DualMPNN: Harnessing Structural Alignments for High-Recovery Inverse Protein Folding

Xuhui Liao, Qiyu Wang, Zhiqiang Liang, Liwei Xiao, Junjie Chen[*]
School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
`{xuhuiliao,qiyuwang,liangzhiqiang,liweixiao}@stu.hit.edu.cn`
`junjiechen@hit.edu.cn`

Abstract

Inverse protein folding addresses the challenge of designing amino acid sequences that fold into a predetermined tertiary structure, bridging geometric and evolutionary constraints to advance protein engineering. Inspired by the pivotal role of multiple sequence alignments (MSAs) in structure prediction models like AlphaFold, we hypothesize that structural alignments can provide an informative prior for inverse folding. In this study, we introduce DualMPNN, a dual-stream message passing neural network that leverages structurally homologous templates to guide amino acid sequence design of predefined query structures. DualMPNN processes the query and template proteins via two interactive branches, coupled through alignment-aware cross-stream attention mechanisms that enable exchange of geometric and co-evolutionary signals. Comprehensive evaluations across on CATH 4.2, TS50 and T500 benchmarks demonstrate DualMPNN achieves state-of-the-art recovery rates of 65.51%, 70.99%, and 70.37%, significantly outperforming base model ProteinMPNN by 15.64%, 16.56%, 12.29%, respectively. Further template quality analysis and structural foldability assessment underscore the value of structural alignment priors for protein design.

1 Introduction

Protein inverse folding addresses the computational challenge of identifying amino acid sequences that fold into a predefined tertiary structure[1]. As the inverse of the traditional protein folding problem that predicts structures from sequences, this task is critical for advancing protein engineering[2, 3, 4], enabling broad applications in drug discovery, peptide design, synthetic biology, and enzyme design[5, 6, 7]. Despite its broad utility, protein inverse folding remains hindered by three key challenges: (1) the non-injective mapping between sequences and structures, where diverse sequences can adopt geometrically similar folds (structural degeneracy); (2) the computational intractability of exhaustively exploring the vast sequences space; (3) the difficulty of ensuring thermodynamic stability and experimental feasibility in designed sequences.

Recent advances in computational methods have made incremental progress. Energy-based methods (Rosetta[8], K Yue et al. [9]) leverage physical potentials but struggle with conformational sampling. Combinatorial optimizing methods (Craig et al.[10], Kleinberg et al.[11]) address sequence space complexity but lack scalability. Deep learning models (ProteinMPNN[12], GraDe-IF[13], ESM-IF[14]) have improved design efficiency through learned sequence-structure relationships. However, these methods often fail to resolve structural degeneracy, producing sequences that prioritize structural compatibility over functional stability. Consequently, achieving high-recovery sequence design that both fold into target structures and exhibit native-like stability remains an open challenge.

---

[*] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

A promising strategy to address degeneracy involves integrating evolutionary or structural priors into the design framework. Multiple sequence alignments (MSAs) have proven instrumental in structure prediction tools (like AlphaFold[15] and RoseTTAFold[16]), encoding evolutionary constraints that guide folding simulations. We posit that homologous structural motifs captured by structural alignments offer complementary prior knowledge for inverse folding. Unlike sequence-based MSAs, structural alignments directly encode conserved spatial and physicochemical constraints across homologs, providing a blueprint for stable sequence-structure compatibility. Preliminary experiments, in which randomly initialized ProteinMPNN's node embeddings with a certain proportions of correct and perturbed sequence templates, support this hypothesis by improved sequences recovery rates (**Supplementary Table S1**), suggesting that structural homology informs viable sequence design. In this work, we present DualMPNN, a dual-stream message passing neural network that leverages structural alignments to guide high-recovery inverse protein folding. DualMPNN jointly reasons over query structure geometry and homologous structural templates via two interactive branches. The nodes in the query branch are iteratively updated from homologous structural templates-derived amino acids through alignment-aware cross-stream attention mechanisms. Consequently, DualMPNN learns to disentangle degenerate sequence solutions while preserving stability constraints, leading to significant improvements in recovery rate of inverse protein folding. The contributions of our work are summarized as: **(1) Dual-stream architecture for structural priors.** We proposed the first framework to explicitly integrate structural alignments-derived templates into inverse protein folding via dual-stream framework, enabling simultaneous learning of target. **(2) Impact of template quality.** We systematically quantify how template selection (e.g. structural similarity) impacts sequence recovery, establishing guidelines for optimal structural alignment utilization in protein design. Structural priors derived from homologous templates can inspire new strategies for protein design. **(3) State-of-the-art performance.** Evaluations across three benchmarks (CATH, TS50 and T500) demonstrate that DualMPNN achieved state-of-the-art in sequence recovery rates, outperforming the base model ProteinMPNN by at least 12%. These results underscore its robustness and capability to leverage structural homology for high-fidelity design. The code is available at https://github.com/chen-bioinfo/DualMPNN.

2 Related work

Graph Neural Networks (GNNs) for Protein Design. GNNs have emerged as a dominant paradigm for modeling protein structures, offering an inductive bias that aligns naturally with the spatial and relational dependencies of residues. By representing proteins as graphs, where nodes encode residue-level features (amino acid embeddings) and edges capture pairwise geometric relationships (e.g. distances or angular orientations), GNNs enable direct learning of sequence-structure compatibility[17]. Early approaches, such as protein design was formulated as a conditional node classification task, predicting amino acid identities given a fixed backbone coordinates [18, 19]. Subsequent advancements introduced SE(3)-equivariant architectures[20, 21], which preserve rotational and translational invariance critical for generalizing across structural conformations. These methods excel at capturing local geometric constraints but often struggle to resolve global topological degeneracy, where distinct sequence neighborhoods map to similar structural motifs.

Diffusion Models for Sequence-Structure Co-Design. To address the one-to-many mapping inherent in inverse folding, recent work has integrated diffusion models with GNN backbones. These frameworks treat sequence design as a stochastic denoising process, iteratively refining sequences to match target structures while sampling from a learned distribution of plausible solutions. For instance, GraDe-IF[13] employs a graph diffusion framework to model residue-wise dependencies, achieving improved recovery rates in structurally ambiguous regions. Similarly, MaskDPD[22] combines masked language modeling with diffusion to probabilistically explore sequence space, demonstrating robustness to backbone perturbations. While diffusion models mitigate degeneracy by sampling diverse solutions, their reliance on purely geometric features limits their ability to incorporate evolutionary or structural priors that could further constrain the design space.

