PETIMOT: A NOVEL FRAMEWORK FOR INFERRING PROTEIN MOTIONS FROM SPARSE DATA USING SE(3)-EQUIVARIANT GRAPH NEURAL NETWORKS

Valentin Lombard
Department of Computational, Quantitative,
and Synthetic Biology (CQSB), UMR 7238
IBPS, Sorbonne Université, CNRS
Paris, 75005, France
Valentin.Lombard@sorbonne-universite.fr

Sergei Grudinin
Univ. Grenoble Alpes,
CNRS, Grenoble INP, LJK
38000 Grenoble, France.
Sergei.Grudinin
@univ-grenoble-alpes.fr

Elodie Laine
Department of Computational, Quantitative, and Synthetic Biology (CQSB), UMR 7238, IBPS,
Sorbonne Université, CNRS Paris, 75005, France
Institut universitaire de France (IUF)
Elodie.Laine@sorbonne-universite.fr

ABSTRACT

Proteins move and deform to ensure their biological functions. Despite significant progress in protein structure prediction, approximating conformational ensembles at physiological conditions remains a fundamental open problem. This paper presents a novel perspective on the problem by directly targeting continuous compact representations of protein motions inferred from sparse experimental observations. We develop a task-specific loss function enforcing data symmetries, including scaling and permutation operations. Our method PETIMOT (Protein sEquence and sTructure-based Inference of MOTions) leverages transfer learning from pre-trained protein language models through an SE(3)-equivariant graph neural network. When trained and evaluated on the Protein Data Bank, PETIMOT shows superior performance in time and accuracy, capturing protein dynamics, particularly large/slow conformational changes, compared to state-of-the-art flow-matching approaches and traditional physics-based models.

1 INTRODUCTION

Proteins orchestrate biological processes in living organisms by interacting with their environment and adapting their three-dimensional (3D) structures to engage with cellular partners, including other proteins, nucleic acids, small-molecule ligands, and co-factors. In recent years, spectacular advances in high-throughput deep learning (DL) technologies have provided access to reliable predictions of protein 3D structures at the scale of entire proteomes (Varadi et al., 2024). These breakthroughs have also highlighted the complexities of protein conformational heterogeneity. State-of-the-art predictors struggle to model alternative conformations, fold switches, large-amplitude conformational changes, and solution ensembles (Chakravarty et al., 2025).

The success of AlphaFold2 (Jumper et al., 2021) has stimulated machine-learning approaches focused on inference-time interventions in the model to generate structural diversity. They include enabling or increasing dropout (Brysbaert et al., 2024; Wallner, 2023), or manipulating the evolutionary information given as input to the model (Kalakoti & Wallner, 2024; Wayment-Steele et al., 2023; Del Alamo et al., 2022; Stein & Mchaourab, 2022). Despite promising results on specific families, several studies have emphasised the difficulties in rationalising the effectiveness of these modifications and interpreting them (Porter et al., 2024; Bryant & Noé, 2024). Moreover, these cannot be transferred to protein language model-based predictors that do not rely on multiple sequence alignments. Researchers have also actively engaged in the development of deep-learning

frameworks based on diffusion, or the more general flow matching, to generate conformational ensembles (Wang et al., 2025). While family-specific models proved useful in exploring native-like conformational landscapes, models trained across protein families still fail to approximate solution ensembles (Abramson et al., 2024).

This work presents a new glance at the protein conformational diversity problem. Instead of learning and sampling from multi-dimensional empirical distributions, we propose to learn eigenspaces (the structure) of the positional covariance matrices in collections of experimental 3D structures and generalize these over different homology levels. Our motivation is that the diversity present within different 3D structures of the same protein or close homologs is a good proxy for the conformational heterogeneity of proteins in solution (Best et al., 2006) and can generally be (almost fully) explained by a small set of linear vectors, also referred to as modes (Lombard et al., 2024a; Yang et al., 2009). Although linear spaces may not be well-suited for capturing highly complex non-linear motions, such as loop deformations, they offer multiple advantages. These include faster learning due to the reduced complexity of the model, improved explainability as the components directly correspond to interpretable data dimensions, faster inference, and the straightforward combination or integration of multiple data dimensions.

To summarize, our main contributions are:

- We provide a novel formulation of the protein conformational diversity problem.
- We present a novel benchmark representative of the Protein Data Bank (PDB) structural diversity and compiled with a robust pipeline (Lombard et al., 2024a), along with data- and task-specific metrics.
- We develop a SE(3)-equivariant Graph Neural Network architecture equipped with a novel symmetry-aware loss function for comparing linear subspaces, with invariance to permutation and scaling. Our model, PETIMOT, leverages embeddings from pre-trained protein language models (pLMs), building on prior proof-of-concept work demonstrating that they encode information about functional protein motions (Lombard et al., 2024b).
- PETIMOT is trained on sparse experimental data without any use of simulation data, in contrast with Timewarp for instance (Klein et al., 2024). Moreover, our model does not require physics-based guidance or feedback, unlike (Wang et al., 2025) for instance.
- Our results demonstrate the capability of PETIMOT to generalise across protein families (contrary to variational autoencoder-based approaches) and to compare favorably in running time and accuracy to AlphaFlow, ESMFlow, and the Normal Mode Analysis.

2 Related Works

Protein structure prediction. AlphaFold2 was the first end-to-end deep neural network to achieve near-experimental accuracy in predicting protein 3D structures, even for challenging cases with low sequence similarity to proteins with resolved structures (Jumper et al., 2021). It extracts information from an input multiple sequence alignment (MSA) and outputs all-atom 3D coordinates. Later works have shown that substituting the input alignment by embeddings from a protein language model can yield comparable performance (Lin et al., 2023; Hayes et al., 2024; Weissenow et al., 2022; Wu et al., 2022).

Generating conformational ensembles. Beyond the single-structure frontier, several studies have underscored the limitations and potential of protein structure predictors (PSP) for generating alternative conformations (Saldaño et al., 2022; Lane, 2023; Bryant & Noé, 2024; Chakravarty et al., 2025). Approaches focused on re-purposing AlphaFold2 include dropout-based massive sampling (Brysbaert et al., 2024; Wallner, 2023), guiding the predictions with state-annotated templates (Faezov & Dunbrack Jr, 2023; Heo & Feig, 2022), and inputting shallow, masked, corrupted, subsampled or clustered alignments (Kalakoti & Wallner, 2024; Wayment-Steele et al., 2023; Del Alamo et al., 2022; Stein & Mchaourab, 2022). Despite promising results, these approaches remain computationally expensive and their generalisability, interpretability, and controllability remain unclear (Bryant & Noé, 2024; Chakravarty et al., 2025). More recent works have aimed at overcoming these limitations by directly optimising PSP learnt embeddings under low-dimensional ensemble constraints (Yu et al., 2025).

Another line of research has consisted in fine-tuning or re-training AlphaFold2 and other single-state PSP under diffusion or flow matching frameworks (Jing et al., 2024; Abramson et al., 2024; Krishna et al., 2024). For instance, the AlphaFlow/ESMFlow method progressively denoises samples drawn from a harmonic prior under flow field controlled by AlphaFold or ESMFold (Jing et al., 2024). It compares favourably with MSA subsampling or clustering baselines, with a substantially superior precision-diversity Pareto frontier. More generally, diffusion- and flow matching-based models allow for efficiently generating diverse conformations conditioned on the presence of ligands or cellular partners (Jing et al., 2023; Ingraham et al., 2023; Wang et al., 2025; Liu et al., 2024; Zheng et al., 2024). Despite their strengths, these techniques are prone to hallucination.

Parallel related works have sought to directly learn generative models of equilibrium Boltzmann distributions using normalising flows (Noé et al., 2019; Klein et al., 2024), or machine-learning force fields based on equivariant graph neural network (GNN) representations (Wang et al., 2024a), to enhance or replace molecular dynamics (MD) simulations.

Protein conformational heterogeneity manifold learning. Unsupervised, physics-based Normal Mode Analysis (NMA) has long been effective for inferring functional modes of deformation by leveraging the topology of a single protein 3D structure (Grudinin et al., 2020; Hoffmann & Grudinin, 2017; Hayward & Go, 1995). While appealing for its computational efficiency, the accuracy of NMA strongly depends on the initial topology (Laine & Grudinin, 2021), limiting its ability to model extensive secondary structure rearrangements. Recent efforts have sought to address these limitations by directly learning continuous, compact representations of protein motions from sparse experimental 3D structures. These approaches employ dimensionality reduction techniques, from classical manifold learning methods (Lombard et al., 2024a) to neural network architectures like variational auto-encoders (Ramaswamy et al., 2021). By projecting motions onto a learned low-dimensional manifold, these methods enable reconstruction of accurate, physico-chemically realistic conformations, both within the interpolation regime and near the convex hull of the training data (Lombard et al., 2024a). Additionally, they assist in identifying collective variables from molecular dynamics (MD) simulations, supporting importance-sampling strategies (Chen et al., 2023; Belkacemi et al., 2021; Bonati et al., 2021; Wang et al., 2020; Ribeiro et al., 2018). Despite these advances, such approaches are currently constrained to family-specific models.

E(3)-equivariant graph neural networks. Graph Neural Networks (GNN) have been extensively used to represent protein 3D structures. They are robust to transformations of the Euclidean group, namely rotations, reflections, and translations, as well as to permutations. In their simplest formulation, each node represents an atom and any pair of atoms are connected by an edge if their distance is smaller than a cutoff or among the smallest k interatomic distances. Many works have proposed to enrich this graph representation with SE(3)-equivariant features informing the model about interatomic directions and orientations (Ingraham et al., 2019; Jing et al., 2020; Dauparas et al., 2022; Krapp et al., 2023; Ingraham et al., 2023; Wang et al., 2024b). For instance, VisNet captures the full local geometric information, including bonds, angles, as well as dihedral torsion and improper angles with node-wise high-order geometric tensors (Wang et al., 2024b). Moreover, to go beyond local 3D neighbourhoods while maintaining sub-quadratic complexity, Chroma adds in randomly sampled long-range connections (Ingraham et al., 2023).

