TTDESIGN: A TEST-TIME DESIGN PARADIGM FOR PROTEIN INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein language models (PLMs) have advanced the development of protein inverse folding, which aims to design amino acid sequences that fold into a target backbone structure. However, their strong generalization capabilities do not always translate into effective designs for individual target structures. In this work, we propose TTDesign, a novel test-time design paradigm for protein inverse folding that tailors sequence designs to each target structure without requiring any model parameter updates. TTDesign establishes a self-optimization loop with sequence-structure consistency as its guiding objective, iteratively refining sequence designs through structural feedback. Furthermore, TTDesign incorporates a retrieval augmentation mechanism to enhance this process with external protein knowledge. Extensive experiments on CATH v4.2 demonstrate that TTDesign outperforms all baselines in sequence-structure consistency, achieving an average scTM-score of 0.835 compared with 0.802 for the strongest baseline. TTDesign also achieves the best performance on the TS50 benchmark, demonstrating the strong generalization ability of TTDesign.

1 INTRODUCTION

Proteins play essential roles in biological processes, from catalysis and molecular recognition to cellular regulation, with their functions closely tied to molecular structure (Koehler Leman et al., 2023). Identifying sequences compatible with a target structure, however, remains challenging because of the vast sequence space and complex dependencies between sequence and structure. Protein inverse folding, the task of designing sequences that fold into target structures, is therefore fundamental to computational protein engineering, supporting the development of proteins with desired structural and functional properties (Chu et al., 2024; Khakzad et al., 2023; Zhu et al., 2026).

Protein inverse folding has advanced substantially over the past decades. Traditional template-based methods retrieve similar folds or local fragments from structural databases and transfer template-derived constraints to guide sequence design (Schmitz et al., 2012; Šali & Blundell, 1993). Subsequent deep learning methods learn geometric representations of protein backbones to guide structure-conditioned sequence generation (Ingraham et al., 2019; Tan et al., 2023; Jing et al., 2020; Dauparas et al., 2022). More recently, protein language models (PLMs) pretrained on large-scale protein corpora have provided more comprehensive representations of protein sequences and structures, thereby advancing the development of protein inverse folding (Zheng et al., 2023; Gao et al., 2024; Hayes et al., 2025; Yuan et al., 2026).

Despite these advances, there is a major limitation in most protein inverse folding methods: The strong generalization abilities of PLMs do not necessarily translate into effective designs for individual target structures. Therefore, addressing this challenge remains a critical step toward developing more effective protein inverse folding methods.

To address this challenge, we propose TTDesign, a test-time design paradigm for protein inverse folding that tailors protein sequences to individual target structures without requiring model parameter updates. TTDesign establishes a self-optimization loop with sequence-structure consistency as its objective, using structural feedback derived from discrepancies between predicted structures of designed sequences and target structures to guide iterative refinement of sequence designs. Furthermore, TTDesign incorporates a retrieval augmentation mechanism to enhance this process with external protein knowledge.

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

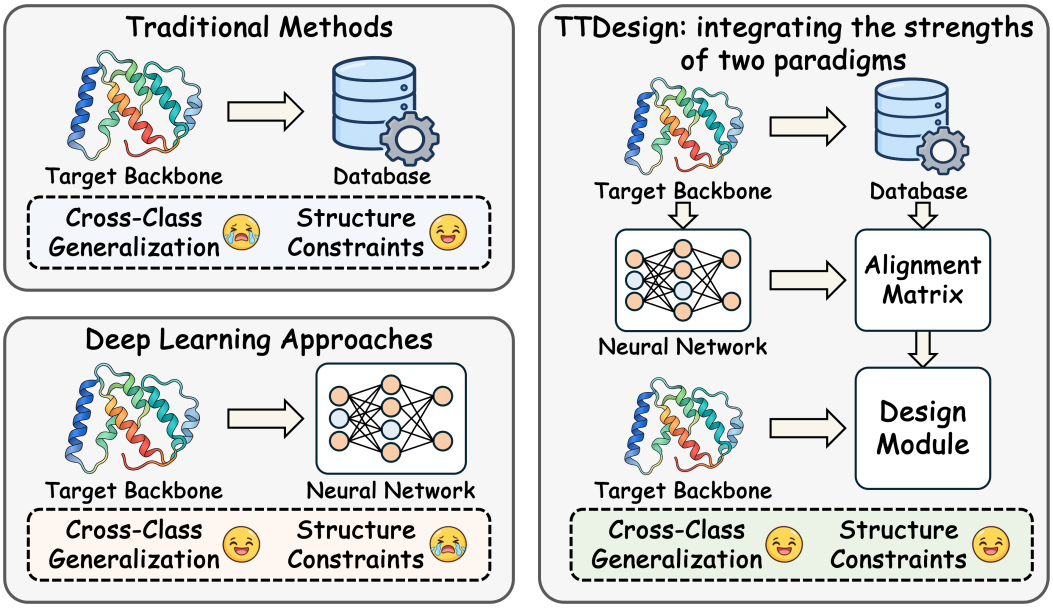


Figure 1: Comparison of different methods. Traditional methods use database-derived structural priors to provide structural constraints but often show poor cross-class generalization. Deep learning approaches generalize better to new backbones yet struggle with sequence-structure consistency. TTDesign incorporates structural priors and uses structural feedback to refine sequence designs at test time, achieving strong generalization ability and improved sequence-structure consistency.

