (λ-repressor)

Table D.18: Mean RMSD, FNC, and GDT-TS for generated and MD reference trajectories (fast folders) all between structures clustered into the native state with respect to the experimental crystal structure.

	Generated			Reference		
Protein	Mean RMSD (Å)	FNC	GDT-TS	Mean RMSD (Å)	FNC	GDT-TS
Chignolin	1.58 ± 0.53	0.70 ± 0.23	0.80 ± 0.10	0.95 ± 0.33	0.91 ± 0.10	0.92 ± 0.07
Trp-Cage	2.24 ± 0.72	0.85 ± 0.10	0.67 ± 0.10	1.57 ± 0.37	0.91 ± 0.05	0.76 ± 0.07
BBA	4.08 ± 1.27	0.76 ± 0.11	0.52 ± 0.11	3.60 ± 1.27	0.74 ± 0.09	0.58 ± 0.11
WW-Domain	3.55 ± 1.16	0.74 ± 0.09	0.55 ± 0.10	1.97 ± 0.55	0.90 ± 0.06	0.73 ± 0.08
Villin	2.74 ± 0.82	0.86 ± 0.06	0.66 ± 0.11	1.76 ± 0.60	0.94 ± 0.04	0.82 ± 0.08
NTL9	1.88 ± 0.56	0.81 ± 0.09	0.75 ± 0.09	0.85 ± 0.34	0.96 ± 0.05	0.95 ± 0.05
BBL	4.92 ± 1.40	0.68 ± 0.09	0.45 ± 0.10	5.01 ± 1.53	0.67 ± 0.09	0.44 ± 0.12
Protein B	4.05 ± 1.23	0.61 ± 0.09	0.48 ± 0.10	3.30 ± 1.36	0.65 ± 0.08	0.56 ± 0.12
Homeodomain	3.98 ± 1.38	0.79 ± 0.08	0.57 ± 0.12	3.96 ± 1.41	0.82 ± 0.08	0.61 ± 0.14
Protein G	2.43 ± 0.76	0.82 ± 0.09	0.66 ± 0.10	1.63 ± 0.62	0.91 ± 0.06	0.82 ± 0.09
α3D	3.87 ± 0.92	0.70 ± 0.06	0.48 ± 0.08	2.91 ± 0.49	0.69 ± 0.04	0.57 ± 0.05
λ-repressor	6.45 ± 1.60	0.54 ± 0.07	0.34 ± 0.09	5.48 ± 1.44	0.56 ± 0.08	0.40 ± 0.10

to 99.7% (Chignolin) of generated frames stay below a bond RMSZ of 10. This fraction of distorted frames is among the lowest for Chignolin and Trp-Cage, the proteins for which Costa et al. (2025) report chemically invalid states.

Table D.19: Covalent geometry per fast folding protein for conformations generated by `MarS-FM-mFlood` with finetuning and for the MD reference, computed over all frames. Besides mean RMSZ, we report the median bond RMSZ and the percentage of generated frames with bond RMSZ above 10, i.e., far beyond the maximum of 3.5 observed in the MD reference. We omit both for angle RMSZ, which shows no comparable tail (99th percentile at most 6.3). Averages over proteins are reported in Table 6.

	Generated				Reference	
	Bond RMSZ			Angle RMSZ	Bond RMSZ	Angle RMSZ
Protein	mean	median	% > 10	mean	mean	mean
Chignolin	2.92	2.65	0.33	2.51	1.95	1.87
Trp-Cage	3.23	2.81	0.93	2.70	2.00	1.99
BBA	2.95	2.62	0.78	2.31	1.96	1.84
WW-Domain	3.21	2.77	1.24	2.52	1.97	1.89
Villin	3.08	2.59	1.38	2.34	1.93	1.86
NTL9	3.08	2.60	1.42	2.27	1.96	1.83
BBL	3.48	2.67	2.58	2.40	1.99	1.86
Protein B	2.95	2.43	1.57	2.21	1.93	1.84
Homeodomain	2.62	2.29	0.96	2.12	1.92	1.84
Protein G	3.35	2.58	2.59	2.29	1.96	1.85
α3D	3.35	2.51	2.94	2.27	1.98	1.86
λ-repressor	3.69	2.57	4.07	2.31	1.99	1.86

Since no model is trained on fast folding proteins, and the finetuned variants are adapted on the 348 K subset of `mdCATH` only (Appendix C.7), both the close agreement for broad native basins and the recovered Protein G landscape are obtained on proteins not seen during training.