3 METHODS

3.1 DATA REPRESENTATION

To generate training data, we exploit experimental protein single chain structures available in the Protein Data Bank. We first clustered these chains based on their sequence similarity. Then, within each cluster, we aligned the protein sequences and used the resulting mapping for superimposing the 3D coordinates (Lombard et al., 2024a). It may happen that some residues in the multiple sequence alignment do not have resolved 3D coordinates in all conformations. To account for this uncertainty, we assigned a confidence score w_i to each residue i computed as the proportion of conformations including this residue. The 3D superimposition puts the conformations' centers of mass to zero and then aims at determining the optimal least-squares rotation minimizing the Root Mean Square Deviation (RMSD) between any conformation and a reference conformation, while accounting for

the confidence scores (Kabsch, 1976; Kearsley, 1989),

$$E = \frac{1}{\sum_i w_i} \sum_i w_i (\vec{r}_{ij} - \vec{r}_{i0})^2, \tag{1}$$

where $\vec{r}_{ij} \in \mathbb{R}^3$ is the ith centred coordinate of the jth conformation and $\vec{r}_{i0} \in \mathbb{R}^3$ is the ith centred coordinate of the reference conformation. Next, we defined our ground-truth targets as eigenspaces of the coverage-weighted Cα-atom positional covariance matrix,

$$C = \frac{1}{m-1} R^c W (R^c)^T = \frac{1}{m-1} (R - R^0) W (R - R^0)^T, \tag{2}$$

where R is the $3N \times m$ positional matrix with N the number of residues and m is the number of conformations, R^0 contains the coordinates of the reference conformation, and W is the $3N \times 3N$ diagonal coverage matrix. The covariance matrix is a $3N \times 3N$ square matrix, symmetric and real. We decompose C as $C = YDY^T$, where Y is a $3N \times 3N$ matrix with each column defining a coverage-weighted eigenvector or a principal component that we interpret as a *linear motion*. D is a diagonal matrix containing the eigenvalues. The latter highly depend on the sampling biases in the PDB and thus we do not aim at predicting them.

3.2 Problem formulation

For a protein of length N, let Y be $3N \times K$ *orthogonal* ground-truth deformations,

$$Y^T Y = I_K. \tag{3}$$

Our goal is to find coverage-weighted vectors $X \in \mathbb{R}^{3N \times L}$ whose components l *approximate* some components k of the ground truth Y:

$$W^{\frac{1}{2}} \tilde{\mathbf{x}}_l \approx \mathbf{y}_k. \tag{4}$$

Below, we provide three alternative formulations for this problem.

3.3 Geometric Loss

The least-square formulation. PETIMOT's loss function serves two key purposes: it enables effective training of the network to predict subspaces representing multiple distinct modes of deformations – *i.e.*, with low overlap between the subspace's individual linear vectors, while preventing convergence to a single dominant mode. For each protein of length N with a coverage W, we compare ground-truth directions Y with predicted motion directions X by computing a weighted pairwise *least-square difference* $\mathcal{L}_{kl}$ for each pair of a k direction in the ground truth and an l direction in the prediction,

$$\mathcal{L}_{kl} = \frac{1}{N} \sum_{i=1}^{N} \|\vec{y}_{ik} - w_i^{1/2} c_{kl} \vec{x}_{il}\|^2 = \frac{1}{N} \mathbf{y}_k^T \mathbf{y}_k - \frac{1}{N} \frac{(\mathbf{y}_k^T W^{\frac{1}{2}} \mathbf{x}_l)^2}{\mathbf{x}_l^T W \mathbf{x}_l}, \tag{5}$$

where we scaled the ground-truth tensors such that $Y^T Y = N I_K$ and we used the fact that the optimal scaling coefficients c_{kl} between the k-th ground truth vector and the l-th prediction are given by

$$c_{kl} = \frac{\sum_{i=1}^{N} w_i^{\frac{1}{2}} \mathbf{y}_{ik}^T \mathbf{x}_{il}}{\sum_{i=1}^{N} w_i \mathbf{x}_{il}^T \mathbf{x}_{il}} = \frac{\mathbf{y}_k^T W^{\frac{1}{2}} \mathbf{x}_l}{\mathbf{x}_l^T W \mathbf{x}_l}. \tag{6}$$

This invariance to global scaling is motivated by the fact that we aim at capturing the relative magnitudes and directions of the motion patterns rather than their sign or absolute amplitudes.

Linear assignment problem. We then formulate an *optimal linear assignment problem* to find the minimum-cost matching between the ground-truth and the predicted directions. Specifically, we aim to solve the following assignment problem for the least-square (LS) costs,

$$\begin{aligned} &\text{LS Loss} = \min_{\pi \in S_J} \sum_{k=1}^{\min(K,L)} \mathcal{L}_{k,\pi(k)} \\ &\text{subject to:} \\ &\pi : \{1, \ldots, \min(K, L)\} \to \{1, \ldots, L\}, \;\; \pi(k) \neq \pi(k') \text{ for } k \neq k', \end{aligned} \tag{7}$$

where K and L are the number of ground-truth and predicted directions respectively, and $\pi(k)$ represents the index of the predicted direction assigned to the k-th ground truth direction. This formulation ensures an optimal one-to-one matching, while accommodating cases where the number of predicted and ground-truth directions differs. We backpropagate the loss only through the optimally matched pairs, using *scipy* linear_sum_assignment. We have also tested a smooth version of the loss above with continuous gradients, but it did not improve the performance.

The subspace coverage formulation. We propose another formulation of the problem in terms of the subspace coverage metrics (Amadei et al., 1999; Leo-Macias et al., 2005; David & Jacobs, 2011). Specifically, we sum up *squared sinus* (SS) dissimilarities between ground-truth and predicted directions (formally computed as one minus squared cosine similarity),

$$\text{SS Loss} = 1 - \frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{y}_k^T W^{\frac{1}{2}}\mathbf{x}_l^{\perp})^2, \tag{8}$$

where the subspace $\{\mathbf{x}_l^{\perp}\}$ is obtained by orthogonalising the coverage-weighted predicted linear subspace $\{W^{\frac{1}{2}}\mathbf{x}_l\}$, where $\mathbf{x}_l^T W \mathbf{x}_l = 1$, using the Gram–Schmidt process. This operation ensures that the loss ranges from zero for mutually orthogonalising subspaces to one for identical subspaces and avoids artificially inflating the SS loss due to redundancy in the predicted motions. The order in which the predicted vectors are orthogonalised does not influence the loss, guaranteeing stable training. Appendix A proves this statement.

Independent Subspace (IS) Loss. We can substitute the orthogonalisation procedure by using an auxiliary loss component for maximising the rank of the predicted subspace. For this purpose, we chose the squared cosine similarity computed between pairs of predicted vectors. The final expression for the *independent subspace* (IS) loss is

$$\text{IS Loss} = \frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{x}_k^T W \mathbf{x}_l)^2 - \frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{y}_k^T W^{\frac{1}{2}}\mathbf{x}_l)^2, \tag{9}$$

where the predictions $\{\mathbf{x}_l\}$ are normalised prior to the loss computation such that $\mathbf{x}_l^T W \mathbf{x}_l = 1$ and the scaling factor K^2 ensures that the loss ranges between 0 and 1. Appendix A analyses the stability of this formulation.

3.4 Architecture

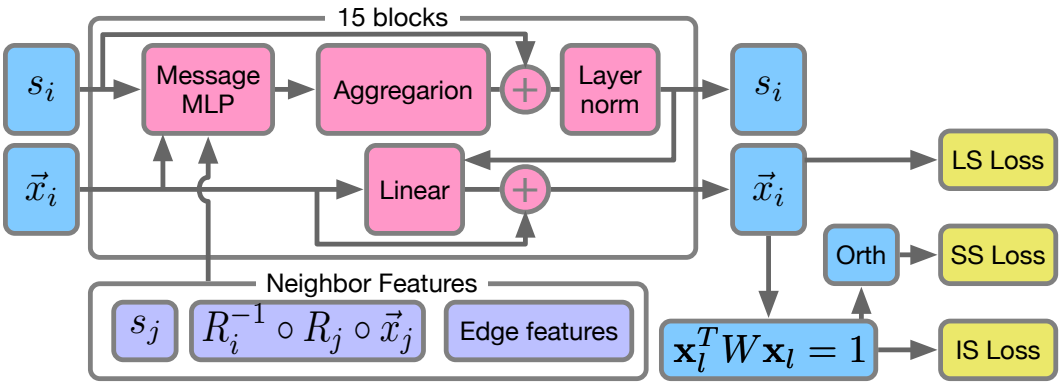

Figure 1: **PETIMOT's architecture overview.** The model processes both sequence embeddings (s) and motion vectors ($\vec{x}$) through 15 message-passing blocks. Each block updates both representations by aggregating information from neighboring residues. Neighbor features are computed in the reference frame of the central residue i, ensuring SE(3) equivariance. The geometric features encoded in the edges capture the relative spatial relationships between residue pairs. Three types of losses (LS, SS, and IS) are computed, with prior normalization of the predictions for the IS and SS losses, and an additional orthogonalisation of the predictions for the SS loss.

Dual-Track Representation. PETIMOT processes protein sequences through a message-passing neural network that simultaneously handles residue embeddings and motion vectors in local coordinate frames (Fig. 1). For each residue i, we define and update a node embedding $\mathbf{s}_i \in \mathbb{R}^d$ initialized from protein language model features and a set of K motion vectors $\{\vec{x}_{ik}\}_{k=1}^{K} \in \mathbb{R}^{3\times K}$ initialized randomly. The message passing procedure is detailed in Algorithm B.1 of Appendix B.2.

Graph Construction. The protein is represented as a graph where nodes correspond to the residues, and edges capture spatial relationships. For each residue i, we connect to its k nearest neighbors based on Cα distances and l randomly selected residues. This hybrid connectivity scheme ensures both local geometric consistency and global information flow, while maintaining sparsity for computational efficiency. Indeed, our model scales *linearly* with the length N of a protein. In our base model we set $k = 5$ and $l = 10$.

Node features. We chose ProstT5 as our default protein language model for initialising node embeddings (Heinzinger et al., 2023). This structure-aware pLM offers an excellent balance between model size – including the number of parameters and embedding dimensionality – and performance (Lombard et al., 2024b).

Local Reference Frames. Each residue's backbone atoms (N, CA, C) define a local reference frame through a rigid transformation $T_i \in SE(3)$. For each residue pair (i, j), we compute their relative transformation $T_{ij} = T_i^{-1} \circ T_j$ from which we extract the rotation $R_{ij} \in SO(3)$ and translation $\vec{t}_{ij} \in \mathbb{R}^3$. Under global rotations and translations of the protein, these relative transformations remain invariant.

Edge Features. Edge features e_{ij} provide an SE(3)-invariant encoding of the protein structure through relative orientations, translational offsets, protein chain distance, and a complete description of peptide plane positioning captured by pairwise backbone atom distances. See Appendix B.3 for more details. The training procedure is detailed in Appendix B.4.

4 RESULTS

Training and evaluation. We trained PETIMOT against linear motions extracted from all ~750,000 protein chains from the PDB (as of June 2023) clustered at 80% sequence identity and coverage. We augmented the data by computing the motions with respect to 5 reference conformations per collection. The full training set comprises 25,595 samples. We set the numbers of predicted and ground-truth motions, $K = L = 4$. See Appendix B.1 for more details.

