ProteinReasoner: A Multi-Modal Protein Language Model with Chain-of-Thought Reasoning for Efficient Protein Design

Chaozhong Liu[1*], Linlin Chao[1*], Shaomin Ji[1], Hao Wang[1], Taorui Jiang[2,†], Zhangyang Gao[1], Yucheng Guo[1], Ming Yang[1], Xiaoming Zhang[1,✉]

[1]BioMap Research, Beijing, China

[2]Stanford University, California, USA

[*]Equal Contribution, [†]Work done during internship at BioMap

✉zhangxiaoming@biomap.com

Abstract

Protein language models (PLMs) have advanced the understanding and engineering of proteins by learning rich representations from large-scale sequence data. However, sequence-only models are limited in their ability to capture structural and evolutionary constraints essential for protein tasks. Although recent multi-modal PLMs integrate sequence and structure, they often fail to explicitly model the stepwise reasoning processes fundamental to protein science, particularly the evolutionary constraints and decision-making logic critical for protein design and optimization. Here, we introduce ProteinReasoner, a multi-modal protein language model that explicitly incorporates the "evolutionary profile" as the intermediate reasoning step between structure and sequence modalities within a chain-of-thought (CoT) framework. We demonstrated that ProteinReasoner achieved improved zero-shot performance in structure prediction, inverse protein folding, and fitness prediction tasks, consistently outperforming larger baselines including ESM3 and DPLM-2. Furthermore, we developed a novel In-context learning (ICL) paradigm for protein optimization that leverages ProteinReasoner's reasoning capabilities to guide sequence generation based on prior experimental feedback. ProteinReasoner outperformed the conventional active learning paradigm in protein optimization tasks, achieving higher predictive accuracy and better generalization. ProteinReasoner offers a scalable, efficient, and generalizable framework for protein modeling and optimization, providing a practical path to accelerate protein engineering workflows and enhance mechanistic understanding of protein biology.

1. Introduction

Protein language models (PLMs) have emerged as powerful tools for understanding and engineering proteins. Models such as the ESM family[1,2] have demonstrated that large-scale self-supervised training on protein sequences can yield representations useful for diverse biological tasks, including structure prediction, mutational effect estimation, and function annotation[1–3]. These sequence-based models have catalyzed a paradigm shift in computational biology, enabling scalable, data-driven protein analysis without the need for extensive experimental inputs. By learning contextual representations of amino acids, PLMs have provided a new lens to explore protein biology, akin to the impact of language models

in natural language processing.

However, the limitations of sequence-only models have become increasingly apparent. Benchmarks such as ProteinGym[4] reveal that models incorporating structural or evolutionary information (Multiple Sequence Alignment, MSA) outperform those based solely on sequence. For example, predicting the impact of point mutations on protein stability or function often requires understanding of 3D structure and evolutionary constraints—information that is not fully captured by sequence alone. Accordingly, there is growing interest in developing multi-modal approaches to protein language modeling that incorporate complementary data sources to more comprehensively represent the complexity of protein biology.

Building on this momentum, recent research has increasingly focused on integrating structural and evolutionary information with sequence-based representations. Notable examples, such as ESM3[5], ProSST[6], SaProt[7], and DPLM-2[8], illustrate this growing trend, leveraging diverse input modalities to improve performance across a range of tasks. However, despite these advances, a central challenge remains: how to pretrain multi-modal protein language models (PLMs) in a way that effectively fuses structural, sequence, and evolutionary signals to fully realize their complementary potential and optimize model capacity.

To answer this question, it is essential to consider how protein scientists interpret and reason about protein function. A compelling analogy can be drawn between protein science workflows and "reasoning" processes. For instance, AlphaFold2[9] implicitly reasons residue-residue relationships using MSA-derived co-evolutionary signals to predict 3D structures (**Figure 1A**). The MSA serves as an information-rich context that encodes which residues co-vary across homologs, implying structural proximity or functional relevance[10,11]. Similarly, protein optimization[12] tasks leverage directed evolution as an intrinsic reasoning mechanism. Here, experimental fitness measurements serve as reasoning signals that guide subsequent mutagenesis decisions. In each round, a set of sequence variants is generated and evaluated for properties such as stability or binding affinity. The results from these fitness assays inform which mutations to carry forward or explore next, effectively constructing an empirical reasoning trajectory through the fitness landscape. This directed evolution process encodes functional insights over time and reflects a latent, iterative decision-making logic grounded in experimental evidence.

These examples suggest that evolutionary reasoning steps are intrinsic to protein workflows, yet they remain underrepresented in most multi-modal PLMs. Current models typically adopt end-to-end mappings from input to output, which introduces two key limitations: they neglect the intermediate reasoning logic that underpins evolutionary constraints, and they often omit MSAs as an input modality. MSAs are not only biologically meaningful but also encode rich, structured representations of residue conservation and covariation. While alignment-free models are appealing for their speed and scalability, they consistently underperform compared to MSA-driven pipelines like AlphaFold2 in terms of accuracy[13–15]. These limitations highlight the need for a new modeling paradigm, which treats evolutionary signals not merely as inputs, but as an explicit reasoning step within the modeling process. By aligning with the logic found in biological workflows, such models can better integrate structural, sequence, and evolutionary modalities.