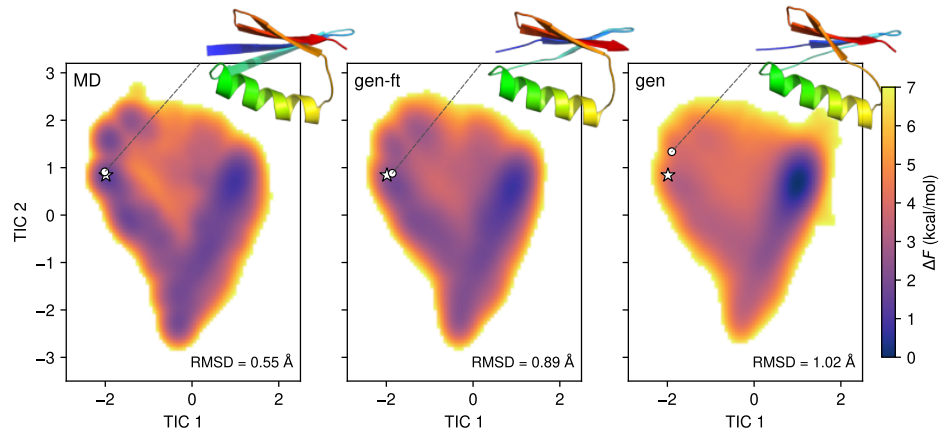


Figure D.5: Free energy landscapes of Protein G in TICA coordinates ($\tau = 10$ ns), comparing the MD reference trajectories (**Left**, `MD`) to samples generated by `MarS-FM-mFlood` with finetuning (**Middle**, `gen-ft`) and without finetuning (**Right**, `gen`). Cartoon insets show the frames closest in RMSD to the experimental crystal structure (1MI0, marked by a star); the corresponding TIC coordinates are marked by a circle, and the per-panel C_α RMSD to the crystal structure is annotated in the lower right. We clamp ΔF, the difference to the lowest basin, at 7 kcal/mol.

E Properties of the masked Flood complex

E.1 Bottleneck stability

Theorem 1. *Let $L \subset \mathbb{R}^d$ be finite, and for each $\sigma \in \mathrm{Del}(L)$ let $X_\sigma, X'_\sigma \subset \mathbb{R}^d$ be finite such that $X_\sigma \subset X_\tau$ and $X'_\sigma \subset X'_\tau$ for all $\tau \subset \sigma$. The masked Flood complex then satisfies the following stability result for all $i \geq 0$:*

$$d_B\big(\,\mathrm{dgm}_i(\mathrm{mFlood}(\{X_\sigma\}_\sigma, L)), \mathrm{dgm}_i(\mathrm{mFlood}(\{X'_\sigma\}_\sigma, L))\big) \leq \max_{\sigma \in \mathrm{Del}(L)} d_H(X_\sigma, X'_\sigma) \ . \tag{E.25}$$

Proof. Let $\sigma \in \mathrm{Del}(L)$ and let $f(\sigma), f'(\sigma)$ denote its respective filtration values. The triangle inequality for directed Hausdorff distances implies

$$f(\sigma) = \max_{p \in \mathrm{conv}(\sigma)} d(p, X_\sigma) \leq \max_{p \in \mathrm{conv}(\sigma)} d(p, X'_\sigma) + \max_{x' \in X'_\sigma} d(x', X_\sigma) = f'(\sigma) + \max_{x' \in X'_\sigma} d(x', X_\sigma) \ , \tag{E.26}$$

and analogously, $f'(\sigma) \leq f(\sigma) + \max_{x \in X_\sigma} d(x, X'_\sigma)$. In sum, we have

$$|f(\sigma) - f'(\sigma)| \leq \max\Big\{ \max_{x \in X_\sigma} d(x, X'_\sigma), \max_{x' \in X'_\sigma} d(x', X_\sigma) \Big\} = d_H(X_\sigma, X'_\sigma) \ . \tag{E.27}$$