To evaluate PETIMOT's ability to capture protein continuous conformational heterogeneity, we tested it on 824 proteins, each one associated with a conformational collection held out during training and validation. At inference, we consider $w_i = 1, \forall i = 1..N$. We rely on four main evaluation metrics aimed at addressing the following questions:

- Is PETIMOT able to approximate at least one of the main linear motions of a given protein? For this, we rely on the minimum LS error over all possible pairs of predicted and ground-truth vectors.
- To what extent does PETIMOT capture the main motion linear subspace of a given protein? For this, we use the global SS error.
- Is PETIMOT able to identify the residues that move the most? Here, we rely on the magnitude error, $\frac{1}{N}\sum_{i=1}^{N}(\|\vec{y}_{ik}\|^2 - \|c_{kl}\vec{x}_{il}\|^2)$.
- How fast is PETIMOT at inference?

Comparison with other methods. PETIMOT showed a better capacity to approximate individual motions and to globally capture motion subspaces than the flow matching-based frameworks AlphaFlow and ESMFlow, and also the unsupervised physics-based Normal Mode Analysis (Fig. 2a-b). It approximated at least one motion with reasonable accuracy (LS error below 0.6) for 43.57% of the test proteins, while the success rate was only 31.80%, 26.82%, and 24.88% for AlphaFlow,

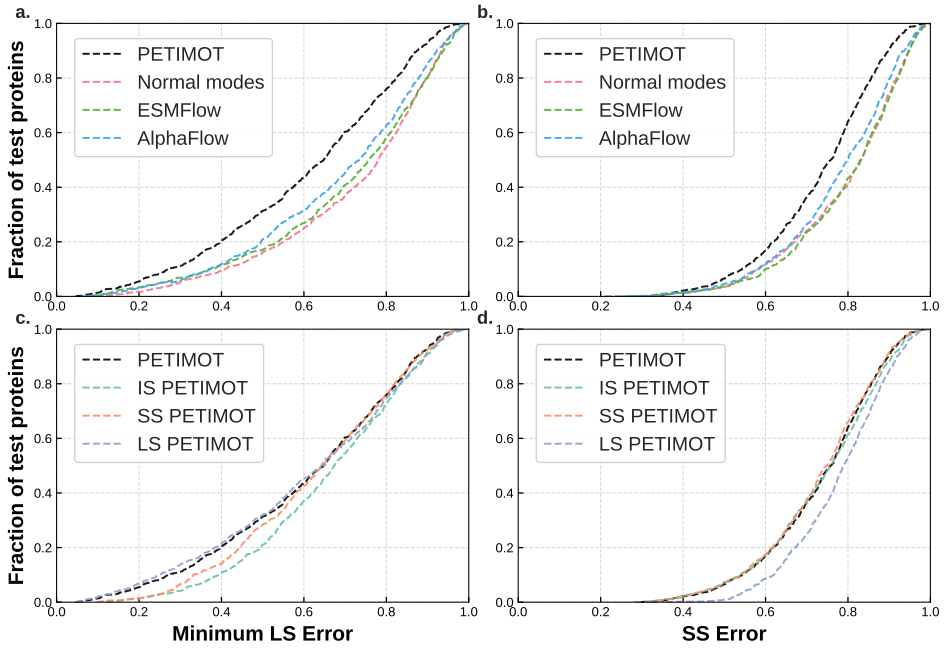

Figure 2: **Cumulative error curves computed on the test proteins. a-b.** Comparison between PETIMOT base model and three other methods. **c-d.** Comparison between different losses implemented in PETIMOT. The loss of the base model is LS + SS. **a,c.** Minimum LS error corresponding to the best matching pair of predicted and ground-truth motions. **b,d.** SS error computed between the entire predicted and ground-truth subspaces.

ESMFlow, and the NMA, respectively. PETIMOT's best predicted vector better matched a ground-truth vector than any other methods in 43.57% of the cases (Fig. 3a). PETIMOT was also better at identifying which residues contribute the most to the motions (Table C.1). PETIMOT was also significantly faster at inference - it took about 16s for the whole test set, followed by NOLB (44s), ESMFlow (11h) and AlphaFlow (38h), see Table C.1. See Appendix B.5 for more evaluation details.

Comparison of problem formulations. Our base model combining the LS and SS loses with equal weights outperforms all three individual losses, LS, SS, and LS (Fig. 2c-d). It strikes an excellent balance between approximating individual motions with high accuracy (Fig. 2c) and globally covering the motion subspaces (Fig. 2d). By comparison, the SS and IS losses tend to underperform on individual motions while the LS loss tends to provide lower coverage of the ground-truth subspaces. See Appendix C for additional results.

Contribution of sequence and structure features. We performed an ablation study to assess the contribution of sequence and structure information to our architecture. Our results show that ProstT5 slightly outperforms the more recent and larger pLM, ESM-Cambrian 600M (ESM Team, 2024) (Fig. B.2). Geometrical information about protein structure provides the most significant contribution, as replacing ProstT5 embeddings with random numbers has only a small impact on network performance. Conversely, the network's performance without structural information strongly depends on the chosen pLM. While the structure-aware embeddings from ProstT5 partially compensate for missing 3D structure information, relying solely on ESM-C embeddings results in poor performance (Fig. B.2). Moreover, connecting each residue to its 15 nearest neighbours (sorted according to Cα-Cα distances) in the protein graph results in lower performance compared to introducing randomly chosen edges or even fully relying on random connectivity (Fig. B.4).

Generalisation capability. Because our conformational collections are defined based on 80% sequence identity and coverage thresholds, some test proteins may be homologs of the training proteins. Yet, the sequence identity and structural similarity (TM-score) of the test proteins with respect

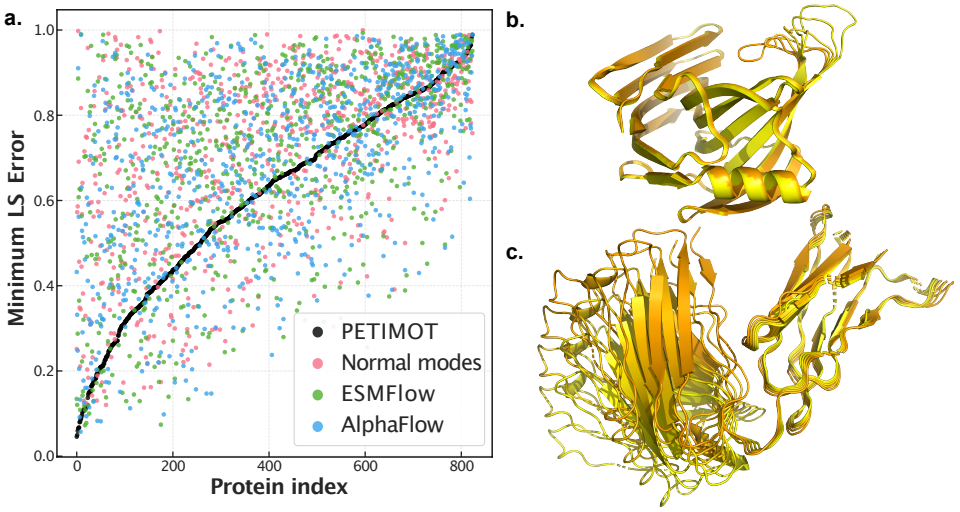

Figure 3: **Individual predictions. a.** The per-protein minimum LS errors, computed for the best-matching pairs between predicted and ground-truth vectors, are reported for PETIMOT (black), the NMA (red), AlphaFlow (blue) and ESMFlow (green). The values are in ascending order of the errors computed for PETIMOT, from best to worse. **b-c.** Trajectories generated by deforming a protein structure along PETIMOT best predicted motion. Five trajectory snapshots are shown colored from yellow to orange. **b.** *Bacillus subtilis* xylanase A (PDB id: 3EXU, chain A). **c.** Murine Fab fragment (PDB id: 7SD2, chain A).

to the training set do not determine the quality of PETIMOT predictions (Fig. C.3). PETIMOT provides high-quality predictions for a number of test proteins that do not share any detectable similarity and only weak structural similarity (TM-score below 0.5) to the training set.

Conformation generation. PETIMOT allows straightforwardly generating conformational ensembles or trajectories by deforming an initial protein 3D structure along one or a combination of predicted motions. We showcase this functionality on two example proteins, the xylanase A from *Bacillus subtilis* and the periplasmic domain of Gliding motility protein GldM from *Capnocytophaga canimorsus* (Fig. 3b-c). We used PETIMOT predictions to generate physically realistic conformations representing either the open-to-closed transition of xylanase A thumb (Fig. 3b) or the flexibility of the heavy chain antibody IgE/Fab anti-profilin Hev b 8 (Fig. 3c). Figures C.4 and C.5 compare predicted motions for these proteins with the ground truth.

5 Conclusion

In this work, we have proposed a new perspective on the problem of capturing protein continuous conformational heterogeneity. Compared to state-of-the-art methods, our approach goes beyond generating alternative protein conformations by directly inferring compact and continuous representations of protein motions. Our comprehensive analysis of PETIMOT's predictive capabilities demonstrates its performance and utility for understanding how proteins deform to perform their functions. Our work opens ways to future developments in protein motion manifold learning, with exciting potential applications in protein engineering and drug development.

Code and Data. The code and the data are available at `https://github.com/PhyloSofS-Team/PETIMOT`.

Acknowledgements. This work has been funded by the European Union (ERC, PROMISE, 101087830). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

MEANINGFULNESS STATEMENT

Understanding protein motions is essential for grasping the dynamic nature of life at the molecular level, as these movements enable biological function. A meaningful representation of life must thus capture continuous, complex, and functional dynamics of proteins beyond static structural snapshots. Our work contributes a novel problem formulation for predicting protein conformational heterogeneity by inferring compact, data-driven representations of protein motions from sparse experimental data. Our approach, PETIMOT, provides a robust framework integrating data symmetries reflecting protein physical and geometric properties alongside evolutionary semantics from protein language models, revealing how proteins move and deform to fulfill their functions.

REFERENCES

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Ahdritz, G., Bouatta, N., Floristean, C., Kadyan, S., Xia, Q., Gerecke, W., O'Donnell, T. J., Berenberg, D., Fisk, I., Zanichelli, N., et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods*, 21(8):1514–1524, 2024.

Amadei, A., Ceruso, M. A., and Di Nola, A. On the convergence of the conformational coordinates basis set obtained by the essential dynamics analysis of proteins' molecular dynamics simulations. *Proteins: Structure, Function, and Bioinformatics*, 36(4):419–424, 1999.