TTDesign is a simple but effective test-time design paradigm, without requiring additional model training or fine-tuning, optimizing sequence designs for each target structure. Experiments on CATH v4.2, CATH v4.3, and TS50 demonstrate that TTDesign outperforms all baselines in scTM-score and RMSD and exhibits strong generalization ability. Most notably, while DualMPNN outperforms all other baselines using the Protein Data Bank (PDB, approximately 1,562,000 chains) database as the retrieval database, TTDesign surpasses DualMPNN on all metrics using the CATH training split as the retrieval database (approximately one-eightieth the size of the retrieval database used by DualMPNN), demonstrating the effectiveness of the proposed TTDesign paradigm.

In summary, our contributions are as follows:

- We propose TTDesign, a novel test-time design paradigm for protein inverse folding that tailors protein sequences to individual target structures without any model fine-tuning.
- TTDesign establishes a self-optimization loop that uses structural feedback to refine sequence designs toward improved sequence-structure consistency.
- TTDesign incorporates a retrieval augmentation mechanism leveraging external protein knowledge to enhance sequence designs.
- TTDesign outperforms all baselines across all evaluated metrics on CATH v4.2 and v4.3 benchmarks. Independent evaluation on TS50 further demonstrates the generalization ability of TTDesign.

2 Related Work

2.1 Protein Inverse Folding

Protein inverse folding aims to design amino acid sequences that fold into a target backbone structure. Geometric neural network based approaches encode backbone geometry and residue interactions to guide sequence generation (Ingraham et al., 2019; Tan et al., 2023; Jing et al., 2020; Dauparas et al., 2022). Protein language models (PLMs), pretrained on large-scale protein corpora,

have advanced the development of protein inverse folding. Methods like RIGA-Fold (Yuan et al., 2026), LM-Design (Zheng et al., 2023) and KW-Design (Gao et al., 2024) utilize more expressive representations provided by PLMs to optimize sequence designs. More recent approaches use preference alignment to steer sequence design toward desired protein properties (Xu et al., 2026; Wang et al., 2025; Liu et al., 2026a). TTDesign introduces a test-time design paradigm for protein inverse folding, tailoring sequence designs to individual target structures.

2.2 Test Time Adaptation

Test-time adaptation leverages information available during inference to improve performance on individual inputs or target distributions. Early studies adapt model parameters to recent observations or self-supervised signals (Krause et al., 2018; Sun et al., 2020). Subsequent work extends adaptation to prompts and nonparametric caches in vision-language models (Shu et al., 2022; Karmanov et al., 2024). For language models, retrieved examples and task demonstrations support instance-specific adaptation (Hardt & Sun, 2024; Akyürek et al., 2024), while adaptive memory architectures enable context-dependent updates (Sun et al., 2024; Behrouz et al., 2026; Feng et al., 2026). Recent approaches further employ reinforcement learning for reasoning and scientific discovery at test time (Zuo et al., 2026; Yuksekgonul et al., 2026). In protein modeling, ProteinTTT adapts PLMs to individual proteins to improve structure, fitness, and function prediction (Bushuiev et al., 2026). TTDesign establishes a test-time self-optimization loop, iteratively refining sequence designs toward improved sequence-structure consistency.

2.3 Retrieval Augmented Generation

Retrieval-augmented generation supplements pretrained models with relevant information from external sources (Lewis et al., 2020). Early studies integrated retrieved evidence into language model pretraining and generative question answering (Guu et al., 2020; Izacard & Grave, 2021). Subsequent work extended retrieval augmentation to large-scale language modeling and few-shot learning (Borgeaud et al., 2022; Izacard et al., 2023). Complementary approaches improve predictions through retrieval without updating language model parameters (Khandelwal et al., 2019; Shi et al., 2024). More recent methods adapt retrieval to generation needs and assess supporting evidence (Jiang et al., 2023; Asai et al., 2024), or combine textual and graph-based information for complex reasoning (Ma et al., 2025). In protein modeling, LLaPA retrieves information from protein-protein interaction networks to support interaction prediction, including for previously unseen proteins (Zhou et al., 2025). For protein inverse folding, DualMPNN uses retrieved structural templates to guide sequence design, demonstrating the value of external structural priors for this task (Liao et al., 2026).

3 Method

As illustrated in Fig. 2, TTDesign is a novel test-time design paradigm for protein inverse folding that tailors sequence designs to each target structure without requiring any model parameter updates. TTDesign establishes a self-optimization loop with sequence-structure consistency as its guiding objective, iteratively refining sequence designs through structural feedback. Furthermore, TTDesign incorporates a retrieval augmentation mechanism to enhance this process with external protein knowledge.

3.1 Task Description

Given a target backbone structure $\mathcal{S}$ of L residues, protein inverse folding aims to design an amino acid sequence $x = (x_1, \ldots, x_L)$ that folds into the target structure. Each $x_i \in \mathcal{V}$ denotes the amino acid at position i, where $\mathcal{V}$ is the set of 20 standard amino acids. The backbone structure $\mathcal{S}$ specifies the coordinates of N, C_α, C, and O at each residue.

3.2 Retrieval Augmentation

TTDesign is built on a pre-trained inverse folding base model f_{base} (e.g., ESM-3), which operates on the geometric representation of $\mathcal{S}$ and outputs residue-wise base probabilities $p_{base} \in \mathbb{R}^{L \times 20}$.