Taking the maximum over all $\sigma \in \mathrm{Del}(L)$ gives a uniform bound, so that we can apply the stability theorem from Cohen-Steiner et al. (2007), which then finishes the proof. □

E.2 Computation from grid

As in Graf et al. (2025) we approximate the filtration values of mFlood using a grid of evenly spaced barycentric coordinates. Given a predefined number $m \in \mathbb{N}$ controlling the number of grid points along each edge, we denote the resulting grid points on a simplex σ by $P_\sigma = \{\sum_{i=0}^k \lambda_i v_i \colon \lambda_i \geq 0, \sum_{i=0}^k \lambda_i = 1, m\lambda_i \in \mathbb{N}_{\geq 0}\} \subset \mathrm{conv}\,\sigma$. We then define $\mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma)$ as

$$\mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma) = \Big\{ \sigma \in \mathrm{Del}(L) \colon P_\sigma \subset \bigcup_{x \in X_\sigma} B_r(x) \Big\} \ . \tag{E.28}$$

Equivalently, mFlood is the sublevel set of the function

$$f : \mathrm{Del}(L) \to \mathbb{R} \qquad \sigma \mapsto \max_{p \in P_\sigma} \min_{x \in X_\sigma} d(p, x) \tag{E.29}$$

which, as opposed to the non-convex optimization problem of finding the filtration values of the exact $\mathrm{mFlood}(\{X_\sigma\}_\sigma, L)$ can be computed directly from the pairwise distances between P_σ and X_σ. For all experiments, we use a resolution of $m = 20$ grid points per edge.

As we show below, f is a well-defined filtration function satisfying $f(\tau) \leq f(\sigma)$ whenever $\tau \subset \sigma$, and thus the family $\mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma)$ for $r \geq 0$ forms a filtration of $\mathrm{Del}(L)$. Indeed, if $\tau \subset \sigma$, then $P_\tau \subset P_\sigma$, and thus

$$f(\tau) = \max_{p \in P_\tau} \min_{x \in X_\tau} d(p, x) \leq \max_{p \in P_\sigma} \min_{x \in X_\tau} d(p, x) \leq \max_{p \in P_\sigma} \min_{x \in X_\sigma} d(p, x) = f(\sigma) \ . \tag{E.30}$$

Moreover, we have the following approximation guarantee.

Proposition 2. *Let $L \subset \mathbb{R}^d$ be a finite set, and for each $\sigma \in \mathrm{Del}(L)$ let $X_\sigma \subset \mathbb{R}^d$ be finite such that $X_\sigma \subset X_\tau$ for all $\tau \subset \sigma$, and given $m \in \mathbb{N}_{>0}$, let $P_\sigma = \{\sum_{i=0}^k \lambda_i v_i \colon \lambda_i \geq 0, \sum_{i=0}^k \lambda_i = 1, m\lambda_i \in \mathbb{N}_{\geq 0}\} \subset \mathrm{conv}\, \sigma$. Then, denoting $g(\{v_0, \ldots, v_k\}) := \sqrt{\sum_{0 \leq a < b \leq k} \|v_a - v_b\|^2}$, we have for every $i \geq 0$*

$$d_B(\mathrm{dgm}_i(\mathrm{mFlood}(\{X_\sigma\}_\sigma, L)), \mathrm{dgm}_i(\mathrm{mFlood}(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma))) \leq \frac{1}{m} \max_{\sigma \in \mathrm{Del}(L)} g(\sigma) \ . \tag{E.31}$$

Proof. We note that (Graf et al., 2025, Theorem 5) already shows $d_B\left(\mathrm{dgm}_i(\mathrm{Flood}(X, L)), \mathrm{dgm}_i(\mathrm{Flood}(X, L, P))\right) \leq \max_{\sigma \in \mathrm{Del}(L)} d_H(P_\sigma, \mathrm{conv}\, \sigma)$ for the original Flood complex, and (Graf et al., 2025, Lemma 6) shows that $d_H(P_\tau, \mathrm{conv}\, \tau) \leq \frac{1}{m}\sqrt{\sum_{i<j} \|v_i - v_j\|^2} = g(\tau)/m$ for any simplex τ. We therefore only need to show that (Graf et al., 2025, Theorem 5) extends to the masked Flood complex.