Belkacemi, Z., Gkeka, P., Lelièvre, T., and Stoltz, G. Chasing collective variables using autoencoders and biased trajectories. *Journal of chemical theory and computation*, 18(1):59–78, 2021.

Best, R. B., Lindorff-Larsen, K., DePristo, M. A., and Vendruscolo, M. Relation between native ensembles and experimental structures of proteins. *Proceedings of the National Academy of Sciences*, 103(29):10901–10906, 2006.

Bonati, L., Piccini, G., and Parrinello, M. Deep learning the slow modes for rare events sampling. *Proceedings of the National Academy of Sciences*, 118(44):e2113533118, 2021.

Bryant, P. and Noé, F. Structure prediction of alternative protein conformations. *Nature Communications*, 15(1):7328, 2024.

Brysbaert, G., Raouraoua, N., Mirabello, C., Véry, T., Blanchet, C., Wallner, B., and Lensink, M. Massivefold: unveiling alphafold's hidden potential with optimized and parallelized massive sampling. 2024.

Chakravarty, D., Lee, M., and Porter, L. L. Proteins with alternative folds reveal blind spots in alphafold-based protein structure prediction. *Current Opinion in Structural Biology*, 90:102973, 2025.

Chen, H., Roux, B., and Chipot, C. Discovering reaction pathways, slow variables, and committor probabilities with machine learning. *Journal of Chemical Theory and Computation*, 19(14):4414–4426, 2023.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

David, C. C. and Jacobs, D. J. Characterizing protein motions from structure. *Journal of Molecular Graphics and Modelling*, 31:41–56, 2011.

Del Alamo, D., Sala, D., Mchaourab, H. S., and Meiler, J. Sampling alternative conformational states of transporters and receptors with alphafold2. *Elife*, 11:e75751, 2022.

ESM Team, . Esm cambrian: Revealing the mysteries of proteins with unsupervised learning. Evolutionary Scale Website, 2024. URL https://evolutionaryscale.ai/blog/esm-cambrian. Available from: https://evolutionaryscale.ai/blog/esm-cambrian.

Faezov, B. and Dunbrack Jr, R. L. Alphafold2 models of the active form of all 437 catalytically-competent typical human kinase domains. *bioRxiv*, pp. 2023–07, 2023.

Grudinin, S., Laine, E., and Hoffmann, A. Predicting protein functional motions: an old recipe with a new twist. *Biophysical journal*, 118(10):2513–2525, 2020.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024. doi: 10.1101/2024.07.01.600583. URL https://www.biorxiv.org/content/early/2024/07/02/2024.07.01.600583.

Hayward, S. and Go, N. Collective variable description of native protein dynamics. *Annual review of physical chemistry*, 46(1):223–250, 1995.

Heinzinger, M., Weissenow, K., Sanchez, J. G., Henkel, A., Steinegger, M., and Rost, B. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, pp. 2023–07, 2023.

Heo, L. and Feig, M. Multi-state modeling of g-protein coupled receptors at experimental accuracy. *Proteins: Structure, Function, and Bioinformatics*, 90(11):1873–1885, 2022.

Hoffmann, A. and Grudinin, S. Nolb: Nonlinear rigid block normal-mode analysis method. *Journal of chemical theory and computation*, 13(5):2123–2134, 2017.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

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

Jing, B., Erives, E., Pao-Huang, P., Corso, G., Berger, B., and Jaakkola, T. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

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

Joosten, R. P., Long, F., Murshudov, G. N., and Perrakis, A. The pdb_redo server for macromolecular structure model optimization. *IUCrJ*, 1(4):213–220, 2014.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976. doi: 10.1107/S0567739476001873. URL https://doi.org/10.1107/S0567739476001873.

Kalakoti, Y. and Wallner, B. Afsample2: Predicting multiple conformations and ensembles with alphafold2. *bioRxiv*, pp. 2024–05, 2024.

Kearsley, S. K. On the orthogonal transformation used for structural comparisons. *Acta Crystallographica Section A: Foundations of Crystallography*, 45(2):208–210, 1989.

Klein, L., Foong, A., Fjelde, T., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noé, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Krapp, L. F., Abriata, L. A., Cortés Rodriguez, F., and Dal Peraro, M. Pesto: parameter-free geometric deep learning for accurate prediction of protein binding interfaces. *Nature communications*, 14(1):2175, 2023.

Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

Laine, E. and Grudinin, S. Hopma: Boosting protein functional dynamics with colored contact maps. *The Journal of Physical Chemistry B*, 125(10):2577–2588, 2021.

Lane, T. J. Protein structure prediction has reached the single-structure frontier. *Nature Methods*, 20(2):170–173, 2023.

Leo-Macias, A., Lopez-Romero, P., Lupyan, D., Zerbino, D., and Ortiz, A. R. An analysis of core deformations in protein superfamilies. *Biophysical journal*, 88(2):1291–1299, 2005.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, J., Li, S., Shi, C., Yang, Z., and Tang, J. Design of ligand-binding proteins with atomic flow matching. *arXiv preprint arXiv:2409.12080*, 2024.

Lombard, V., Grudinin, S., and Laine, E. Explaining conformational diversity in protein families through molecular motions. *Scientific Data*, 11(1):752, 2024a.

Lombard, V., Timsit, D., Grudinin, S., and Laine, E. Seamoon: Prediction of molecular motions based on language models. *bioRxiv*, pp. 2024–09, 2024b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Porter, L. L., Artsimovitch, I., and Ramírez-Sarmiento, C. A. Metamorphic proteins and how to find them. *Current opinion in structural biology*, 86:102807, 2024.

Ramaswamy, V. K., Musson, S. C., Willcocks, C. G., and Degiacomi, M. T. Deep learning protein conformational space with convolutions and latent interpolations. *Physical Review X*, 11(1): 011052, 2021.

Ribeiro, J. M. L., Bravo, P., Wang, Y., and Tiwary, P. Reweighted autoencoded variational bayes for enhanced sampling (rave). *The Journal of chemical physics*, 149(7), 2018.

Saldaño, T., Escobedo, N., Marchetti, J., Zea, D. J., Mac Donagh, J., Velez Rueda, A. J., Gonik, E., García Melani, A., Novomisky Nechcoff, J., Salas, M. N., et al. Impact of protein conformational diversity on alphafold predictions. *Bioinformatics*, 38(10):2742–2748, 2022.

Stein, R. A. and Mchaourab, H. S. Speach_af: Sampling protein ensembles and conformational heterogeneity with alphafold2. *PLOS Computational Biology*, 18(8):e1010483, 2022.

Varadi, M., Bertoni, D., Magana, P., Paramval, U., Pidruchna, I., Radhakrishnan, M., Tsenkov, M., Nair, S., Mirdita, M., Yeo, J., et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1):D368–D375, 2024.

Wallner, B. Afsample: improving multimer prediction with alphafold using massive sampling. *Bioinformatics*, 39(9):btad573, 2023.

Wang, T., He, X., Li, M., Li, Y., Bi, R., Wang, Y., Cheng, C., Shen, X., Meng, J., Zhang, H., et al. Ab initio characterization of protein molecular dynamics with ai2bmd. *Nature*, pp. 1–9, 2024a.

Wang, Y., Ribeiro, J. M. L., and Tiwary, P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Current opinion in structural biology*, 61:139–145, 2020.

Wang, Y., Wang, T., Li, S., He, X., Li, M., Wang, Z., Zheng, N., Shao, B., and Liu, T.-Y. Enhancing geometric representations for molecules with equivariant vector-scalar interactive message passing. *Nature Communications*, 15(1):313, 2024b.

Wang, Y., Wang, L., Shen, Y., Wang, Y., Yuan, H., Wu, Y., and Gu, Q. Protein conformation generation via force-guided se (3) diffusion models, 2024. doi: 10.48550. *arXiv preprint ARXIV.2403.14088*, 2025.

Wayment-Steele, H. K., Ojoawo, A., Otten, R., Apitz, J. M., Pitsawong, W., Hömberger, M., Ovchinnikov, S., Colwell, L., and Kern, D. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, pp. 1–3, 2023.

Weissenow, K., Heinzinger, M., and Rost, B. Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. *Structure*, 30(8):1169–1177, 2022.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022.

Yang, L.-W., Eyal, E., Bahar, I., and Kitao, A. Principal component analysis of native ensembles of biomolecular structures (pca_nest): insights into functional dynamics. *Bioinformatics*, 25(5): 606–614, 2009.

Yu, Z., Liu, Y., Lin, G., Jiang, W., and Chen, M. ESMAdam: a plug-and-play all-purpose protein ensemble generator. *bioRxiv*, pp. 2025–01, 2025.

Zheng, S., He, J., Liu, C., Shi, Y., Lu, Z., Feng, W., Ju, F., Wang, J., Zhu, J., Min, Y., Zhang, H., Tang, S., Hao, H., Jin, P., Chen, C., Noé, F., Liu, H., and Liu, T.-Y. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5): 558–567, 2024. doi: 10.1038/s42256-024-00837-3. URL https://doi.org/10.1038/s42256-024-00837-3.

APPENDICES

A INVARIANCE OF THE PROPOSED LOSSES

Theorem A.1. *SS Loss is invariant under unitary transformations of X and Y subspaces.*

Proof. Without loss of generality, let us assume that we apply a unitary transformation $U \in \mathbb{R}^{K\times K}$ to a subspace $X^{\perp} \in \mathbb{R}^{3N\times K}$, such that the result $X' = X^{\perp}U$, with $X' \in \mathbb{R}^{3N\times K}$, spans the same subspace as $X^{\perp}$, as it is a linear combination of the original basis vectors from $X^{\perp}$. Then, let us rewrite the SS loss as

$$\text{SS Loss} = 1 - \frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{y}_k^T W^{\frac{1}{2}}\mathbf{x}_l^{\perp})^2 = 1 - \frac{1}{K}||Y^T W^{\frac{1}{2}} X^{\perp}||_F^2. \tag{A.1}$$

As the Frobenius matrix norm is invariant under orthogonal, or more generally, unitary, transformations, $||Y^T W^{\frac{1}{2}} X^{\perp} U||_F^2 = ||Y^T W^{\frac{1}{2}} X^{\perp}||_F^2$, which completes the proof. □

Corollary A.1.1. *The SS loss is invariant to the direction permutations in the Gram-Schmidt orthogonalization process.*

Proof. Let us consider two linear subspaces $X_1^{\perp}$ and $X_2^{\perp}$ resulting from the Gram-Schmidt orthogonalization of X, where we arbitrarily choose the order of the orthogonalization vectors. Both $X_1^{\perp}$ and $X_2^{\perp}$ will span the same subspace as X, and since both $X_1^{\perp}$ and $X_2^{\perp}$ are also orthogonal, one is a unitary transformation of the other, $X_2^{\perp} = X_1^{\perp}U$, which completes the proof. □