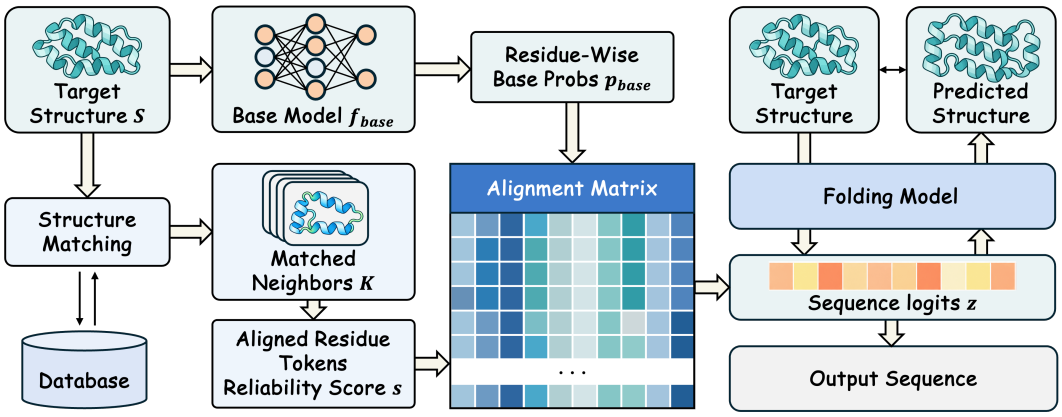


Figure 2: Overview of TTDesign. We perform structural matching to retrieve K matched neighbors for the target backbone $\mathcal{S}$. The base design and aligned neighbors form an alignment matrix A, which is used to initialize sequence logits z_0. TTDesign incorporates a protein folding model to establish a test-time self-optimization loop, iteratively evaluating discrepancies between predicted structures and the target backbone to optimize sequence logits without requiring model parameter updates.

However, the strong generalization abilities of PLMs do not necessarily translate into effective designs for individual target structures. To enrich the initial design with structural priors, TTDesign retrieves structurally similar proteins and organizes their aligned residues into an alignment matrix.

Specifically, we search a reference database $\mathcal{D}$ to find a set of matched neighbors for the target structure $\mathcal{S}$. We employ Foldseek to identify the top-K structural candidates based on geometric similarity. Let S_k denote the amino acid sequence of retrieved neighbor k, for $k = 1, \ldots, K$. For each residue i in the query structure $\mathcal{S}$, the structural alignment either identifies a corresponding residue j in S_k or indicates that position i does not align with any residue. We further quantify the quality of each neighbor using a reliability confidence score s_k (specifically, the TM-score). We organize the base design and retrieved structural priors into an alignment matrix A with $(K + 1)$ rows and L columns. Row 0 records the amino acid identities of the base design. The remaining K rows contain the aligned residues and reliability confidence scores of retrieved neighbors.

At each target position, we accumulate confidence scores by amino acid identity across the aligned neighbors and normalize the resulting contributions over amino acid categories. We combine these contributions with p_{base} using position-specific weights that reflect retrieval coverage and structural similarity, yielding the initial sequence distribution p_0. Positions without neighbor evidence retain the corresponding distribution in p_{base}. The resulting distribution integrates the base design with retrieved residue information and initializes the sequence logits for subsequent test-time optimization:

$$z_0 = \log p_0, z_0 \in \mathbb{R}^{L \times 20} \tag{1}$$

3.3 Constructing the Self-Optimization Loop

Starting from the initial sequence logits z_0, TTDesign constructs a test-time self-optimization loop that uses structural feedback to refine sequence designs for each target backbone. A frozen protein folding model connects candidate sequences to their predicted structures, allowing discrepancies from the target structure to guide subsequent sequence updates. Each target follows a single optimization trajectory initialized by retrieval augmentation.

At iteration t, the current logits $z_t \in \mathbb{R}^{L \times 20}$ define residue-wise amino acid preferences. We use Gumbel-Softmax to obtain a one-hot representation $Y_t \in \{0, 1\}^{L \times 20}$ of the candidate sequence x_t and predict its structure using ESMFold:

$$Y_t = \text{GumbelSoftmax}(z_t), \widehat{\mathcal{S}}_t = f_\theta(Y_t). \tag{2}$$

Here, f_θ denotes the folding model with fixed parameters θ, and $\widehat{\mathcal{S}}_t$ is the predicted structure of x_t. The one-hot selections are mapped to amino acid embeddings for the folding model. A straight-

through estimator preserves discrete amino acid choices during the forward pass while allowing gradients to propagate through the relaxed selections to z_t.