Dual-Stream Architectures for Multimodal Learning. Dual-stream neural network architectures, although primarily explored outside the protein domain, offer compelling insights into multi-modal and hierarchical representation learning. These architectures employ two parallel processing pathways to extract and fuse complementary information from distinct input modalities or feature scales. In computer vision, dual-stream models have achieved notable success. For example, DS-Net[23]

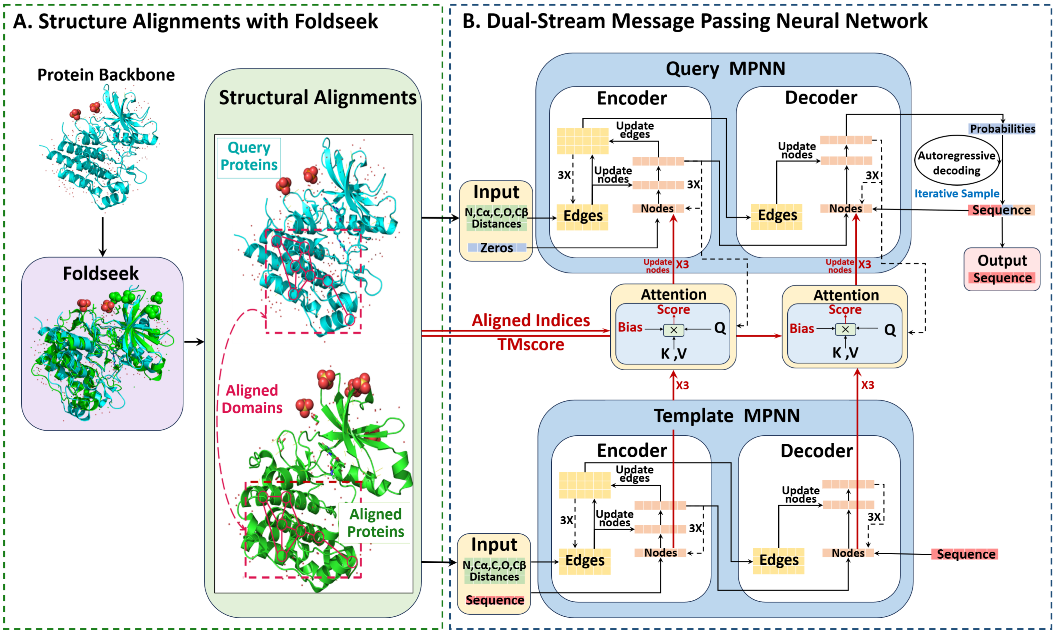


Figure 1: Illustration of DualMPNN model. DualMPNN leverages structural templates as prior knowledge to guide the inverse protein folding through dual-branch processing. Structural templates are identified using Foldseek against the PDB. Query branch processes geometric features from backbone atom coordinates to perform sequence recovery through inverse folding. Template branch leverages aligned template structures and sequences to guide the query branch's sequence recovery. These two branches share the identical MPNN components.

simultaneously captures fine-grained local patterns and global context, outperforming single-stream baselines in image classification tasks. Dual-stream Reasoning Network (DRNet)[24] designed for abstract visual reasoning demonstrates strong generalization across multiple benchmarks by jointly modeling spatial and semantic cues. Beyond vision, dual-stream architectures have been applied to sign language recognition [25] and behavioral analysis [26], showcasing their flexibility in processing heterogeneous input sources such as raw video and spatiotemporal patterns. These successes suggest that dual-stream architectures may hold promise for leveraging the rich prior information encoded in homologous structural templates.

3 Methods

We introduce DualMPNN that integrates structural alignments as prior information to guide the inverse protein folding through dual-branch processing (**Fig. 1**). Given a query structure $\mathcal{G}_q = (\mathbf{V}_q, \mathbf{E}_q)$ and a homologous structural template $\mathcal{G}_t = (\mathbf{V}_t, \mathbf{E}_t)$, where $\mathbf{V}_q$ and $\mathbf{V}_t$ represent residues, and $\mathbf{E}_q$ and $\mathbf{E}_t$ represents pairwise residue relationships, DualMPNN learns to recover the sequence s_q by jointly reasoning over $\mathcal{G}_q$ and $\mathcal{G}_t$.

3.1 Structural Templates Acquisition

Homologous templates are identified using Foldseek[27] against the Protein Data Bank (PDB). Foldseek takes one protein structure as input and returns a list of candidate proteins, sorting by their similarity. To prevent data leak from identical proteins or other chains of the same protein from being used as pairings, we exclude near-identical matches (TM-score > 0.99) as well as same PDB IDs. The remaining highest similar template is structurally aligned to query structure via TM-align [28], yielding residue correspondences $\mathbf{A} = \{(i_q, i_t)\}$ and global similarity scores $S_{TM} \in [0, 1]$. There are some certain cases of the structure alignments according to TM-scores (see **Supplementary B**).

3.2 Protein Representation

In this study, we employed ProteinMPNN as the base model. Therefore, the query representation and template representation follow the formula of ProteinMPNN. Their key difference lies in the availability of sequence information. For templates, their sequences are known, allowing the node features to be initialized using the corresponding amino acid sequences. In contrast, the query protein does not have sequence information available.

Query Representation ($\mathcal{G}_q$): Query representation is the same as the base model ProteinMPNN. Since the sequence is to be designed, node features are initialized to zeros. The edge features are pairwise distances between five backbone-derived positions:

$$\mathbf{E}_q = \left\{e_{ab}^{ij}\right\} \;\; a,b \in \{N, C_\alpha, C, O, C_\beta\}, 1 \le i,j \le L_q \;, \tag{1}$$

where a virtual C_β atom is estimated geometrically from N-C_α-C coordinates. L_q denotes the length of the template sequence.

Template Representation ($\mathcal{G}_t$): The template representation uses the same formula for edge features $\mathbf{E}_t = \left\{e_{ab}^{ij}\right\}$ to calculate pairwise relationships among residues. The amino acid sequence information is incorporated via one-hot encoding, which includes 20 standard amino acid types along with 1 additional "unknown" type. The resulting node features are represented as:

$$\mathbf{V}_t = \{\mathbf{v_i} \in \{0,1\}^{21}\}, \quad i \in \{1, \dots, L_t\}, \tag{2}$$

where L_t denotes the length of the template sequence. This integration of sequence priors enables the template branch to provide richer contextual information compared to the query branch.

Alignment Representation ($\mathcal{A}$): The query and template structures are aligned via TM-align to extract detailed domain alignment information. The alignment representation $\mathcal{A} = (\mathbf{A}(i_q, j_t), S)$ includes aligned pairs $\mathbf{A}(i_q, j_t)$ and a global measure score S. The aligned pairs $\mathbf{A}(i_q, j_t)$ is denoted:

$$\mathbf{A}(i_q, j_t) = \begin{cases} 1 & \text{matched} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where tuple (i_q, j_t) indicates the *i*-th amino acid of query protein aligned with j-th amino acid of template protein, $i_q \in \{1, \dots L_q\}$, $j_t \in \{1, \dots L_t\}$. The sets of matched nodes in query and template structures are $\mathbf{A}_{query} = \{i_q | \mathbf{A}(i_q, j_t) = 1\}$ and $\mathbf{A}_{template} = \{j_t | \mathbf{A}(i_q, j_t) = 1\}$, respectively.

3.3 DualMPNN Architecture

3.3.1 MPNN Module

The shared MPNN module employs a hierarchical encoder-decoder architecture with interleaved message passing between structural encoding and sequence decoding stages. The node and edge embeddings are computed in encoding stage: $\mathbf{h}_V = \mathbf{V}\mathbf{W}_v \in \mathbb{R}^{N \times d}$, $\mathbf{h}_E = \mathbf{E}\mathbf{W}_e \in \mathbb{R}^{N \times K \times d}$, where N denotes node number, K denotes the number of neighbors for each aggregated node, d denotes hidden dims, $\mathbf{W}_v$ and $\mathbf{W}_e$ denote learnable weights. Notably, the node features $\mathbf{V}$ are exclusively incorporated into the template branch, while the hidden states $\mathbf{h}_V$ within the query branch are initialized as zero vectors.