Indeed, it is clear from the definitions that $\tau \in \mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L)$ implies $\tau \in \mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma)$ since $P_\tau \subset \mathrm{conv}\, \tau$. Conversely, let $\tau \in \mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma)$. Thus, for each $p \in P_\tau$ there is $x_p \in X_\tau$ with $d(p, x_p) \leq r$. Moreover, with $\varepsilon := \max_{\sigma \in \mathrm{Del}(L)} d_H(P_\sigma, \mathrm{conv}\, \sigma)$ there is for any $q \in \mathrm{conv}\, \tau$ some $p' \in P_\tau$ with $d(q, p') \leq \varepsilon$. Hence,

$$d(q, X_\tau) \leq d(q, x_{p'}) \leq d(q, p') + d(p', x_{p'}) \leq r + \varepsilon \ .$$

which implies $\mathrm{mFlood}_r(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma) \subset \mathrm{mFlood}_{r+\varepsilon}(\{X_\sigma\}_\sigma, L)$. The classic stability theorem from Cohen-Steiner et al. (2007) then implies

$$\begin{aligned} &d_B\left(\mathrm{dgm}_i(\mathrm{mFlood}(\{X_\sigma\}_\sigma, L)), \mathrm{dgm}_i(\mathrm{mFlood}(\{X_\sigma\}_\sigma, L, \{P_\sigma\}_\sigma))\right) \\ &\quad \leq \max_{\sigma \in \mathrm{Del}(L)} d_H(P_\sigma, \mathrm{conv}\, \sigma) \leq \frac{1}{m} \max_{\sigma \in \mathrm{Del}(L)} g(\sigma) \ . \end{aligned}$$

□

F Runtime and memory

We performed runtime and peak-memory profiling for the PH computation (H_0,H_1,H_2), using 500 randomly selected `mdCATH` domains for runtime and the ten largest domains for memory. `Flood PH` is fastest at 14.4 ms per frame; `mFlood PH` requires 28.4 ms, comparable to edge-collapsed `Rips PH` (C_α) at 25.0 ms (but Rips scales substantially worse with protein size), and 3.3× faster than `Alpha PH` at 94.1 ms. All-atom Rips was infeasible at this scale. The overall cost on `mdCATH` arises from its large number of conformations, i.e., ∼12.5M conformations (∼5k domains with 5 replicas with ∼500 time steps), rather than individual protein size (maximal 500 C_α, ∼8000 atoms). Due to the low memory requirements, PH computation is parallelizable across domains (trivially by starting one process per domain), which reduces total wall time by an order of magnitude. Moreover, batching over simplices (as is default in the Flooder implementation) reduces `mFlood PH` peak GPU memory

Table F.20: Comparison of runtime and memory usage for PH computation in degrees 0 – 2.

Method	Batch	Wall / diagram (ms)	Est. total wall time (h)	Peak GPU memory (MB)	Peak host memory (MB)
`Rips PH` (C_α)	—	25.0 ± 56.0	79	—	333
`Alpha PH`	—	94.1 ± 63.5	301	—	409
`Flood PH`	1024	14.4 ± 7.1	46	93	1548
`Flood PH`	4096	14.9 ± 10.5	48	164	1550
`mFlood PH`	1024	28.4 ± 21.1	91	384	1528
`mFlood PH`	4096	28.4 ± 21.0	91	1185	1516

Table G.21: Ablation of homology degree contribution to CATH classification performance.

	Linear	**MLP**
`mFlood PH` (H_0)	0.692 ± 0.002	0.679 ± 0.009
`mFlood PH` (H_1)	0.767 ± 0.015	0.766 ± 0.023
`mFlood PH` (H_2)	0.752 ± 0.002	0.796 ± 0.029
`mFlood PH` (H_0, H_1)	0.767 ± 0.003	0.785 ± 0.011
`mFlood PH` (H_0, H_2)	0.761 ± 0.012	0.802 ± 0.004
`mFlood PH` (H_1, H_2)	0.789 ± 0.017	0.806 ± 0.033
`mFlood PH` (H_0, H_1, H_2)	0.791 ± 0.007	0.812 ± 0.021

from 1,185 MB to 384 MB without affecting runtime, while its approximately 1.5 GB host-memory footprint is dominated by fixed CUDA/Triton process overhead. The total wall time column lists the estimate without any parallelization across systems (which is trivially doable).