To guide sequence updates toward improved sequence-structure consistency, we combine two complementary structural losses. The backbone frame aligned point error (FAPE), $\mathcal{L}_{\text{FAPE}}^{(t)}$, compares predicted and target backbone frame origins in their respective residue-local coordinate systems, averaged over the structure module's prediction layers. The distogram loss, $\mathcal{L}_{\text{dist}}^{(t)}$, measures cross-entropy between predicted residue-pair distance distributions and target distance categories obtained by binning pseudo-beta distances into 64 classes. Both losses use outputs from the same folding model evaluation and structural supervision derived from the target structure $\mathcal{S}$. The resulting objective is

$$\mathcal{L}_t = 0.5\mathcal{L}_{\text{FAPE}}^{(t)} + 0.3\mathcal{L}_{\text{dist}}^{(t)}. \tag{3}$$

We backpropagate $\mathcal{L}_t$ through the frozen folding model and update z_t. The updated logits generate the next candidate, closing the loop between sequence design and structural feedback without requiring model parameter updates. After T iterations, we return the candidate with the highest TM-score between its predicted structure and the target backbone:

$$t^\star \in \arg\max_{0 \le t < T} \text{TMscore}(\widehat{\mathcal{S}}_t, \mathcal{S}), x^\star = x_{t^\star}. \tag{4}$$

4 Experiments

4.1 Experimental Setup

Datasets We conduct experiments on both **CATH v4.2** and **CATH v4.3** (Orengo et al., 1997), where proteins are categorized based on the CATH hierarchical classification of protein structures, to ensure a comprehensive analysis. Following the standard data splitting (Ingraham et al., 2019; Hsu et al., 2022), the CATH v4.2 dataset consists of 18,024 proteins for training, 608 proteins for validation and 1,120 proteins for testing; the CATH v4.3 dataset consists of 16,153 proteins for training, 1457 proteins for validation and 1,882 proteins for testing. To further assess the generalization ability of TTDesign, we additionally include the TS50 benchmark (Li et al., 2014) for evaluation.

Baselines To validate the effectiveness of TTDesign, we compare it with several state-of-the-art methods, including StructGNN (Ingraham et al., 2019), GraphTrans (Ingraham et al., 2019), GCA (Tan et al., 2023), GVP (Jing et al., 2020), ProteinMPNN (Dauparas et al., 2022), PiFold (Gao et al., 2022a), AlphaDesign (Gao et al., 2022b), KW-Design (Gao et al., 2024), ESM-3 (Hayes et al., 2025), DualMPNN (Liao et al., 2026) and ProtInvTree (Liu et al., 2026b). Details of the baseline settings are provided in Appendix A.1.

Evaluation Metrics The quality of protein sequence designs is evaluated using two key metrics: *scTM-score* and *RMSD*. *Self-Consistency TM-score (scTM-score)* measures the structural similarity between the predicted structure of a designed sequence and its target structure, with higher scores indicating greater sequence-structure consistency. *RMSD* measures the spatial deviation between corresponding C_α atoms in superimposed structures, with lower values indicating stronger geometric congruence. We also take *predicted local distance difference test (pLDDT)* into consideration. *pLDDT* measures the folding model's confidence in its structural predictions at each residue, with higher scores indicating greater confidence.

Prevention of Data Leakage and Information Leakage We use the CATH training split as the retrieval database, preventing any form of data leakage (such as overlap in CATH topology classifications between the training, validation and test sets). TTDesign incorporates ESMFold to establish the self-optimization loop. We compute scTM-score and RMSD between the target backbone and the structure predicted by ESMFold2 from the designed sequence. The evaluator is different from the protein folding model within TTDesign, eliminating information leakage.

Table 1: Structure consistency performance comparison between TTDesign and different baseline approaches on the CATH v4.2 dataset. We report results for the Short and Single-chain subsets and the full test set (All). Bold and underlined indicate the best and the second-best performance.

Model	scTM-score (↑)			RMSD (↓)		
	Short	Single-chain	All	Short	Single-chain	All
StructGNN	0.566	0.589	0.762	7.860	8.915	5.564
GraphTrans	0.575	0.609	0.759	7.942	9.038	5.587
GCA	0.563	0.578	0.756	8.090	8.646	5.716
GVP	0.573	0.605	0.766	8.084	8.211	5.444
ProteinMPNN	0.587	0.637	0.775	7.269	7.658	5.320
PiFold	0.589	0.612	0.787	7.956	8.199	5.034
AlphaDesign	0.584	0.630	0.756	7.238	7.930	5.508
KW-Design (ESM-2)	0.592	0.629	0.796	7.552	7.663	4.928
ProtInvTree	0.600	0.625	0.788	6.873	8.194	4.674
DualMPNN	<u>0.643</u>	<u>0.691</u>	<u>0.802</u>	<u>6.246</u>	<u>6.995</u>	<u>4.628</u>
ESM-3	0.594	0.631	0.799	7.949	8.400	5.083
TTDesign (ESM-3)	**0.671**	**0.715**	**0.835**	**5.600**	**6.587**	**3.815**

Table 2: Structural consistency comparison between TTDesign and baseline approaches on CATH 4.3. Bold and underlined indicate the best and the second-best performance.

Model	scTM-score (↑)	RMSD (↓)
StructGNN	0.737	5.864
GraphTrans	0.732	6.041
GCA	0.733	6.076
GVP	0.754	5.598
ProteinMPNN	0.773	5.220
AlphaDesign	0.748	5.757
PiFold	0.776	5.281
KW-Design (ESM-2)	0.792	4.982
ProtInvTree	0.785	4.908
DualMPNN	<u>0.801</u>	<u>4.647</u>
ESM-3	0.791	5.177
TTDesign (ESM-3)	**0.827**	**4.077**

Table 3: De novo protein design results on TS50. We compare structural consistency of the following methods. Bold and underlined indicate the best and the second-best performance.