Structure Encoder. The MPNN encoding stage comprises three successive message-passing layers, each employing a shared architectural schema.

- ***Node Update***: (1) *Construct message function.* Collect the node features from k neighbors $\mathbf{h}_V^{neigh} = \bigoplus_{k=1}^{K} \mathbf{h}_V^k$, where $\bigoplus$ denotes stacking along the third dim and $\mathbf{h}_V^{neigh} \in \mathbb{R}^{N \times K \times d}$. Then expand the dims of $\mathbf{h}_V$ for concatenation $\tilde{\mathbf{h}_V} \in \mathbb{R}^{N \times K \times d}$. Concatenates the features $\mathbf{h}_{EV} = \mathbf{Cat}[\tilde{\mathbf{h}_\mathbf{V}}, \mathbf{h}_V^{neigh}, \mathbf{h}_E]$ and then constructs the message function $\mathbf{m} = \mathbf{MLP}(\mathbf{h}_{EV}) \in \mathbb{R}^{N \times K \times d}$. (2) *Passing step of messages*. Utilizing the aggregated message to update nodes by the following functions:

$$\Delta \mathbf{h}_V = \frac{1}{K} \sum_{k=1}^{K} m^{(:,:,k,:)} \tag{4}$$

$$\mathbf{h}_V = \mathcal{N}_2(\mathbf{h}_V + \mathcal{D}_3(score) + \mathcal{D}_2(\mathcal{F}(\mathcal{N}_1(\mathbf{h}_V + \mathcal{D}_1(\Delta \mathbf{h}_V))))) \tag{5}$$

- ***Edge Update***: (1) *Construct message function*. The message function is nearly the same as that in *node update*. The $\mathbf{h}_V$ and $\mathbf{h}_E$ used for the computation are updated in the previous node update step. (2) *Passing step of messages.* Utilizing the aggregated message to update edges by the following function:

  $$\mathbf{h}_E = \mathcal{N}_3(\mathbf{h}_V + \mathcal{D}_3(m)) \tag{6}$$

  where $\mathcal{N}(\cdot)$ denotes normalization layer, $\mathcal{D}(\cdot)$ denotes dropout layer and $\mathcal{F}(\cdot)$ denotes feedforward layer. $Score$ denotes the attention score from the interactive alignment-aware attention, see Eq. 8.

Sequence Decoder. The MPNN decoder comprises three hierarchical decoding layers, each ingesting node-edge feature pairs from the encoder outputs while incorporating the native protein sequence embeddings to perform autoregressive sequence recovery.

- ***Autoregressive decoding.*** The sequence decoder takes both sequence embeddings and node embeddings as input and generates outputs through chain-rule factorization:

  $$p_\theta(s|v,e) = \prod_{l=1}^{L} p_\theta(s_l|s_{<l}, v_{>l}, e) \tag{7}$$

  where l denotes the l-th node that is decoding, each generated element s_l depends on the latent node $v_{>l}$, edge states e and previously generated elements $s_{<l}$. The sequence will be decoded sequentially. For each node i, we consider the following:

  (1) *For node features*: $\mathbf{h}_{V_i}^{neigh} = \bigoplus_{k=i+1}^{K} h_{V_i}^{k}$, $\mathbf{h}_{EV} = \mathbf{Cat}[\tilde{\mathbf{h}_\mathbf{V}}; \mathbf{h}_V^{neigh}; \mathbf{h}_E]$, where node i aggregates the next up latent node features $h_{V_i}^{>i}$ to update.

  (2) *For sequence embeddings*: $\mathbf{h}_{S_i}^{neigh} = \bigoplus_{k=1}^{i} h_{S_i}^{k}$, $\mathbf{h}_{ES} = \mathbf{Cat}[\tilde{\mathbf{h}_\mathbf{S}}; \mathbf{h}_S^{neigh}; \mathbf{h}_E]$, where node i aggregates the previous sequence features $h_{S_i}^{<i}$ to update.

  (3) *Message construction and passing*: $\mathbf{h}_{ESV} = \mathbf{Cat}[h_{EV}; h_{ES}]$, $\mathbf{m_S} = \mathbf{MLP}(h_{ESV})$, $\Delta \mathbf{h}_V = \frac{1}{K} \sum_{k=1}^{K} m_S^{(:,:,k,:)}$, $\mathbf{h}_V = \mathcal{N}_2(\mathbf{h}_V + \mathcal{D}_3(score) + \mathcal{D}_2(\mathcal{F}(\mathcal{N}_1(\mathbf{h}_V + \mathcal{D}_1(\Delta \mathbf{h}_V)))))$, where $[\cdot, \cdot]$ represents row-wise concatenation, $\mathcal{N}(\cdot)$ denotes normalization layer, $\mathcal{D}(\cdot)$ denotes dropout layer and $\mathcal{F}(\cdot)$ denotes feedforward layer, $score$ denotes the attention score from the interactive alignment-aware attention, see Eq. 8.

- ***Iterative Sampling.*** During inference, the decoder employs autoregressive sampling to iteratively generate sequence embeddings. At step t, it synthesizes a subsequence s_t conditioned on historical outputs $\{s_{1:t-1}\}$ and predicted node features $h_{t:K}$ from latent projections.

3.3.2 Interactive Attention Layer

The DualMPNN architecture employs a cross-modal attention layer to enable synergistic information exchange between the query and template branches. The interaction leverages structural alignment-guided attention to fuse query and template features. Let $h_V \in \mathbb{R}^{L \times d}$ and $h_V^t \in \mathbb{R}^{L^t \times d}$ denote node embeddings from the *query and template MPNN* encoders or decoders, respectively. The query-key-value projections are computed as: $\mathbf{Q} = \mathcal{W}_1 \cdot h_V^t[\mathbf{A}_{query}]$, $\mathbf{K} = \mathcal{W}_2 \cdot h_V^t[\mathbf{A}_{template}]$, $\mathbf{V} = \mathcal{W}_3 \cdot h_V^t[\mathbf{A}_{template}]$, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L^A \times d}$ share the same shapes, L^A is the length of aligned residue pairs; $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3 \in \mathbb{R}^{d \times d}$ are learnable projection matrices; $\mathbf{A}_{query} \in \mathbb{N}^{L^A}$ and

$\mathbf{A}_{template} \in \mathbb{N}^{L^A}$ are the aligned residue indices extracted from the TM-align output that controls the attention area between the query and template branch.

The attention score is calculated and utilized by the following functions:

$$\mathbf{score} = \sigma(\frac{Q \times K}{\sqrt{d}}) \times V \times S \tag{8}$$

$$h_V[\mathbf{A}_{query}] = \mathcal{N}(h_V[\mathbf{A}_{query}] + \mathcal{D}(\mathbf{score})) \tag{9}$$

where $S \in [0, 1]$ is the TM-score between query and template proteins. $\mathcal{N}(\cdot), \mathcal{D}(\cdot), \sigma(\cdot)$ denote normalization, dropout, and softmax layers, respectively.