In terms of end-to-end training cost to obtain vectorizations, the k-means++ clustering on the points in the persistence diagram can be done in less than 10 min (Douze et al., 2024) (using FAISS on the GPU), and VAMPNet training at corpus-scale in less than 20 min on our system (NVIDIA H100 GPU, see Appendix C.8).

For generative modeling, once VAMPNet features are pre-computed for all training MD trajectories, no additional computational cost is added to `MarS-FM`.

G Ablations

G.1 Homology degree

We report which homology dimensions provide relevant information for the CATH-classification experiment. Indeed, higher-dimensional features do provide additional information, highlighting the relevance for methods (such as our masked Flood complex) that enable efficiently computing the latter.

G.2 Masking strategy

In the table below, we compare the current masking variant, `mFlood PH` (nbs0), which suppresses within-residue interactions while retaining longer-range geometry, with `mFlood PH` (nbs1), which also masks interactions between neighboring residues. This more aggressive mask leaves classification and regression largely unchanged: balanced accuracy changes from 0.812 ± 0.021 to 0.819 ± 0.014, with mixed R^2 changes of at most 0.016, but degrades MSM consistency. Although VAMP-2 increases from 1.503 to 1.515, CK RMSE worsens from 0.034 to 0.052 and π-JS from 0.120 to 0.250, eliminating the gains over the unmasked Flood complex variant. Thus, additional masking is not uniformly beneficial, and the original minimal rule provides the best balance across tasks. This ablation also illustrates why VAMP-2 should not be interpreted in isolation.

Table G.22: Ablation on neighborhood size (number of residues) in the masking of `mFlood PH`. Regression probes here use a 5-layer MLP head trained for at most 10 epochs; shared rows therefore differ from Table 3 by at most 0.005.

(a) CATH classification (balanced accuracy)

Feature	Head	Balanced Acc.
Flood	mlp	0.800
mFlood (nbs0)	mlp	0.812
mFlood (nbs1)	mlp	0.819

(b) MSM quality assessment

Feature	VAMP2	CK-RMSE	π-JS
Flood	1.482	0.051	0.219
mFlood (nbs0)	1.503	0.034	0.120
mFlood (nbs1)	1.515	0.052	0.250

(c) Observable regression (R^2)

Feature	Head	RG	SSF	FNC	RMSD
Flood	mlp	0.864	0.848	0.864	0.781
mFlood (nbs0)	mlp	0.864	0.839	0.860	0.781
mFlood (nbs1)	mlp	0.848	0.851	0.858	0.770

G.3 Sensitivity to hyperparameters

For vectorizing persistence diagrams we need to select two hyperparameters: The number of centers, controlling the number of features the vectorization produces and the stretch parameter that controls how much low-lifetime points are downweighted. We evaluate their effect on three experiments: CATH classification, regression on observables and MSM construction. Throughout all experiments, k-means++ is used to determine cluster centers (re-initialized 5 times; objective function = total squared errors, as implemented in FAISS). For the third hyperparameter, the bandwidth, we use half the distance from each center to its closest other center, as in ATOL (Royer et al., 2021).

When varying the number of centers between 32–256, all results are fairly stable; only once the discretization becomes too coarse (e.g., 16 per homology dimension), results on all tasks deteriorate. Increasing the stretch quantile by an order of magnitude (1% to 10%) lowers CATH accuracy from 0.812 to 0.778, while regression and MSM quality are unaffected.

Table G.23: Sensitivity analysis on persistent homology vectorization parameters for CATH classification. * denotes the configuration reported in the main part of the manuscript.