Model	scTM-score (↑)	RMSD (↓)
StructGNN	0.841	3.582
GraphTrans	0.821	3.663
GCA	0.841	3.565
GVP	0.852	3.311
ProteinMPNN	0.866	3.267
AlphaDesign	0.859	3.358
PiFold	0.877	3.219
KW-Design (ESM-2)	0.872	3.219
ProtInvTree	0.838	3.682
DualMPNN	<u>0.894</u>	<u>2.731</u>
ESM-3	0.881	2.945
TTDesign (ESM-3)	**0.930**	**1.592**

4.2 Fixed Backbone Protein Design

We evaluate TTDesign on CATH v4.2 and CATH v4.3 for fixed backbone protein design, reporting scTM-score and RMSD in Tables 1 and 2. On CATH v4.2, we report results for three evaluation groups: Short, Single-chain, and All, with All denoting the full test set. The Short subset contains protein chains with at most 100 residues. The single-chain category contains only those models in which the single chain accounted for the entire protein record in the PDB. TTDesign achieves the highest scTM-score and lowest RMSD among all evaluated baselines across both datasets, demonstrating the effectiveness of TTDesign. On the full CATH v4.2 test set, TTDesign increases the average scTM-score from 0.802 to 0.835 compared with DualMPNN, the strongest baseline, while reducing RMSD from 4.628 to 3.815, a relative reduction of 17.6%. This advantage extends to the Short and Single-chain subsets, where TTDesign achieves scTM-scores of 0.671 and 0.715, compared with 0.643 and 0.691 for DualMPNN, respectively. TTDesign also attains the lowest RMSD in both subsets, showing consistent gains across the evaluated protein groups.

On CATH v4.3, TTDesign maintains its lead with an scTM-score of 0.827 and an RMSD of 4.077, compared with 0.801 and 4.647 for DualMPNN. The 12.3% relative reduction in RMSD further demonstrates closer congruence with the target backbones. TTDesign also increases scTM-score from 0.791 to 0.827 and reduces RMSD from 5.177 to 4.077 compared to ESM-3. Together with

the gains on CATH v4.2, these results demonstrate that TTDesign consistently refines sequence designs across both CATH benchmarks.

TTDesign also outperforms the test-time search baseline ProtInvTree on both metrics across both datasets. Moreover, while DualMPNN outperforms all other baselines using the Protein Data Bank (PDB, approximately 1,562,000 chains) as the retrieval database, TTDesign surpasses DualMPNN on all metrics using the CATH training split as the retrieval database (approximately one-eightieth the size of the retrieval database used by DualMPNN). Together, these results demonstrate the effectiveness of TTDesign, which refines sequence designs for individual target structures without updating model parameters, highlighting its potential in translating pretrained capabilities into more effective sequence designs for individual target structures.

4.3 De Novo Protein Design

To provide a more comprehensive evaluation and demonstrate the generalization ability of TTDesign, we also evaluate it on the TS50 benchmark. We present the quantitative results in Table 3, which reveal the following: (1) TTDesign outperforms all baseline methods in terms of both scTM-score and RMSD, achieving a scTM-score of 0.930 and a RMSD of 1.592 compared with 0.894 and 2.731 for DualMPNN, highlighting its superior ability to tailor protein sequences to individual target structures. (2) We compare TTDesign with its pretrained ESM-3 base model to evaluate the benefits of our test-time design paradigm. Using the same pretrained PLM, TTDesign increases scTM-score from 0.881 to 0.930 and reduces RMSD from 2.945 to 1.592, a relative reduction of 45.9%. These improvements demonstrate the effectiveness of TTDesign. Together, the TS50 results further support the generalization ability of TTDesign.

Table 4: Robustness of TTDesign across retrieval quality levels. CATH v4.2 test targets are grouped by the highest TM-score among their retrieved neighbors. We compare TTDesign with its ESM-3 base model within each group.

Model	**scTM-score (↑)**				**RMSD (↓)**			
	0-0.3	**0.3-0.5**	**0.5-0.7**	**0.7-1.0**	**0-0.3**	**0.3-0.5**	**0.5-0.7**	**0.7-1.0**
Counts	459	514	108	39	459	514	108	39
ESM-3	0.796	0.818	0.732	0.773	5.840	4.191	6.363	4.403
TTDesign (ESM-3)	**0.829**	**0.845**	**0.800**	**0.868**	**4.577**	**3.358**	**3.236**	**2.492**

4.4 Robustness across Retrieval Quality

To assess whether TTDesign improves sequence design across different retrieval conditions, we group CATH v4.2 test targets by the highest TM-score among their retrieved matched structures. The four intervals are $[0, 0.3)$, $[0.3, 0.5)$, $[0.5, 0.7)$, and $[0.7, 1.0)$. Of the 1,120 test targets, 973 (86.9%) fall into the two lower-similarity groups, while the remaining 147 belong to the two higher-similarity groups. Table 4 compares TTDesign with its ESM-3 base model within each group, revealing higher scTM-scores and lower RMSD values throughout.

The two lower-similarity groups account for most targets and therefore provide particularly relevant comparisons. In the $[0, 0.3)$ group, TTDesign increases scTM-score from 0.796 to 0.829 and reduces RMSD from 5.840 to 4.577, a relative reduction of 21.6%. In the $[0.3, 0.5)$ group, scTM-score increases from 0.818 to 0.845, while RMSD decreases from 4.191 to 3.358, a relative reduction of 19.9%. These gains demonstrate effective sequence refinement in groups characterized by low retrieval quality. Since grouping uses the highest retrieval TM-score for each target, even the closest retrieved neighbor has limited structural similarity in these groups. These improvements demonstrate that TTDesign effectively refines sequence designs even when no closely matching structure is found among the retrieved neighbors.