Interactive attention will be applied within every encoder and decoder layer. This mechanism enables prior information guidance between the two branches at each layer, allowing the model to refine its internal projections in a context-aware manner. Structure alignment-guided sequence recovery provides a bridge between protein structure and sequence space. It enhances our ability to recover protein sequences with aligned structures by incorporating evolutionary and structural constraints.

4 Experiments

4.1 Experimental Protocol

Dataset. We trained and evaluated DualMPNN on CATH, following the standardized data partition from prior work GraphTrans [17] and GraDe-IF [13]. The proteins are categorized into a division of 18,024 proteins for training, 608 for validation, and 1,120 for testing. The model is tested in 3 different categories: short, single-chain, and all proteins. The short chain category is those with sequence lengths shorter than 100. The single-chain category contains only those models in which the single chain accounted for the entire protein record in the PDB. Additionally, we tested our model on the T500 and TS50 datasets introduced by DenseCPD[29], which includes 9,888 structures for training and two distinct test datasets containing 50 (TS50) and 500 (T500) structures, respectively.

Training setup. The MPNN blocks possess a hidden dimension of 128 for the node and edge projections. The number of neighbors for each aggregated node is 48 in the *query MPNN* and 4 in the *template MPNN*. The interactive attention layer shares the same hidden dimension as the MPNN block. In addition, we utilize a dropout rate of 0.1 to avoid overfitting both in the MPNN block and in the attention layer. The model is trained on 40 epochs, and the learning rate is scheduled by the Adam optimizer.

Generation of Structure Alignments. We utilize Foldseek[27] to perform multiple structural alignments for a given query protein. The search mode is "easy-search". The alignment-type is "TM alignment". The prefilter-mode is "nofilter". Other arguments are default. After generating protein candidates, We filter them by removing the same protein and those with TM-scores greater than 0.99 to prevent data leak from identical proteins.

Evaluation Metric. The quality of recovered protein sequences is evaluated using two key metrics: *perplexity* and *recovery rate*. *Perplexity* quantifies the alignment between the model's predicted amino acid probability distribution and the actual residues observed at each sequence position, with lower values indicating superior model-data compatibility. The *recovery rate* measures predictive accuracy by calculating the percentage of amino acids in the reconstructed sequence that correctly match the native protein sequence, with higher values reflecting enhanced sequence reconstruction capability from structural inputs. The quality of protein structure alignment is quantified by *TM-score* and *RMSD*. *TM-score* employs a length-normalized assessment that evaluates both local structural matches and global topological similarity, producing values ranging from 0 to 1 , with scores above 0.5 generally indicating biologically meaningful structural relationships. *RMSD* measures the average spatial deviation between corresponding C_α atoms in superimposed structures, with lower values indicating stronger geometric congruence.

4.2 Inverse Folding

Table 1 shows the results of the different structure-aware models in the CATH, T500 and TS50 datasets. DualMPNN achieves state-of-the-art performance across all metrics, demonstrating superior sequence recovery capabilities and lower perplexity scores. Specifically, our model reaches a recovery

Table 1: Comparison of recovery rate and perplexity performance on **CATH**, **TS50**, and **T500**. Models marked with † use CATH v4.3, the rest use CATH v4.2. **PPL** denotes perplexity, **Rec.** denotes recovery rate.

Models	**CATH**						**TS50**		**T500**	
	PPL ↓			**Rec. % ↑**			**PPL ↓**	**Rec.% ↑**	**PPL ↓**	**Rec.% ↑**
	Short	Single	All	Short	Single	All				
STRUCTGNN [17]	8.29	8.74	6.40	29.44	28.26	35.91	5.40	43.89	4.98	45.69
GRAPHTRANS [17]	8.39	8.83	6.63	28.14	28.46	35.82	5.60	42.20	5.16	44.66
GCA [30]	7.09	7.49	6.05	32.62	31.10	37.64	5.09	47.02	4.72	47.74
GVP [31]	7.23	7.84	5.36	30.60	28.95	39.47	4.71	44.14	4.20	49.14
GVP-LARGE [14] †	7.68	6.12	6.17	32.60	39.40	39.20	–	–	–	–
ALPHADESIGN [32]	7.32	7.63	6.30	34.16	32.66	41.31	5.25	48.36	4.93	49.23
ESM-IF1 [14] †	8.18	6.33	6.44	31.30	38.50	38.30	–	–	–	–
PROTEINMPNN [12]	6.21	6.68	4.57	36.35	34.43	49.87	3.93	54.43	3.53	58.08
PIFOLD [33]	6.04	6.31	4.55	39.84	38.53	51.66	3.86	58.72	3.44	60.42
GRADE-IF [13]	5.49	6.21	4.35	45.27	42.77	52.21	3.71	56.32	3.23	61.22
DualMPNN	**4.42**	**5.04**	**3.18**	**55.97**	**52.41**	**65.51**	**2.76**	**70.99**	**2.71**	**70.37**

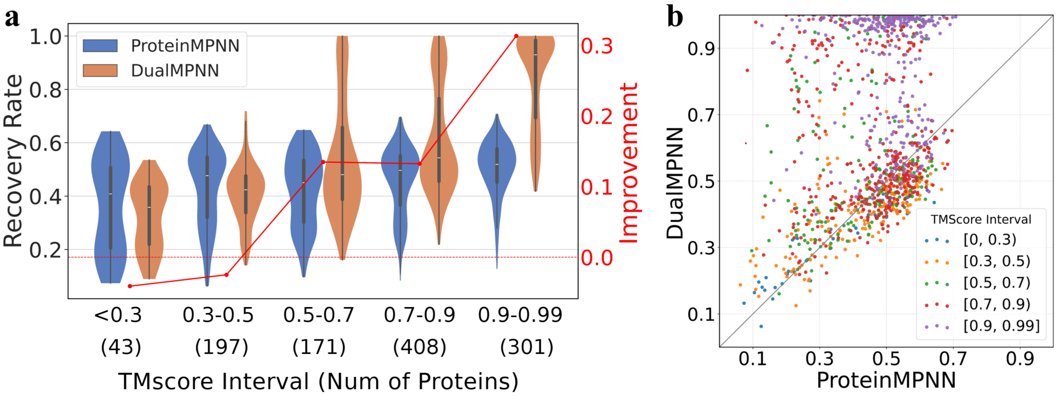


Figure 2: Impact of template quality on sequence recovery rate. (**a**) Violin plot comparing sequence recovery rates of DualMPNN and ProteinMPNN across distinct TM-score intervals on the CATH test set. (**b**) Scatter plot of per-protein recovery rates, colored by TM-score intervals. Points above the dashed parity line ($y = x$) highlight instances where DualMPNN outperforms the baseline, particularly for high-quality templates (TM-score > 0.5).

rate of 65.51% and a perplexity of 3.19. The recovery rate improved 13.3% compared to GraDe-IF. Since most of the data from CATH consists of oligomers or multimers, the chains within these assemblies are significantly influenced by interchain interactions[34]. In multichain proteins, such interactions induce context-dependent conformational plasticity, causing identical sequences to adopt distinct structural geometries in oligomeric assemblies compared to their monomeric states. This occurs through cooperative adjustments of backbone dihedrals, side-chain rotamers, and quaternary-stabilized folding motifs. Therefore, single chains are particularly challenging to predict in the CATH dataset. However, DualMPNN leverages direct guidance from prior structural information, enabling it more accurately to predict single-chain structures, reaching a recovery rate of 52.41% and a perplexity of 5.04. Short chains also face the challenge of a lack of similar data within the dataset, which makes accurate prediction more difficult. Despite this limitation, DualMPNN demonstrates its robustness by achieving an impressive recovery rate of 55.97%, showcasing its ability to handle even the most data-scarce scenarios effectively. Furthermore, the results in T500 and TS50 datasets (**Table 1**) also show that DualMPNN remains the best performance across all metics, reaching a recovery rate of 70.37% , 70.99% and a perplexity of 2.71, 2.76 in T500, TS50, respectively.