Theorem A.2. *IS Loss is invariant under unitary transformations of X and Y subspaces.*

Proof. Following the previous proof, without loss of generality, let us assume that we apply an orthogonal (unitary) transformation $U \in \mathbb{R}^{K\times K}$ to a subspace $X \in \mathbb{R}^{3N\times K}$, such that the result $X' = XU$, with $X' \in \mathbb{R}^{3N\times K}$, spans the same subspace as X. Then, let us rewrite the IS loss as

$$\text{IS Loss} = \frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{x}_k^T W \mathbf{x}_l)^2 - \frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}(\mathbf{y}_k^T W^{\frac{1}{2}}\mathbf{x}_l)^2 = \frac{1}{K^2}||X^T W X||_F^2 - \frac{1}{K^2}||Y^T W^{\frac{1}{2}} X||_F^2. \tag{A.2}$$

As the Frobenius matrix norm is invariant under orthogonal transformations, $||Y^T W^{\frac{1}{2}} X U||_F^2 = ||Y^T W^{\frac{1}{2}} X||_F^2$, and $||U^T X^T W X U||_F^2 = ||X^T W X||_F^2$, which completes the proof. □

B METHODS DETAILS

B.1 TRAINING DATA

Conformational collections. To generate the training data, we utilized DANCE (Lombard et al., 2024a) to construct a non-redundant set of conformational collections representing the entire PDB as of June 2023. Wherever possible, we enhanced the data quality by replacing raw PDB coordinates with their updated and optimized counterparts from PDB-REDO (Joosten et al., 2014). Each conformational collection was designed to include only closely related homologs, ensuring that any two protein chains within the same collection shared at least 80% sequence identity and coverage. Collections with insufficient data points were excluded as we require at least 5 conformations. To simplify the data, we retained only Cα atoms (option `-c`) and accounted for coordinate uncertainty by applying weights (option `-w`).

Handling missing data. The conformations in a collection may have different lengths reflected by the introduction of gaps when aligning the amino acid sequences. We fill these gaps with the coordinates of the conformation used to center the data. In doing so, we avoid introducing biases through reconstruction of the missing coordinates. Moreover, to explicitly account for data uncertainty, we

assign confidence scores to the residues and include them in the structural alignment step and the eigen decomposition. The confidence score of a position i reflects its coverage in the alignment,

$$w_i = \frac{1}{m} \sum_S \mathbb{1}_{a_i^S \neq \text{"X"}}, \tag{B.1}$$

where "X" is the symbol used for gaps and m is the number of conformations. The structural alignment of the jth conformation onto the reference conformation amounts to determining the optimal rotation that minimises the following function (Kabsch, 1976; Kearsley, 1989),

$$E = \frac{1}{\sum_i w_i} \sum_i w_i (r_{ij}^c - r_{i0}^c)^2, \tag{B.2}$$

where r_{ij}^c is the ith centred coordinate of the jth conformation and r_{i0}^c is the ith centred coordinate of the reference conformation. The resulting aligned coordinates are then multiplied by the confidence scores prior to the PCA, as we explain below.

Eigenspaces of positional covariance matrices. The Cartesian coordinates of each conformational ensemble can be stored in a matrix R of dimension $3N \times m$, where N is the number of residues (or positions in the associated multiple sequence alignment) and n is the number of conformations. Each position is represented by a C-α atom. We compute the coverage-weighted (to account for missing data, as explained above) covariance matrix as in Eq. 2. The covariance matrix is a $3N \times 3N$ square matrix, symmetric and real.

We decompose C as $C = VDV^T$, where V is a $3N \times 3N$ matrix with each column defining a sqrt-coverage-weighted eigenvector or a principal component that we interpret as a *linear motion*. D is a diagonal matrix containing the eigenvalues. Specifically, the kth principal component was expressed as a set of 3D (sqrt-coverage-weighted) displacement vectors $\vec{x}^{\text{GT}} ik, i = 1, 2, ...L$ for the L Cα atoms of the protein residues. To enable cross-protein comparisons, the vectors were normalized such that $\sum i = 1^L |\vec{x}_{ik}^{\text{GT}}|^2 = L$. The sum of the eigenvalues $\sum_{k=1}^{3m} \lambda_k$ amounts to the total positional variance of the ensemble (measured in Å^2) and each eigenvalue reflects the amount of variance explained by the associated eigenvector.

Data augmentation. The reference conformation used to align and center the 3D coordinates corresponds to the protein chain with the most representative amino acid sequence. To increase data diversity, four additional reference conformations were defined for each collection. At each iteration, the new reference conformation was selected as the one with the highest RMSD relative to the previous reference. This iterative strategy maximizes the variability of the extracted motions by emphasizing the impact of changing the reference.

B.2 MESSAGE PASSING

The node embeddings and predicted motion vectors are updated iteratively according to the following algorithm.

Algorithm B.1 PETIMOT Message Passing Block

function MESSAGEPASSING($\{\mathbf{s}_i\}, \{\vec{x}_i\}, \{\mathcal{N}eigh(i)\}, \{R_{ij}, e_{ij}\}$):
# $\{\mathbf{s}_i\}_{i=1}^N$ ▷ Node embeddings
# $\{\vec{x}_i\}_{i=1}^N$ ▷ Motion vectors in local frames
# $\{\mathcal{N}eigh(i)\}_{i=1}^N$ ▷ Node neighborhoods
# $\{R_{ij}, e_{ij}\}$ ▷ Relative geometric features
  for $i = 1$ to N **do**
    for $j \in \mathcal{N}eigh(i)$ **do**
      $\vec{x}_j^i \leftarrow R_{ij}\vec{x}_j$ ▷ Project motion in frame i
      $m_{ij} \leftarrow \text{MessageMLP}(\mathbf{s}_i, \mathbf{s}_j, \vec{x}_i, \vec{x}_j^i, e_{ij})$
    end for
    $m_i \leftarrow \text{Mean}_j(m_{ij})$ ▷ Aggregate messages
    $\mathbf{s}_i \leftarrow \mathbf{s}_i + \text{LayerNorm}(m_i)$ ▷ Update embedding
    $\vec{x}_i \leftarrow \vec{x}_i + \text{Linear}([\mathbf{s}_i, \vec{x}_i])$ ▷ Update motion
  end for
  return $\{\mathbf{s}_i\}_{i=1}^N, \{\vec{x}_i\}_{i=1}^N$
end function

B.3 SE(3)-EQUIVARIANT FEATURES

We represent protein structures as attributed graphs. The node embeddings are computed with the pre-trained protein language model ProstT5 (Heinzinger et al., 2023). It is a fine-tuned version of the sequence-only model T5 that translates amino acid sequences into sequences of discrete structural states and reciprocally.

The edge embeddings are computed using SE(3)-invariant features derived from the input backbone, similarly to prior works (Ingraham et al., 2023; Dauparas et al., 2022; Ingraham et al., 2019). Specifically, the features associated with the edge e_{ij} from node (atom) i to node (atom) j are:

- **Quaternion representation:** A 4-dimensional quaternion encoding the relative rotation R_{ij} between the local reference frames of residues i and j.
- **Relative translation:** A 3-dimensional vector representing the translation $\vec{t}_{ij}$ between the local reference frames.
- **Chain separation:** The sequence separation between residues i and j, encoded as $\log(|i - j| + 1)$.
- **Spatial separation:** The logarithm of the Euclidean distance between residues i and j, computed as $\log(\|\vec{t}_{ij}\| + \epsilon)$, where $\epsilon = 10^{-8}$.
- **Backbone atoms distances:** Distances between all backbone atoms (N, Cα, C, O) at residues i and j, encoded through a radial basis expansion. For each pairwise distance d_{ab}, we compute:

$$f_k(d_{ab}) = \exp\left(-\frac{(d_{ab} - \mu_k)^2}{2\sigma^2}\right), \tag{B.3}$$

  where $\{\mu_k\}_{k=1}^{20}$ are centers spaced linearly in $[0, 20]$ Å and $\sigma = 1$ Å. This creates a 16 × 20 = 320 dimensional feature vector, as we have 16 pairwise distances (4 × 4 atoms) each expanded in 20 basis functions.

B.4 TRAINING PROCEDURE

The model was optimized using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 5e-4 and weight decay of 0.01. We employed gradient clipping with a maximum norm of 10.0 and mixed precision training with PyTorch's Automatic Mixed Precision. The learning rate was adjusted using torch's ReduceLROnPlateau scheduler, which monitored the validation loss, reducing the learning rate by a factor of 0.2 after 10 epochs without improvement. Training was performed with a batch size of 32 for both training and validation sets. We implemented early stopping with a patience of 50 epochs, monitoring the validation loss. The model achieving the best validation performance was

selected for final evaluation. We trained the model on a single NVIDIA A100-SXM4-80GB GPU. One epoch took about 9 minutes of real time.

B.5 Evaluation procedures

Comparison with AlphaFlow and ESMFlow and ESMFlow. We compared our approach with the flow-matching based frameworks AlphaFlow and ESMFlow for generating conformational ensembles. For this, we downloaded the distilled "PDB" models from `https://github.com/bjing2016/alphaflow`. We executed AlphaFlow using the following command,

```
python predict.py --noisy_first --no_diffusion --mode alphafold
--input_csv seqs.csv --msa_dir msa_dir/
--weights alphaflow_pdb_distilled_202402.pt --samples 50
--outpdb output_pdb/
```

AlphaFlow relies on OpenFold (Ahdritz et al., 2024) to retrieve the input multiple sequence alignment (MSA). ESMFlow was launched using the same command with an additional `--mode esmfold` flag and its corresponding weights. We used AlphaFlow and ESMFlow to generate 50 conformations for each test protein and then we treated each ensemble as a conformational collection. We then aligned all members of the created collections to the reference conformations of the ground-truth collections. We used the identity coverage weights here. Finally, from the aligned collections, we extracted the principal linear motions. We shall additionally mention that we did not filter or adapt our test set to the AlphaFlow and ESMFlow methods. In other words, there can be certain data leakage between AlphaFlow/ESMFlow train data and our test examples.

Comparison with the Normal Mode Analysis. We also compared our approach with the physics-based unsupervised Normal Mode Analysis (NMA) method (Hayward & Go, 1995). The NMA takes as input a protein 3D structure and builds an elastic network model where the nodes represent the atoms and the edges represent springs linking atoms located close to each other in 3D space. The four lowest normal modes are obtained by diagonalizing the mass-weighted Hessian matrix of the potential energy of this network. We used the highly efficient NOLB method, version 1.9, downloaded from `https://team.inria.fr/nano-d/software/nolb-normal-modes/` (Hoffmann & Grudinin, 2017) to extract the first K normal modes from the test protein 3D conformations. Specifically, we used the following command

```
NOLB INPUT.pdb -c 10 -x -n 4 --linear -s 0 --format 1 --hetatm
```

We retained only the Cα atoms and defined the edges in the elastic network using a distance cutoff of 10 Å.