Improvements also extend to the two higher-similarity groups. In the $[0.5, 0.7)$ and $[0.7, 1.0)$ groups, TTDesign increases scTM-score from 0.732 to 0.800 and from 0.773 to 0.868, respectively. The corresponding relative reductions in RMSD are 49.1% and 43.4%. These groups exhibit the largest absolute scTM-score gains among the four retrieval-quality levels, with increases of 0.068 and 0.095,

Table 5: Foldability of sequences generated by TTDesign. Bold and underlined indicate the best and the second-best performance, respectively.

Model	CATH v4.2		CATH v4.3		TS50	
	Success (%)	pLDDT (↑)	Success (%)	pLDDT (↑)	Success (%)	pLDDT (↑)
StructGNN	84.11	74.41	83.58	74.19	92.00	76.61
GraphTrans	83.66	73.98	83.53	73.71	86.00	76.66
GCA	82.14	73.68	83.63	74.13	90.00	76.88
GVP	82.68	74.62	85.28	75.33	88.00	79.28
ProteinMPNN	84.73	75.73	86.40	77.01	90.00	78.76
PiFold	84.29	76.85	86.18	77.32	92.00	79.77
AlphaDesign	82.86	73.68	84.06	74.70	<u>94.00</u>	78.35
KW-Design (ESM-2)	85.18	<u>79.38</u>	87.83	<u>79.93</u>	90.00	81.04
ProtInvTree	86.88	76.08	88.68	76.49	88.00	78.08
DualMPNN	<u>88.48</u>	77.98	<u>90.60</u>	79.41	**96.00**	80.06
ESM-3	86.25	77.93	89.37	77.99	90.00	<u>81.34</u>
TTDesign (ESM-3)	**91.16**	**81.55**	**92.77**	**81.03**	**96.00**	**82.75**

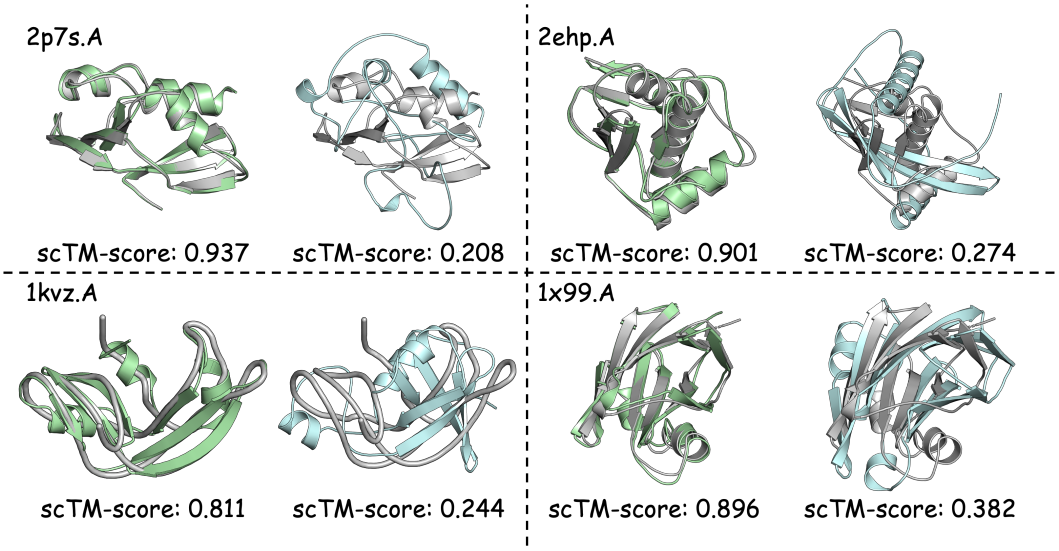


Figure 3: TTDesign generates more structurally consistent sequences. Ground-truth structures are shown in gray. Tertiary structures of sequences generated by TTDesign and ESM-3 are shown in green and blue, respectively.

respectively. These findings complement the improvements observed in the lower-similarity groups, showing that the test-time design process is also effective when more closely related structural neighbors are available. Together, these results show that TTDesign improves upon the same pre-trained model across all evaluated retrieval-quality levels. These results demonstrate the robustness of TTDesign across retrieval quality levels.

4.5 Foldability

To further validate the foldability of TTDesign, we follow GraDe-IF to evaluate each target by scTM-score, where a scTM-score above 0.5 indicates a successful design. As shown in Table 5, TTDesign achieves the highest success rates on CATH v4.2 and CATH v4.3, reaching 91.16% and 92.77%, respectively. On TS50, TTDesign achieves a success rate of 96.00%, matching the best baseline result. The higher success rates on CATH complement the improvements in average structural consistency, showing that more targets satisfy the successful-design criterion. TTDesign also attains the highest pLDDT on all three benchmarks, with scores of 81.55, 81.03, and 82.75, respectively. These scores exceed the strongest corresponding baselines, indicating greater confidence in

the predicted structures of the designed sequences. Together, the success rates and pLDDT results further support the effectiveness of TTDesign.