4.3 Template Quality Matters

To evaluate the impact of template quality, we divided the CATH test dataset based on their TM-scores against the corresponding template proteins into five categories: $[0, 0.3), [0.3, 0.5), [0.5, 0.7), [0.7, 0.9), [0.9, 0.99]$. Among the alignments, 880 proteins had a TM score greater than 0.5, while 240 proteins had a TM score less than 0.5. We then compared the performance improvements of DualMPNN against ProteinMPNN within these categories. As shown in **Fig. 2(a)**, DualMPNN demonstrates significant advantages over ProteinMPNN when the template quality exceeds a TM score of 0.5, with improvements becoming more pronounced as the quality of the structural alignment increases. Specifically, DualMPNN achieves a recovery rate of 58.7%, 61.5% and 84.0%, outperforming the baseline model ProteinMPNN by 13.4%, 13.5% and 31.3%, within the TM score interval of [0.5,0.7), [0.7,0.9) and [0.9,0.99], respectively.

Furthermore, we compared the recovery rate performance of a certain protein between ProteinMPNN and DualMPNN (**Fig. 2(b)**). The diagonal line indicates equal performance between the two models. Points above the diagonal represent cases where DualMPNN outperforms ProteinMPNN, while points below the diagonal represent cases where ProteinMPNN performs better. We observed that ProteinMPNN rarely achieves a recovery rate above 70%, while DualMPNN can easily recover certain protein sequences with a recovery rate above 70%, and in some cases, even close to 100%.

These results highlight DualMPNN's superior ability to leverage high-quality templates for accurate sequence recovery. However, our model does have some limitations. For templates with low quality, it remains challenging to significantly improve sequence recovery performance. Fortunately, with the help of Foldseek, we can identify a large number of high-quality templates through structural alignments, which ensures that the approach remains practical and effective in real-world applications.

4.4 Ablation Study

To evaluate the contributions of the components in DualMPNN, we conducted an ablation study (**Table 2**). Starting with ProteinMPNN, we incorporated aligned domain node initialization using template proteins. By leveraging the prior knowledge of template proteins, the recovery rate improved significantly, increasing from 49.87% to 61.29%. Next, we utilized a dual-stream protein message passing neural network to update node projections within the encoding and decoding layers, where nodes were directly updated by added features. Additionally, we used interactive attention layers to refine the node features, enabling the model to sense contextual information and enhance protein sequence recovery. The interactive attention layers contributed significantly, increasing the recovery rate from 62.13% to 64.78% and reducing perplexity from 3.48 to 3.23. Finally, by applying the TM score as a similarity bias between the template and query proteins, DualMPNN achieved a recovery rate of 65.35% and maintained a perplexity of 3.20. This demonstrates the effectiveness of each component in improving sequence recovery and model performance. Furthermore, we discovered that sampling 10 times at the inference stage with different random seeds and using the most frequent amino acid type as the answer further improved the recovery rate to 65.51% and reduced the perplexity to 3.18.

Table 2: Ablation study on different components of DualMPNN on CATH. **PPL** denotes perplexity, **Rec.** denotes recovery rate.

	Configuration	**PPL** ↓			**Rec.** % ↑		
		Short	Single-chain	All	Short	Single-chain	All
A	Baseline Model	6.21	6.68	4.57	36.35	34.43	49.87
B	**A** + Node init by template	5.35	5.83	3.55	50.85	47.03	61.29
C	**B** + Dual-stream update	4.82	5.54	3.48	53.78	50.23	62.13
D	**C** + Interactive attention update	4.57	5.23	3.29	54.95	51.11	64.78
E	**D** + TM score bias	4.46	5.09	3.20	55.74	52.19	65.35
F	**E** + Sample 10 times	4.42	5.04	3.18	55.97	52.41	65.52

4.5 Generalization

To assess generalization on novel samples, we stratified test proteins by structural similarity (TM-score) to the training set. As summarized in **Table 3**, DualMPNN is substantially better than

ProteinMPNN, especially at low similarity. For TM-score < 0.3, DualMPNN attains 66.9% recovery vs. 50.9% for ProteinMPNN; for 0.3–0.5, 60.5% vs. 45.3%; and for 0.5–0.7, 58.6% vs. 40.9%. These marginal gains indicate robustness beyond close structural matches and highlight superior performance on structurally novel cases, indicating that DualMPNN does not simply rely on near-neighbor memorization but scales across the full novelty spectrum.

Table 3: Comparison of generalization in terms of structural similarity (TM-score) or sequence identity stratification.

Generalization Benchmarks	**TM-score (Test vs Train)**					**Sequence Identity (Test vs Template)**				
	<0.3	**0.3–0.5**	**0.5–0.7**	**0.7–0.9**	**0.9–0.99**	**<0.3**	**0.3–0.5**	**0.5–0.7**	**0.7–0.9**	**0.9–0.99**
# Samples	**706**	**307**	**73**	**29**	**5**	**803**	**153**	**79**	**53**	**32**
ProteinMPNN	50.9	45.3	40.9	45.1	48.1	48.5	52.4	48.0	50.7	51.8
DualMPNN	66.9	60.5	58.6	74.5	71.6	60.6	68.4	83.8	87.1	95.5

To validate the model generalization on low-quality templates, we binned test samples by sequence identity to templates and compared recovery across five identity ranges. DualMPNN exhibits strong performance in all bins, achieving 60.6% recovery even sequence identity <0.3. While recovery rate increases with template quality (higher identity), DualMPNN shows a consistently larger improvement margin. This indicates it more effectively leverages higher-quality template signals without sacrificing performance when they are weak. These results are corroborated by TM-score analysis, confirming gains are consistent across both evolutionary (sequence) and structural (TM-score) notions of divergence. In addition, lower perplexity (where Section 4.2 reported) aligns with higher recovery on both the test and holdout splits, suggesting better-calibrated sequence distributions that translate into improved top-1 design accuracy without sacrificing stability.

4.6 Foldability

To validate the foldability of the generated protein sequences, we fold them with AlphaFold2[15] and Alphafold3[35] then align the folded structures with the native structures using TM-align to compare their similarity. Following GraDe-IF[13], we evaluate each novel sequence by the TM-score between its predicted structure and the native structure, where a TM-score above 0.5 indicates a successful design. We fold 100 generated sequences from the CATH test set (first 100 sequences ordered alphabetically by PDB ID), and summarize the results in **Table 4**.