B.6 Ablation Studies

To understand the impact of different components on the performance of our model, we carried out ablation studies. We list them blow.

Model architecture variations.

- Network depth: We experimented with different numbers of message-passing layers (5 and 10 layers compared to our default value of 15 layers).
- Layer sharing: We tested a variant where all message-passing layers share the same parameters, as opposed to our default where each layer has unique parameters.
- Reduced internal embedding dimension: We tested a model with a smaller internal embedding dimension of 128 instead of the default 256.

Figure B.1 shows the evaluation of these modifications. A shallow 5-layers network underperforms on all evaluation metrics. The difference between other variants is not very significant.

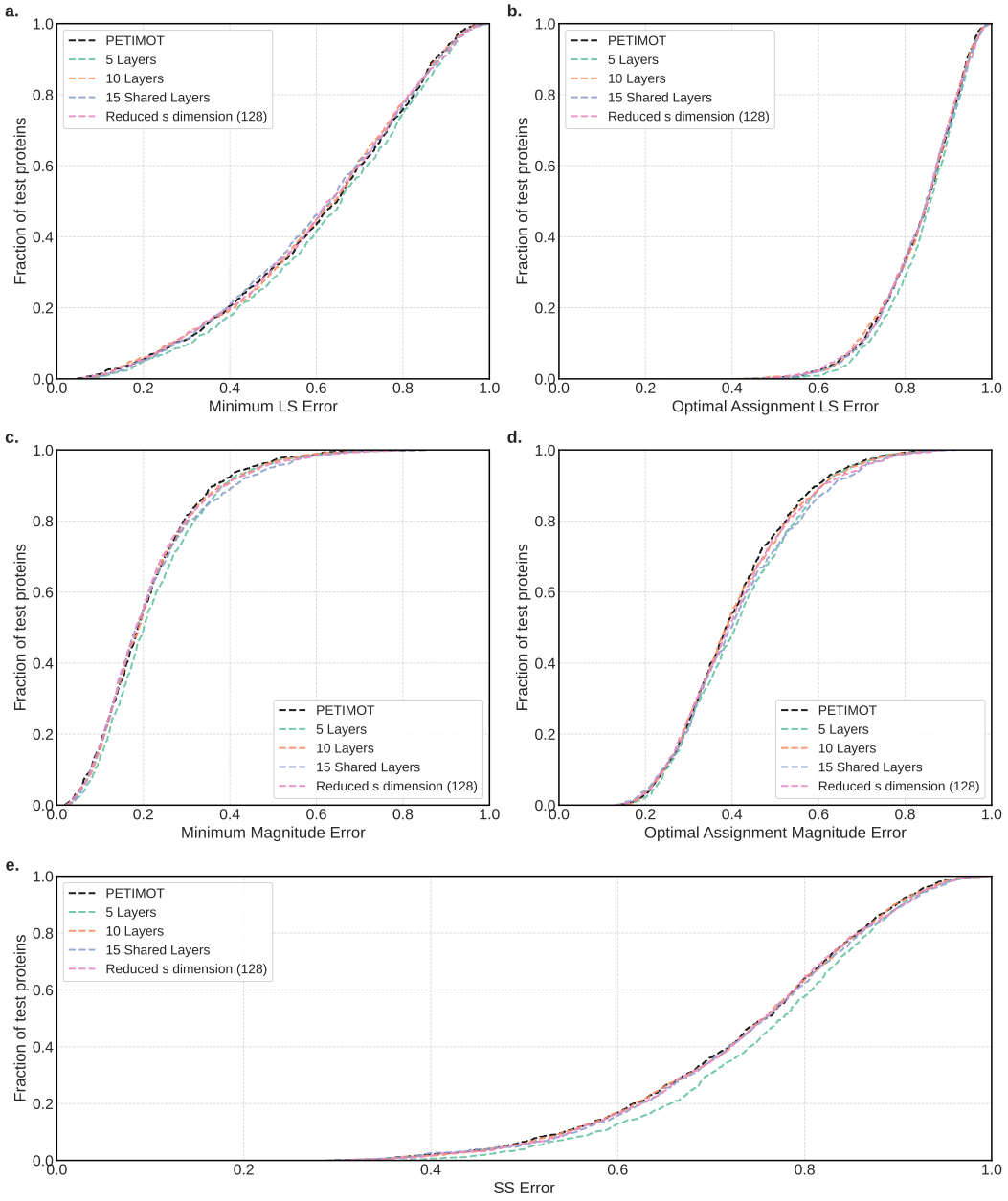

Figure B.1: **Network depth ablation.** We report cumulative curves for LS error (a-b), magnitude error (c-d), and SS error (e). For each protein, we computed the error either for the best-matching pair of predicted and ground-truth vectors (a,c) or for the best combination of four pairs of predicted and ground-truth vectors (b,d). We vary the number of layers in the network and the embedding dimension.

Structure and sequence information ablation.

- Structure ablation: We removed all structural information from the model to assess the importance of geometric features and the performance with the PLM embeddings only. We did it by removing the edge attributes of the input of the message passing MLP.

- Sequence ablation: We ablated sequence information by replacing protein language model embeddings with random embeddings, testing them both with and without structural information.
- Embedding variants: We evaluated a different protein language model (ESMC-600M), both with and without structural tokens.

The evaluation results are shown in Fig. B.2. The results demonstrate that while both ProstT5 and ESM-Cambrian 600M perform similarly when combined with structural information, removing structural features leads to markedly different outcomes. ProstT5 embeddings partially compensate for the missing structural information, likely due to their structure-aware training, while relying solely on ESM-C embeddings results in poor performance.

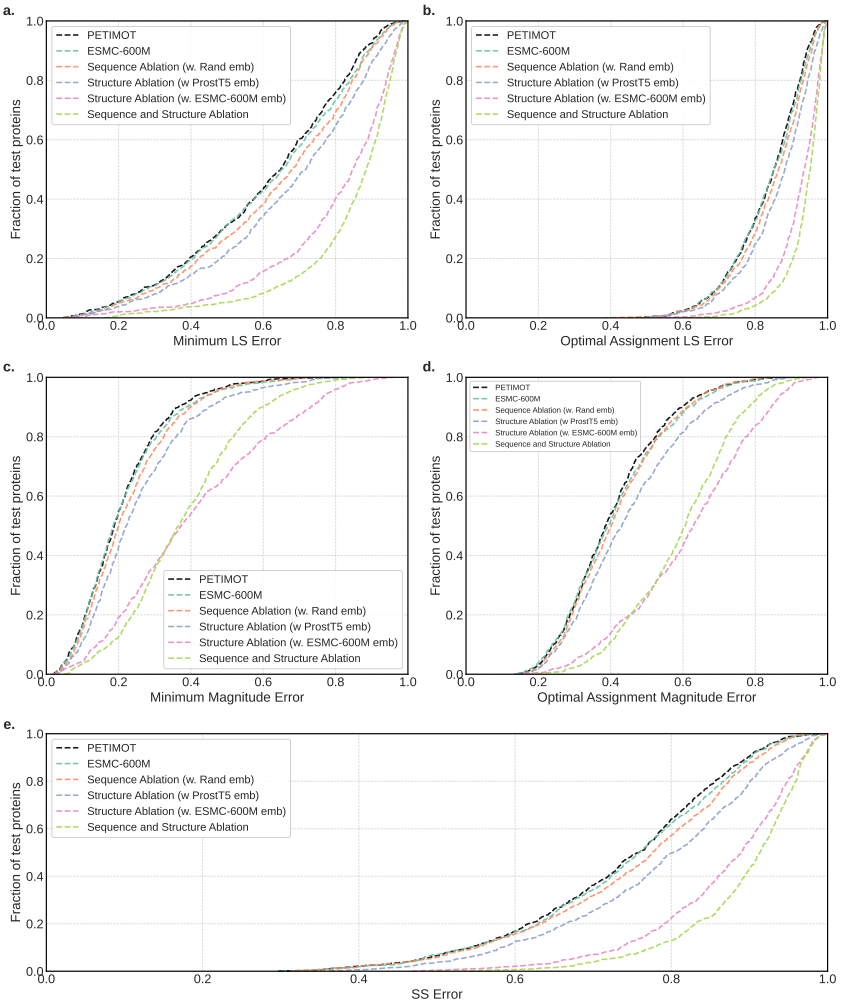

Figure B.2: **Structure and sequence information ablation study.** We report cumulative curves for LS error (a-b), magnitude error (c-d), and SS error (e). For each protein, we computed the LS and magnitude errors either for the best-matching pair of predicted and ground-truth vectors (a,c) or for the best combination of four pairs of predicted and ground-truth vectors (b,d).

Problem formulation ablation. We analyzed different combinations of our loss terms (compared to our default balanced weights of LS + SS):

- Least Square loss (LS): Using only the LS loss (weight 1.0).
- Squared Sinus loss (SS): Using only the SS loss (weight 1.0).

- Independent Subspaces (IS): Using only the IS loss (weight 1.0).

Figure B.3 compares three individual losses with the default option. The IS problem formulation underperforms on all the metrics. The default LS + SS formulation performs slightly better than those with individual loss components.