We also visualize some of the designed sequences in Fig. 3. The ground-truth structures are shown in gray. The tertiary structures of sequences designed by TTDesign and ESM-3 are shown in green and blue, respectively. In the illustrated cases, structures predicted from TTDesign sequences align more closely with the target backbones than those predicted from ESM-3 sequences. These results further demonstrate the effectiveness of TTDesign.

Table 6: Ablation study on CATH v4.2. Bold and underlined indicate the best and the second-best performance, respectively.

Configuration	**scTM-score** (↑)			**RMSD** (↓)		
	Short	**Single-chain**	**All**	**Short**	**Single-chain**	**All**
ESM-3	0.594	0.631	0.799	7.949	8.400	5.083
w/o optimization loop	0.609	0.646	0.784	7.519	7.838	5.162
w/o retrieval augmentation	<u>0.636</u>	<u>0.661</u>	<u>0.814</u>	<u>6.972</u>	<u>7.026</u>	<u>4.320</u>
TTDesign (ESM-3)	**0.671**	**0.715**	**0.835**	**5.600**	**6.587**	**3.815**

4.6 Ablation

We conduct ablation studies to examine the contributions of the self-optimization loop and retrieval augmentation. Table 6 compares the full TTDesign framework with two variants and the ESM-3 base model. The variant without the optimization loop retains retrieval-augmented initialization but omits iterative structural feedback. The variant without retrieval augmentation optimizes sequences initialized directly from the base model. TTDesign achieves the highest scTM-score and lowest RMSD across all three evaluation groups.

The self-optimization loop plays a central role in translating initial designs into improved sequences. Compared with the variant without the loop, TTDesign increases scTM-score from 0.784 to 0.835 and reduces RMSD from 5.162 to 3.815 on the full test set, corresponding to a relative reduction of 26.1%. The improvements also extend to both the Short and Single-chain subsets. Even without retrieval augmentation, the optimization loop improves upon ESM-3 across all reported metrics, increasing the overall scTM-score from 0.799 to 0.814 and reducing RMSD from 5.083 to 4.320. These results demonstrate the value of structural feedback for refining sequences at test time without updating model parameters.

Retrieval augmentation further strengthens the optimization process by enriching sequence initialization. Compared with optimization without retrieval, the full framework increases the overall scTM-score from 0.814 to 0.835 and reduces RMSD from 4.320 to 3.815, a relative reduction of 11.7%. On the Short and Single-chain subsets, retrieval augmentation increases scTM-score from 0.636 to 0.671 and from 0.661 to 0.715, respectively, with corresponding RMSD reductions from 6.972 to 5.600 and from 7.026 to 6.587. These consistent gains support retrieval augmentation as an effective enhancement to the optimization starting point.

5 Conclusion

In this work, we introduce TTDesign, a test-time design paradigm for protein inverse folding that tailors sequences to individual target structures without requiring model parameter updates. TTDesign establishes a self-optimization loop that uses structural feedback to iteratively refine sequences toward improved sequence-structure consistency. Retrieval augmentation complements this loop by incorporating external structural priors into sequence initialization. Experiments on CATH v4.2 and CATH v4.3 demonstrate superior performance in scTM-score and RMSD, while evaluation on TS50 further demonstrates strong generalization ability. Additional analyses show strong robustness across retrieval-quality levels and confirm the contributions of both the optimization loop and retrieval augmentation. Together, these findings establish TTDesign as an effective approach to translating pretrained model capabilities into improved designs for individual protein structures.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing and to draft selected sections of the paper. All AI-assisted text was carefully reviewed, revised and verified by the authors for accuracy, clarity and consistency with the underlying methods and experimental results. We did not use generative AI tools for the other purposes listed in the submission form. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including CATH v4.2, CATH v4.3 and TS50, were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

The information needed to reproduce the main experimental results is provided in Section 4.1, Appendix A.1 and Appendix A.2.

REFERENCES

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Anton Bushuiev, Roman Bushuiev, Olga Pimenova, Nikola Zadorozhny, Raman Samusevich, Elisabet Manaskova, Rachel Seongeun Kim, Hannes Stärk, Jiri Sedlar, Martin Steinegger, et al. One protein is all you need. In *International Conference on Learning Representations*, volume 2026, pp. 59865–59909, 2026.

Alexander E Chu, Tianyu Lu, and Po-Ssu Huang. Sparks of function by de novo protein design. *Nature biotechnology*, 42(2):203–215, 2024.

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.

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Wenhao Huang, Di He, and Tianle Cai. In-place test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 114452–114472, 2026.

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

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

Zhangyang Gao, Cheng Tan, Xingran Chen, Yijie Zhang, Jun Xia, Siyuan Li, and Stan Z Li. Kw-design: pushing the limit of protein design via knowledge refinement. In *International Conference on Learning Representations*, volume 2024, pp. 34556–34574, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54625–54640, 2024.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

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*, pp. 8946–8970. PMLR, 2022.

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

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pp. 874–880, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251): 1–43, 2023.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

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.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14162–14171. IEEE, 2024.