Under Alphafold2 inference, DualMPNN attains a success rate of 94%, a mean TM-score of approximately 0.86, an average pLDDT near 0.91, and an RMSD around 1.5 Å. These results indicate that the sequences produced by DualMPNN not only fold into well-defined structures but also receive high model confidence. Native sequences processed by AF2 exhibit an average pLDDT of about 0.91, which underscores that the confidence of our generated sequences approaches that of the native counterparts. When evaluated with Alphafold3, DualMPNN reaches a success rate of 95%, a mean TM-score close to 0.87, an average pLDDT near 0.92, and RMSD values on the order of 1.4 Å. Relative to ProteinMPNN under AF3, DualMPNN delivers higher success, stronger structural similarity, and higher confidence while maintaining comparable geometric deviation. Collectively, these results demonstrate that DualMPNN produces sequences that fold reliably, attain high structural agreement with native targets, and are assigned consistently high confidence by modern structure predictors.

Table 4: Foldability comparison between generated structures and the native structures. The methods with † are generated by Alphafold3 and the rest using Alphafold2.

Method	Success↑	TM score ↑	avg pLDDT ↑	avg RMSD ↓
PiFOLD	85	0.80 ± 0.22	0.84 ± 0.15	1.67 ± 0.99
ProteinMPNN	94	0.86 ± 0.16	0.89 ± 0.10	**1.36 ± 0.81**
GRaDe-IF	94	0.86 ± 0.17	0.86 ± 0.08	1.47 ± 0.82
DualMPNN	94	0.86 ± 0.16	0.91 ± 0.10	1.49 ± 0.86
ProteinMPNN †	94	0.86 ± 0.18	0.88 ± 0.12	1.41 ± 0.76
DualMPNN †	**95**	**0.87 ± 0.16**	**0.92 ± 0.11**	1.39 ± 0.80

Recovery rate: 0.53
TM score: 0.96
Avg. pLDDT: 92.6
RMSD: 1.10

Recovery rate: 0.69
TM score: 0.97
Avg. pLDDT: 95.8
RMSD: 0.92

Recovery rate: 0.88
TM score: 0.95
Avg. pLDDT: 93.3
RMSD: 1.52

Figure 3: Folding comparison of generated sequences (blue) and native sequences (orange).

Specific cases of folded sequences, selected from the CATH test set, are presented in **Fig. 3**. The three folded structures (3GA2, 3GOC, and 3HKV) exhibit sequence recovery rates of 0.53, 0.69, and 0.88, respectively. All achieved TM scores greater than 0.95 and average pLDDT scores exceeding 0.92, which are comparable to the native sequences. The generated structures are well-aligned with their native counterparts, highlighting the exceptional foldability of DualMPNN.

5 Conclusion

We introduce DualMPNN, a dual-stream message-passing neural network that synergizes geometric and co-evolutionary signals from query structures and their structurally aligned homologs. DualMPNN is a novel paradigm that leverages structural alignments as informative priors for inverse protein folding. The architecture comprises two distinct branches, which interact through alignment-aware cross-attention mechanisms, enabling feature enhancement while maintaining weight independence. Extensive empirical validation demonstrates DualMPNN's state-of-the-art performance, surpassing existing baselines by significant margins. Structural validity assessments confirm the biological plausibility of designed sequences, with AlphaFold2-predicted structures exhibiting high confidence and geometric fidelity to native backbones. Crucially, we identify a TM-score threshold of 0.5 as a critical determinant of template utility, beyond which structural homology significantly enhances sequence recovery.

Our work underscores the transformative potential of structural alignment priors in protein design, bridging geometric constraints with evolutionary insights to advance sequence generation accuracy. By circumventing reliance on explicit sequence covariation data, DualMPNN offers a scalable framework for *de novo* protein engineering, with implications for designing functional proteins in low-MSA regimes. Future directions include integrating dynamic template selection strategies and extending the framework to multi-state protein design. This approach not only advances computational protein engineering but also deepens our understanding of the structure-sequence interplay in biology.

6 Acknowledgments

This work is supported by the National Natural Science Foundation of China (62573164), Guangdong Basic and Applied Basic Research Foundation (2025A1515010185), Guangdong Provincial Colleges and Universities Key Areas Program (2025ZDZX2079), Key Project of Heilongjiang Provincial Natural Science Joint Foundation (ZL2025E003), Shenzhen Science and Technology Program (JCYJ20230807094318038, KQTD20240729102154066, KCXFZ20230731093001002).

References

[1] George A Khoury, James Smadbeck, Chris A Kieslich, and Christodoulos A Floudas. Protein folding and de novo protein design for biotechnological applications. *Trends in biotechnology*, 32(2):99–109, 2014.

[2] Zhourun Wu, Mingyue Guo, Xiaopeng Jin, Junjie Chen, and Bin Liu. Cfago: cross-fusion of network and attributes based on attention mechanism for protein function prediction. *Bioinformatics*, 39(3):btad123, 2023.

[3] Jun Zhang, Hao Zeng, Junjie Chen, and Zexuan Zhu. Inab: identify nucleic acid binding domain via cross-modal protein language models and multiscale computation. *Briefings in Bioinformatics*, 26(5):bbaf509, 2025.

[4] Xianliang Liu, Jiawei Luo, Xinyan Wang, Yang Zhang, and Junjie Chen. Directed evolution of antimicrobial peptides using multi-objective zeroth-order optimization. *Briefings in Bioinformatics*, 26(1), 2024.

[5] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.

[6] Ke Yan, Shutao Chen, Bin Liu, and Hao Wu. Accurate prediction of toxicity peptide and its function using multi-view tensor learning and latent semantic learning framework. *Bioinformatics*, 41(9):btaf489, 2025.

[7] Ke Yan, Hongwu Lv, Jiangyi Shao, Shutao Chen, and Bin Liu. Tppred-sc: multi-functional therapeutic peptide prediction based on multi-label supervised contrastive learning. *Science China Information Sciences*, 67(11):212105, 2024.

[8] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

[9] Kaizhi Yue and Ken A Dill. Inverse protein folding problem: designing polymer sequences. *Proceedings of the National Academy of Sciences*, 89(9):4163–4167, 1992.

[10] Roger A Craig, Jin Lu, Jinquan Luo, Lei Shi, and Li Liao. Optimizing nucleotide sequence ensembles for combinatorial protein libraries using a genetic algorithm. *Nucleic acids research*, 38(2):e10–e10, 2010.

[11] Jon M Kleinberg. Efficient algorithms for protein sequence design and the analysis of certain evolutionary fitness landscapes. In *Proceedings of the third annual international conference on Computational molecular biology*, pages 226–237, 1999.

[12] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[13] Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Liò, and Yuguang Wang. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36:10238–10257, 2023.

[14] 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 *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

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

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

[17] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[18] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

[19] Jiawei Luo, Kejuan Zhao, Junjie Chen, Caihua Yang, Fuchuan Qu, Yumeng Liu, Xiaopeng Jin, Ke Yan, Yang Zhang, and Bin Liu. imfp-lg: Identify novel multi-functional peptides using protein language models and graph-based deep learning. *Genomics, Proteomics & Bioinformatics*, 22(6):qzae084, 2024.