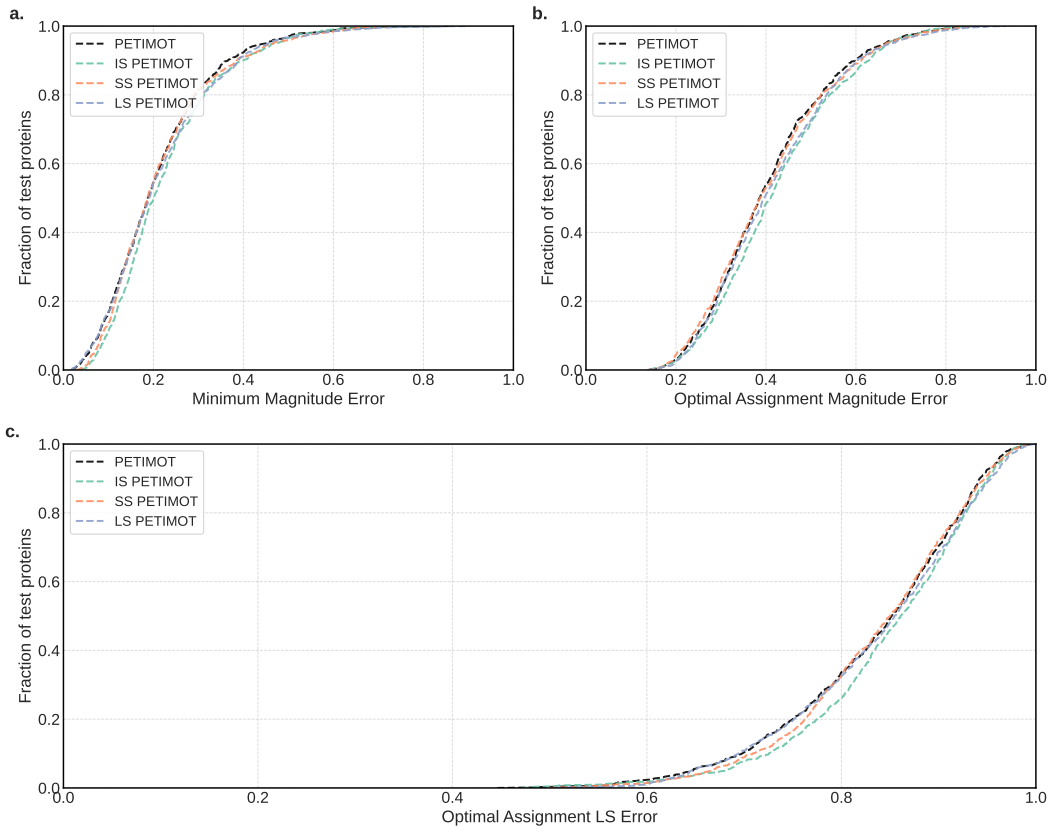

Figure B.3: **Performance comparison of different problem formulations.** We report cumulative curves for magnitude error (a,b) and LS error (c). For each protein, we computed the error either for the best-matching pair of predicted and ground-truth vectors (a) or for the best combination of four pairs of predicted and ground-truth vectors (b,c).

Graph connectivity ablation. We investigated different approaches to constructing the protein graph:

- Nearest neighbor-only: Using 15 nearest neighbors (sorted according to the corresponding Cα-Cα distances) without random edges.
- Random connections-only: Using 15 random edges without nearest neighbors. This set is updated between every layer at each epoch.
- Static connectivity: Using a fixed set of random neighbors between the layers. This set is updated at each epoch.

Figure B.4 shows the ablation results. We can see that the nearest neighbor-only setup underperforms on all the metrics. Among other options, the random connectivity-only option gets lower results at higher metrics values. The default option performs on par with the static connectivity, showing slightly better results on the optimal assignment magnitude error metrics.

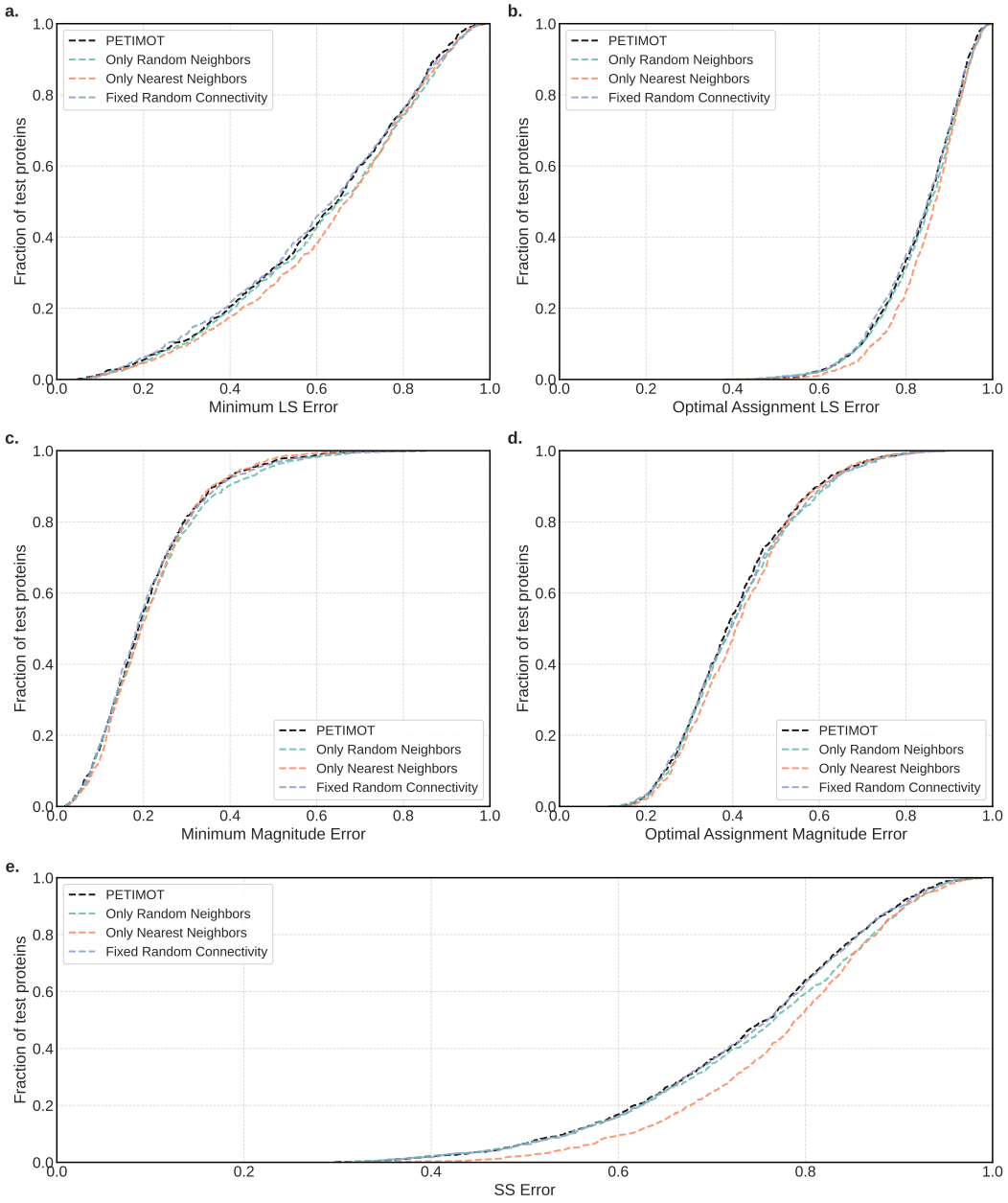

Figure B.4: **Graph connectivity ablation.** We report cumulative curves for LS error (a-b), magnitude error (c-d), and SS error (e). For each protein, we computed the error either for the best-matching pair of predicted and ground-truth vectors (a,c) or for the best combination of four pairs of predicted and ground-truth vectors (b,d). Only Random Neighbors: each residue (node) is connected to 15 randomly chosen residues and the connectivity changes after each layer. Only Nearest Neighbors: each residue (node) is connected to its 15 nearest neighbors in the input 3D structure. Fixed Random Connectivity: each residue (node) is connected to 15 residues randomly chosen at the beginning.

C ADDITIONAL RESULTS

Table C.1 lists additional results. The first line represents the success rates of four methods on the test set. The success rate is defined as the proportion of test proteins with at least one motion predicted

Metrics	PETIMOT	AlphaFlow	ESMFlow	NMA
Success Rate (%) ↑	**43.57**	31.80	26.82	24.88
Running time ↓	**15.82s**	38h 07min	10h 41min	43.59s
Min. LS Error ↓	**0.61** ± **0.22**	0.68 ± 0.21	0.70 ± 0.22	0.72 ± 0.20
Min. Magnitude Error ↓	**0.21** ± **0.12**	0.24 ± 0.12	0.26 ± 0.14	0.27 ± 0.14
OLA LS Error ↓	**0.83** ± **0.10**	0.86 ± 0.10	0.87 ± 0.10	0.88 ± 0.10
OLA Magnitude Error ↓	**0.41** ± **0.14**	0.43 ± 0.14	0.47 ± 0.15	0.48 ± 0.15
Global SS Error ↓	**0.73** ± **0.14**	0.78 ± 0.14	0.80 ± 0.14	0.79 ± 0.14

Table C.1: **Success rate and average performance on the test set.** Min. stands for the best matching pair of predicted and ground-truth vectors. OLA refers to the optimal linear assignment between all predicted and ground-truth vectors. Arrows indicate whether higher (↑) or lower (↓) metrics values are better. Best results are shown in **bold**. All results are averaged over 824 test proteins from the PDB test set. Running times are recorded on a Intel(R) Xeon(R) W-2245 CPU @ 3.90GHz equipped with GeForce RTX 3090. PETIMOT (with 4 directions), AlphaFlow (50 models), and ESMFlow (50 models) were executed on a GPU, while NOLB NMA (with 10 lowest modes) only used CPU.

at a reasonable accuracy, namely an LS error below 0.6. Other lines compare the least-square and magnitude errors computed for the best matching pair of the ground-truth and predicted directions, the least-square and magnitude errors using the optimal linear assignment method (comparing full four-dimensional subspaces), and the squared sinus error for the full subspaces. For all the metrics PETIMOT performs the best, with a particular striking difference in performance for the success rate metrics. However, it maybe not very informative to look at a single value averaged over the whole test set. Thus, we also suggest to analyze more informative plots, *e.g.* those in Fig. 2 and Fig. 3a.

Figure C.1 evaluates PETIMOT against NMA, ESMFlow and AlphaFlow approaches using additional metrics. These include the minimum magnitude error, the optimal assignment magnitude error, and the optimal assignment LS error. On all the metrics we see that PETIMOT outperforms the three other tested approaches.

We also experimented with a different number of predicted components. For these experiments, we trained additional models with the LS loss only, which are listed below:

- Single component prediction (1 mode).
- Reduced component prediction (2 modes).
- Extended component prediction (8 modes).

We compare these options with our default setting of 4 components. Figure C.2 shows the results. Increasing the number of predicted components from 1 to 8 improves the minimum LS errors, as having more predicted vectors naturally increases the chance of matching at least one ground-truth motion well. However, when evaluating the optimal linear assignment metrics, which measures overall subspace alignment, models with 1 or 2 components have an artificial advantage since they face fewer matching constraints. The 8-components model similarly benefits from having more candidate vectors to match against the 4 ground-truth components.

Figure C.3 compares the accuracy of the predicted test proteins (minimum LS loss) with the structural (TM-score) and sequence (sequence identity) distances to the training set. We do not see a clear correlation between the prediction accuracy and the similarity to the training examples. Please also see Fig. 3b-c for comparison.