Hamed Khakzad, Ilia Igashov, Arne Schneuing, Casper Goverde, Michael Bronstein, and Bruno Correia. A new age in protein design empowered by deep learning. *Cell Systems*, 14(11):925–939, 2023.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

Julia Koehler Leman, Pawel Szczerbiak, P Douglas Renfrew, Vladimir Gligorijevic, Daniel Berenberg, Tommi Vatanen, Bryn C Taylor, Chris Chandler, Stefan Janssen, Andras Pataki, et al. Sequence-structure-function relationships in the microbial protein universe. *Nature communications*, 14(1):2351, 2023.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. Dynamic evaluation of neural sequence models. In *International Conference on Machine Learning*, pp. 2766–2775. PMLR, 2018.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Zhixiu Li, Yuedong Yang, Eshel Faraggi, Jian Zhan, and Yaoqi Zhou. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins: Structure, Function, and Bioinformatics*, 82(10): 2565–2573, 2014.

Xuhui Liao, Zhiqiang Liang, Liwei Xiao, Junjie Chen, et al. Dualmpnn: Harnessing structural alignments for high-recovery inverse protein folding. *Advances in Neural Information Processing Systems*, 38:111623–111643, 2026.

Junqi Liu, Xiaoyang Hou, Chence Shi, Xin Liu, Zhi Yang, and Jian Tang. Property-driven protein inverse folding with multi-objective preference alignment. In *International Conference on Learning Representations*, volume 2026, pp. 151064–151087, 2026a.

Mengdi Liu, Xiaoxue Cheng, Zhangyang Gao, Hong Chang, Cheng Tan, Shiguang Shan, and Xilin Chen. Protinvtree: Deliberate protein inverse folding with reward-guided tree search. *Advances in Neural Information Processing Systems*, 38:28337–28360, 2026b.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pp. 52782–52806, 2025.

Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Andrej Šali and Tom L Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*, 234(3):779–815, 1993.

Christophe Schmitz, Robert Vernon, Gottfried Otting, David Baker, and Thomas Huber. Protein structure determination from pseudocontact shifts using rosetta. *Journal of molecular biology*, 416(5):668–677, 2012.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 8371–8384, 2024.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Cheng Tan, Zhangyang Gao, Jun Xia, Bozhen Hu, and Stan Z Li. Global-context aware generative protein design. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. Proteinzero: Self-improving protein generation via online reinforcement learning. *arXiv preprint arXiv:2506.07459*, 2025.

Junde Xu, Zijun Gao, Xinyi Zhou, Xingyi Cheng, Le Song, Guangyong Chen, Pheng-Ann Heng, and Jiezhong Qiu. Protein inverse folding from structure feedback. *Advances in Neural Information Processing Systems*, 38:24274–24299, 2026.

Sisi Yuan, Jiehuang Chen, Junchuang Cai, Dong Xu, Xueliang Li, Zexuan Zhu, and Junkai Ji. Rigafold: A general framework for protein inverse folding via recurrent interaction and geometric awareness. *arXiv preprint arXiv:2602.04637*, 2026.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv preprint arXiv:2601.16175*, 2026.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. In *International conference on machine learning*, pp. 42317–42338. PMLR, 2023.

Peng Zhou, Pengsen Ma, Jianmin Wang, Xibao Cai, Haitao Huang, Wei Liu, Longyue Wang, Lai Hou Tim, and Xiangxiang Zeng. Large language and protein assistant for protein-protein interactions prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11312–11327, 2025.

Yiheng Zhu, Zitai Kong, Jialu Wu, Mingze Yin, Weize Liu, Yuqiang Han, Hongxia Xu, Chang-Yu Hsieh, Tingjun Hou, and Jian Wu. Generative ai for controllable protein sequence design: A survey. *npj Drug Discovery*, 3(1):19, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Experimental Setup

A.1 Baseline Settings

We evaluate eleven protein inverse folding baselines: StructGNN (Ingraham et al., 2019), GraphTrans (Ingraham et al., 2019), GCA (Tan et al., 2023), GVP (Jing et al., 2020), ProteinMPNN (Dauparas et al., 2022), PiFold (Gao et al., 2022a), AlphaDesign (Gao et al., 2022b), KW-Design (Gao et al., 2024), ESM-3 (Hayes et al., 2025), DualMPNN (Liao et al., 2026), and ProtInvTree (Liu et al., 2026b). Pretrained checkpoints for StructGNN, GraphTrans, GCA, GVP, ProteinMPNN, PiFold, AlphaDesign, and KW-Design are available at `https://zenodo.org/record/8031783`. Pretrained checkpoints for ESM-3 are available through `https://github.com/Biohub/esm`. We retrain DualMPNN using the publicly available implementation at `https://github.com/chen-bioinfo/DualMPNN`. For ProtInvTree, we independently implement and evaluate the method, as its source code is not publicly available.

A.2 Implementation Details

We use ESM-3 as the inverse folding base model, ESMFold for structural feedback during optimization, and ESMFold2 for final structural evaluation. For retrieval, we use Foldseek to search the CATH training structure database and retain the top-K neighbors per target with K=10. We optimize the sequence logits for $T = 200$ iterations using Adam with learning rate 0.02. We evaluate one candidate per iteration and return the candidate with the highest TM-score along the trajectory, yielding one final sequence per target. The backbone FAPE and distogram loss weights are set to 0.5 and 0.3, respectively. For backbone FAPE, both the distance cutoff and the normalization scale are 10 Å, with a numerical stabilizer of 10^{-4}. The distogram uses 64 distance classes with 63 boundaries spanning 2.3125 Å to 21.6875 Å at intervals of 0.3125 Å.