[20] Yang Tan, Bingxin Zhou, Yuanhong Jiang, Yu Guang Wang, and Liang Hong. Multi-level protein representation learning for blind mutational effect prediction. *arXiv preprint arXiv:2306.04899*, 2023.

[21] Fuchuan Qu, Yijin Zhao, Tao Huang, Xinyan Wang, Jun Zhang, and Junjie Chen. Hiphd: Hierarchical classification for protein remote homology detection by incorporating protein sequential and structural information. *IEEE Transactions on Computational Biology and Bioinformatics*, 2025.

[22] Peizhen Bai, Filip Miljković, Xianyuan Liu, Leonardo De Maria, Rebecca Croasdale-Wood, Owen Rackham, and Haiping Lu. Mask-prior-guided denoising diffusion improves inverse protein folding. *Nature Machine Intelligence*, pages 1–13, 2025.

[23] Mingyuan Mao, Renrui Zhang, Honghui Zheng, Teli Ma, Yan Peng, Errui Ding, Baochang Zhang, Shumin Han, et al. Dual-stream network for visual recognition. *Advances in Neural Information Processing Systems*, 34:25346–25358, 2021.

[24] Kai Zhao, Chang Xu, and Bailu Si. Learning visual abstract reasoning through dual-stream networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16979–16988, 2024.

[25] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. *Advances in Neural Information Processing Systems*, 35:17043–17056, 2022.

[26] Ezechukwu Israel Nwokedi, Rasneer Sonia Bains, Luc Bidaut, Xujiong Ye, Sara Wells, and James M Brown. Dual-stream spatiotemporal networks with feature sharing for monitoring animals in the home cage. *Sensors*, 23(23):9532, 2023.

[27] Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

[28] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[29] Yifei Qi and John ZH Zhang. Densecpd: improving the accuracy of neural-network-based computational protein sequence design with densenet. *Journal of chemical information and modeling*, 60(3):1245–1252, 2020.

[30] Cheng Tan, Zhangyang Gao, Jun Xia, Bozhen Hu, and Stan Z Li. Generative de novo protein design with global context. *arXiv preprint arXiv:2204.10673*, 2022.

[31] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

[32] Zhangyang Gao, Cheng Tan, and Stan Z Li. Alphadesign: A graph protein design method and benchmark on alphafolddb. *arXiv preprint arXiv:2202.01079*, 2022.

[33] Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

[34] Tobias Sikosek and Hue Sun Chan. Biophysics of protein evolution and evolutionary protein biophysics. *Journal of The Royal Society Interface*, 11(100):20140419, 2014.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The results of our experiments prove that.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mentioned these in Section 4.3

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided model details in Section 3.2 and training details in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code and data will be released upon acceptance

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provided this information in Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provided the standard deviation in the foldability experiment, and a violin plot of the sequence recovery rate distribution.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This is a computational technical work which does not conduct realistic biological application.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We didn't introduce new datasets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Supplementary Files

A Preliminary Experiments

Before this work begins, we've performed some preliminary experiments to evaluate the feasibility. Our initial hypothesis is that the node information within the graph network constructed by ProteinMPNN can significantly enhance model performance when enriched with prior knowledge derived from sequences corresponding to homologous structures. In ProteinMPNN, these node features are originally initialized to zero, leaving substantial room for improvement by incorporating more informative priors. Theoretically, augmenting this initialization with sequence context from homologous structures should effectively improve the model's accuracy by providing a richer representation of the underlying protein features.

On the CATH dataset, we initialized a certain proportion of the ProteinMPNN node features with the native protein sequences, while another certain proportion was initialized with incorrect sequences. This setup aimed to simulate the characteristics of amino acids in homologous structural pairings, where some regions share identical sequences while others contain mismatches. The purpose of this approach was to better mimic real-world scenarios where structural alignments provide partial but imperfect sequence information. The preliminary experimental results are shown in **Table S1**. When the wrong initialization rate is fixed, increasing the correct initialization rate improves recovery. For example, with 0% wrong rate, recovery rises from 49.87% (0% correct) to 54.57% (10%) and 61.39% (30%). At a fixed correct rate, higher wrong initialization lowers recovery. With 10% correct, recovery drops from 53.42% (20% wrong) to 51.82% (40%); with 20% correct, it drops from 58.67% (10%) to 52.61% (60%). Both correct and wrong initialization rates markedly affect recovery, underscoring the importance of template quality. This is consistent with the analysis in *Section 4.3*.

Table S1: Sequence perplexity and recovery rate under different initializing ratio in preliminary experiments.

Init ratio			**Perplexity↓**	**Recovery rate↑**
Correct	Wrong	Zeros		
0%	0%	100%	5.41	49.87%
10%	0%	90%	4.09	54.57%
10%	20%	70%	4.12	53.42%
10%	40%	50%	4.73	51.82%
20%	10%	70%	4.53	58.67%
20%	60%	20%	4.62	52.61%
30%	0%	70%	3.77	61.39%
30%	20%	50%	3.85	60.01%
40%	40%	20%	3.57	62.11%

B Cases of Protein Templates in Five TM-score Intervals

Protein templates are divided into five intervals: $[0, 0.3), [0.3, 0.5), [0.5, 0.7), [0.7, 0.9), [0.9, 0.99]$. There are some cases of query-template proteins down below (**Fig. S1**), each presents a combined view of sequence and structural alignment between a query protein and a template protein, typically used to assess homology modeling or structural similarity. On the left, the sequence alignments highlight the correspondence between residues in the query and template, with matched or similar residues indicated by colored symbols, where exact matches often shown as aligned letters, and conservative substitutions marked with symbols such as plus signs. Gaps introduced to optimize alignment are represented by dashes. The sequence alignments are annotated with residue indices to indicate positional context. On the right, ribbon diagrams overlay the 3D structures of the two proteins—commonly using distinct colors (blue for the query and orange for the template) to visually demonstrate structural conservation. Quantitative metrics such as TM-score, RMSD, aligned length, and matched length are usually provided to objectively evaluate the quality of the alignment, with higher TM-scores and lower RMSD values indicating greater structural similarity.

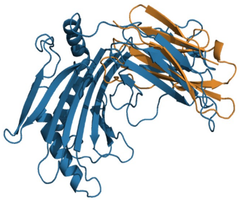
These template cases illustrate the structure and sequence similarities between proteins across different TM-score intervals. When the TM-score is below 0.5, the alignment tends to be short, and the type of aligned amino acid shows low similarity. In contrast, when the TM-score exceeds 0.5, the alignment length increases, and the residue-level correspondence improves significantly. These examples support the findings presented in *Section 4.3* of the experiments, further highlighting the critical role of template quality in influencing the accuracy of sequence recovery by the model.