(a) Number of structure elements

Feature	Head	Balanced Acc.
mFlood (16)	mlp	0.794 ± 0.031
mFlood (32)	mlp	0.827 ± 0.004
mFlood (64)	mlp	0.813 ± 0.034
mFlood (128)*	mlp	0.812 ± 0.021
mFlood (256)	mlp	0.807 ± 0.018

(b) Stretch quantile

Feature	Head	Balanced Acc.
mFlood (128, 1%)*	mlp	0.812 ± 0.021
mFlood (128, 5%)	mlp	0.809 ± 0.021
mFlood (128, 10%)	mlp	0.778 ± 0.015

H Societal impact

By improving MSM-based sampling within the generative `MarS-FM` framework, our approach may enable more efficient exploration of protein conformational landscapes, with potential applications in areas such as drug discovery and molecular design. Besides these benefits, considerations regarding societal impact analogous to those discussed in Kapuśniak et al. (2026) apply. For example, increased accessibility and realism of generative protein modeling may lower practical barriers to protein engineering, or encourage unwarranted confidence when applied outside validated regimes. While we regard these concerns legitimate, we currently assess the risk of misuse as relatively low and, therefore, do not identify a need for specific technical safeguards beyond standard responsible research practices.

Table G.24: Sensitivity analysis on persistent homology vectorization parameters for observable regression (R^2). * denotes the configuration reported in the main part of the manuscript. Regression probes here use a 5-layer MLP head trained for at most 10 epochs; the * rows therefore differ from Table 3 by at most 0.003.

(a) Number of structure elements

Feature	Head	RG	SSF	FNC	RMSD
mFlood (16)	mlp	0.841	0.806	0.840	0.764
mFlood (32)	mlp	0.861	0.832	0.857	0.778
mFlood (64)	mlp	0.856	0.820	0.849	0.773
mFlood (128)*	mlp	0.864	0.839	0.860	0.781
mFlood (256)	mlp	0.866	0.842	0.862	0.783

(b) Stretch quantile

Feature	Head	RG	SSF	FNC	RMSD
mFlood (128, 1%)*	mlp	0.864	0.839	0.860	0.781
mFlood (128, 5%)	mlp	0.864	0.838	0.859	0.781
mFlood (128, 10%)	mlp	0.864	0.837	0.858	0.780

Table G.25: Sensitivity analysis for MSM quality assessment. * denotes the configuration reported in the manuscript.

(a) Number of structure elements

Feature	VAMP2	CK-RMSE	π-JS
mFlood (16)	1.458	0.033	0.102
mFlood (32)	1.488	0.032	0.114
mFlood (64)	1.498	0.032	0.118
mFlood (128)*	1.503	0.034	0.120
mFlood (256)	1.500	0.034	0.121

(b) Stretch quantile

Feature	VAMP2	CK-RMSE	π-JS
mFlood (128, 1%)*	1.503	0.034	0.120
mFlood (128, 5%)	1.511	0.033	0.120
mFlood (128, 10%)	1.501	0.033	0.121

I ADDITIONAL FIGURES

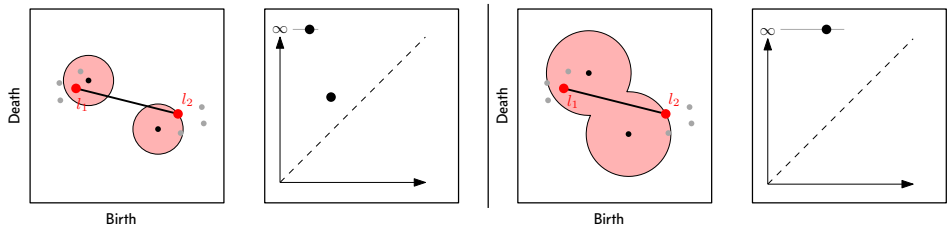


Figure I.6: Filtration values and H_0-diagrams of the masked Flood complex. Red markers indicate the landmarks. Points $x \in X_e$ that do not contribute to the flooding of the edge $e = \{l_1, l_2\}$ are grayed out. **Left**: All simplices have distinct filtration values. Thus, at the shown radius, there are two connected components, ($\{l_1\}$ and $\{l_2\}$). **Right**: $\{l_2\}$ and $\{l_1, l_2\}$ have equal filtration value, and $\forall r \geq 0$, there is at most one connected component.