Figures C.4 and C.5 show predicted (blue arrows) and ground-truth (red arrows) motion vectors for the xylanase A from *Bacillus subtilis* and the periplasmic domain of Gliding motility protein GldM from *Capnocytophaga canimorsus*, respectively.

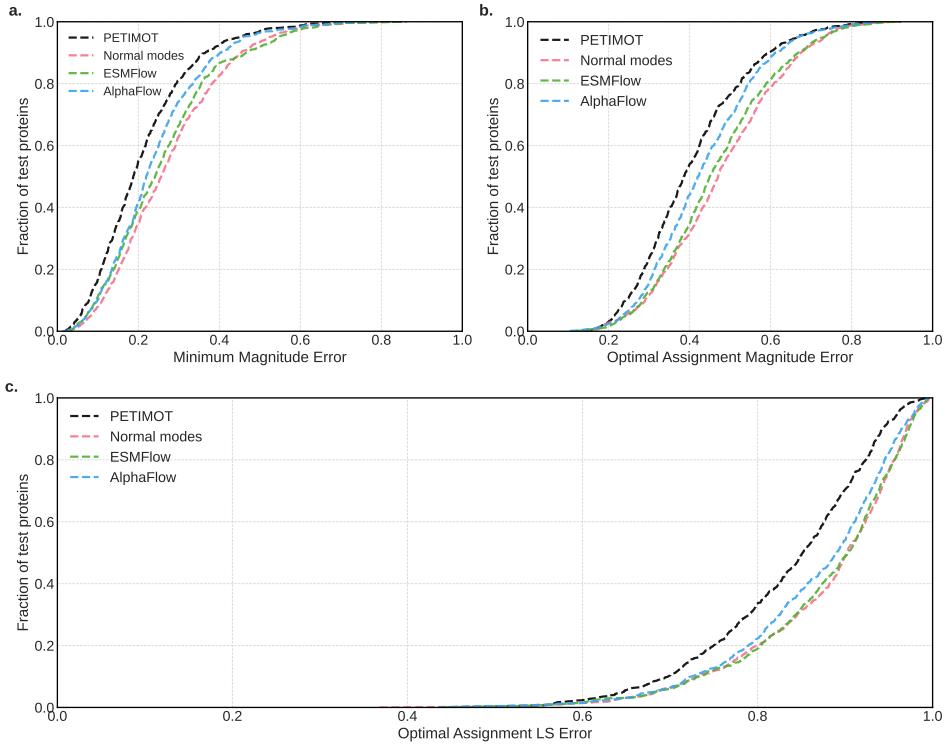

Figure C.1: **Performance comparison with other methods on the test proteins.** We report cumulative curves for magnitude error (a,b) and LS error (c). For each protein, we computed the error either for the best-matching pair of predicted and ground-truth vectors (a) or for the best combination of four pairs of predicted and ground-truth vectors (b,c).

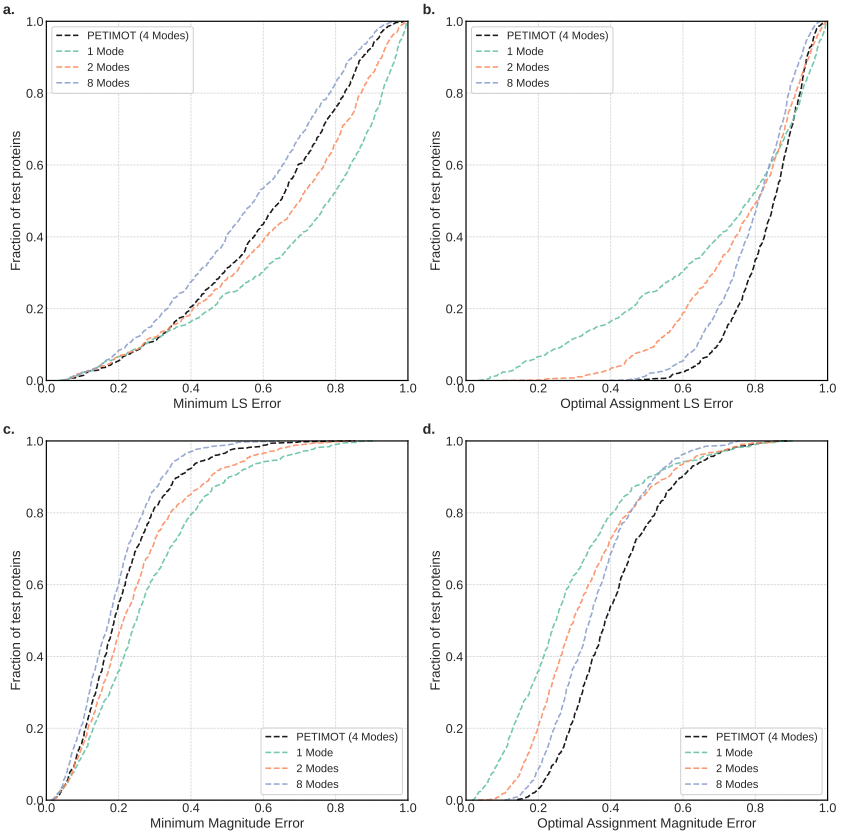

Figure C.2: **Impact of the number of predicted components.** We report cumulative curves for LS error (a-b) and magnitude error (c-d). For each protein, we computed the error either for the best-matching pair of predicted and ground-truth vectors (a,c) or for the best combination of all pairs of predicted and ground-truth vectors using optimal linear assignment (b,d). We compare models trained to predict different numbers of components (modes): 1, 2, 4, or 8, using only the LS loss.

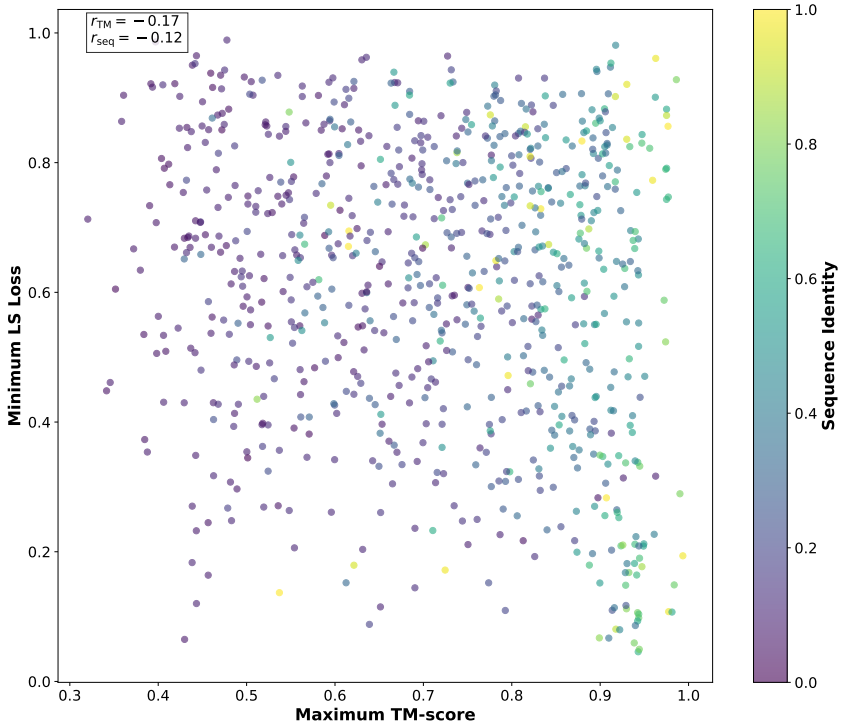

Figure C.3: **Relationship between PETIMOT's prediction accuracy and structural/sequence similarity with the training set.** The minimum LS error is plotted against the maximum TM-score between each test protein and any protein in the training set. Points are colored by the maximum sequence identity to the training samples.

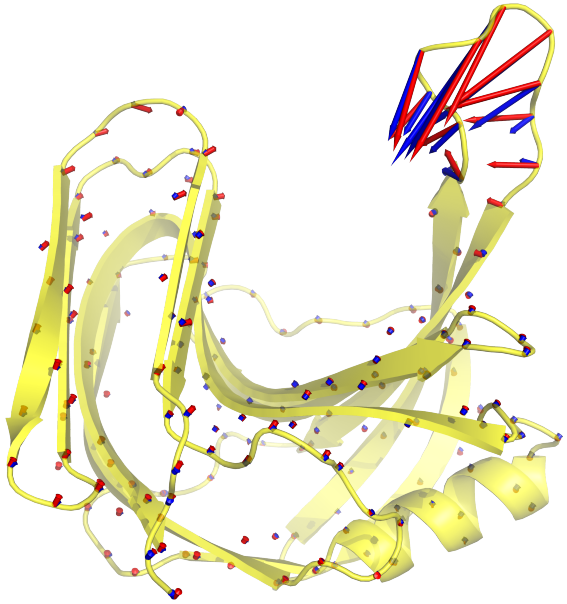

Figure C.4: **Visualization of predicted (blue arrows) and ground-truth (red arrows) motion vectors for PDB structure 3EXU (chain A), with LS error of 0.20.** The predicted deformation was used to generate the interpolated conformations shown in Fig. 3b.

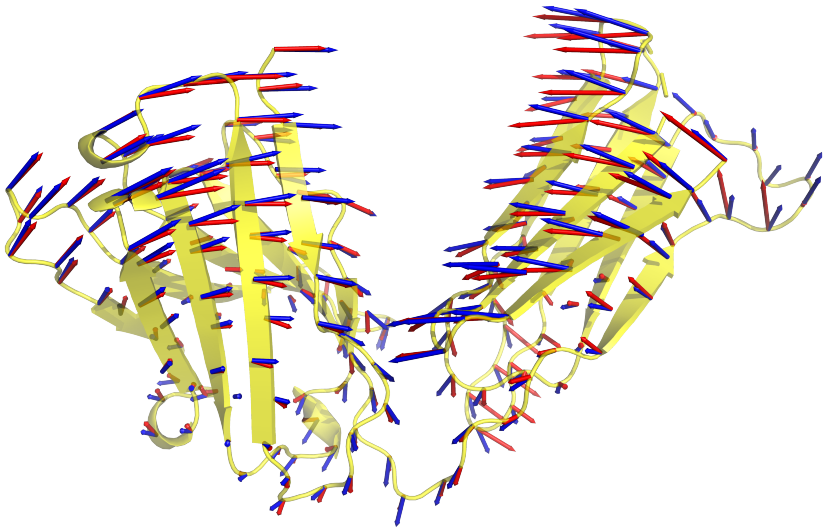

Figure C.5: **Visualization of predicted (blue arrows) and ground-truth (red arrows) motion vectors for PDB structure 7SD2, with LS error of 0.18.** The predicted deformation was used to generate the interpolated conformations shown in Fig. 3c.