Query: 3IT8.D   Template: 6DDV.C

TM-score=0.27, RMSD=2.65, Aligned length=91, Matched length=21

```
Q 182 TTPTVKVTGNELEDGNMTLECTVNSFYPPDVITKWIESEHFKGEYKYVNGRYYPEWGRKSNYEPGEPGFPWNIKKDKDAN 261
        P V VT +E  +GN+T+ C  +SFYP ++I  W +                                 W
T   2 VPPMVNVTRSEASEGNITVTCRASSFYPRNIILTWRQD----------------------GVSLSHDTQQWGDVLPDGNG 59

Q 262 TYSLTDLVRTTSKMSSQPVCVVFHDTLEAQVY 294
      TY      R +     +  C + H +     +
T  60 TYQTWVATRISRGEEQRFTCYMEH-SGNHSTH 92
```

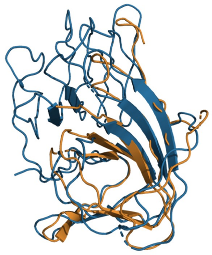


Query: 3MYX.A   Template: 3BCW.A

TM-score=0.43, RMSD=2.13, Aligned length=111, Matched length=32

```
Q 117 PDKSGITALDRLALLTPSSPPDPS-IISPLPQCRSNNLFEDTASTLRIGVWDSTPYERISRPHK--IHELNLIEGRVVLS 193
       DKS +  +D   ++ P     PS  I+     R+   FE     +  GVW+ST            I   ++IEG   L
T   1 HDKSRLVRIDTGPMINP-VAGKPSRPIAGDASFRTVTAFEGGQGKVESGVWESTSGSF-QSNTTGYIEYCHIIEGEARLV 78

Q 194 LENGSSLTVNTGDTVFVAQGAPCKWTSTGYVRKFYAVT 231
        +G+   V  GD   + +G   +W    +V+K Y VT
T  79 DPDGTVHAVKAGDAFIMPEGYTGRWEVDRHVKKIYFVT 115
```

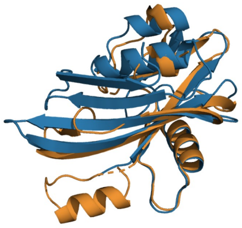


Query: 3FK9.A   Template: 3SMD.A

TM-score=0.63, RMSD=2.19, Aligned length=108, Matched length=38

```
Q   2 QRVTNCIVVDH-DQVLLLQKPRRGWWVAPGGKMEAGESILETVKREYWEETG--ITVKNPELKGIFSM----VIFDEGKI 74
            ++ +   ++L+ Q P   +W  P+G +E GE+  E V RE WEETG  + VK    KG+F        +  G
T  19 MPSVAAVIKNEQGELLF-QYP---YWSLPAGAIEPGETPEEAVIREVWEETGLKVQVKK--QKGVFGGKEFRYTYANGDK 92

Q  75 VSEWMLFTFKATEHEGEMLKQSPEGKLEWKKKDE--VLELPMAAGDKWIF 122
      V E+++  F+     G++ K      L++    E   L LP    DK IF
T  93 V-EYIVVVFECEITSGKLKK------LQYFSFSEKPPLALPYP--DK-IF 132
```

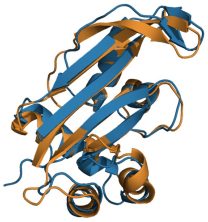


Query: 2WNY.A   Template: 2OGK.B

TM-score=0.81, RMSD=2.17, Aligned length=134, Matched length=45

```
Q   5 IHNISYCLMVYGTEDEEKVIEALRNVIPGA-TPERESAEGYHGNPITVLRGRLDRRRALREFMEKFTEV----FRGRMDE 79
      I  +    +V+ TED EKV EA+  + P         A+G++GNP+  L   L +   +++F + + E+        +
T   2 IEWVRVSAVVHSTEDREKVGEAISTLFPFEFEIAVSKAKGHYGNPMEYLEVELTKSSEIKKFWKNLLELLGEQAEEILST 81

Q  80 LEDRFDENGNLFLRLDKQKALEGVWEPVRHGDAIHLKIKVEAYPAKREVAVENIRKIL 137
      LEDR DE+  L +R+DKQKA  G       GD I +K+++  YP+KRE  +E  R++
T  82 LEDRIDEQNVLHIRIDKQKAYLGEVSLTSGGDPIAVKLRLVTYPSKREKVIEFARELC 139
```



Query: 1UAS.A   Template: 4NZJ.A

TM-score=0.95, RMSD=1.77, Aligned length=359, Matched length=154

```
Q   3 NGLGRTPQMGWNSWNHFYCGINEQIIRETADALVNTGLAKLGYQYVNIDDCWAEYSRDSQGNFVPNRQTFPSGIKALADY 82
        L  TP MGWNSWN F   + E+++ +TADA++  G+  LGY Y+NIDD W    R + G++  ++  FP GIK +ADY
T  91 DELLLTPPMGWNSWNTFGQHLTEELVLQTADAMITNGMRDLGYSYINIDDFWQLPERGADGHLQIDKTKFPRGIKYVADY 170

Q  83 VHAKGLKLGIYSDAGSQTCSNKMPGSLDHEEQDVKTFASWGVDYLKYDNCNDAGRSV--MERYTRMSNAMKTYGKNIFFS 160
       H +G+KLGIYSDA+ +TC   + GS  +EE D K FASWGVD LKYD CN     V  MERY +M  A++   ++I +S
T 171 LHERGFKLGIYSDAAEKTCG-GVCGSYGYEETDAKDFASWGVDLLKYDYCNAPVDRVEAMERYAKMGRALRATNRSIVYS 249

Q 161 LCEWGKENPATWAGR-MGNSWRTTGDIADNW---G-------SMTSRADENDQWAAYAGPGGWNDPDMLEVG-NG----G 224
       CEWG+  P  WA +  G+ WR +GDI+D W   G        + +  + N   + YAGP GWNDPDML VG +G    G
T 250 VCEWGQREPWKWAKQVGGHLWRVSGDIGDIWYRDGNRVGGLHGILNILEINAPLSEYAGPSGWNDPDMLVVGIDGKSMEG 329

Q 225 MSEAEYRSHFSIWALAKAPLLIGCDVRSMSQQTKNILSNSEVIAVNQDSLGVQGKKVQSDNGLEVWAGPLSNNRKAVVLW 304
       ++ +Y+SHFS+W +  +PLL G DVR+M+  T  IL + ++IA+NQD LG Q+++    +  ++W  PL++ RKAV  +
T 330 CTQEQYKSHFSLWCMMASPLLSGNDVRNMNDSTLKILLDPDLIAINQDVLGRQAERSIRSDHYDIWVKPLADGRKAVACF 409

Q 305 NRQSYQATITAHWSNIGLAGSVAVTARDLWAHSSFAAQG----QISASVAPHDCKMYVL 359
      NR S   T+  + + I+           L  H +    G    ++   +AP+ CK+Y++
T 410 NRASSPQTVILNENTIADLSF--EQIYCLDNHLT--KSGSDSKELIVKLAPYQCKVYIF 464
```

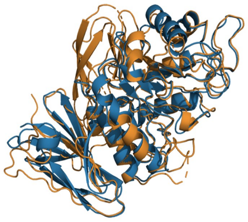


Figure S1: Five cases of protein templates across different TM-score intervals. The query-template pairs with higher structural TM-scores tend to exhibit more similar sequence alignments.