Building on this intuition, we introduce ProteinReasoner, a generative foundation model that incorporates structure and sequence as primary modalities, with the "evolutionary profile" (profile)

inspired by ProfileBFN[16] serving as an intermediate reasoning modality (**Figure 1A**, middle). This profile, derived from position-wise amino acid frequency to represent natural or directed evolution signals, provides latent constraints that guide the model's multi-modal understanding. Rather than treating evolutionary information as an auxiliary feature, ProteinReasoner integrates it as a central component of its reasoning process, analogous to chain-of-thought prompting in large language models[17].

During pretraining, ProteinReasoner captures the logic of protein science tasks by modeling directional flows between modalities, including both *sequence* → *profile* → *structure* and its reverse. This design leads to improved zero-shot performance across key tasks, including structure prediction, inverse folding, and fitness prediction. Beyond pretraining, ProteinReasoner supports protein optimization through in-context learning (ICL)[18], using few-shot prompts grounded in prior experimental assays. Unlike active learning methods used in protein optimization[19–21], which typically require retraining and model updates for each task, the reasoning-driven ICL approach efficiently leverages prior examples by explicitly reasoning in context, making better use of foundation model capabilities without additional training.

In summary, ProteinReasoner represents a conceptual and technical advance in the design of multi-modal PLMs. By inserting reasoning steps within the modeling pipeline, it reflects a logic-driven approach to protein modeling. This framework leverages explicit reasoning steps to improve multi-modal integration and support zero-shot generalization across diverse protein design tasks.

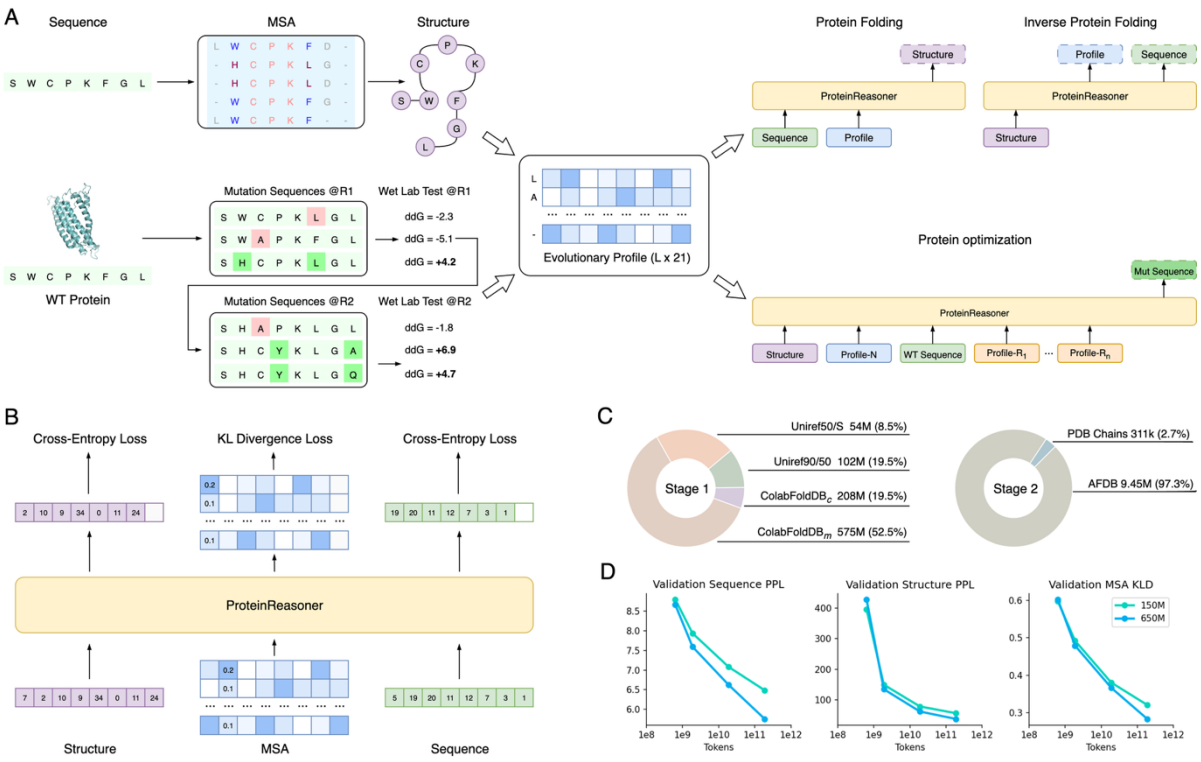


Figure 1. ProteinReasoner is a multi-modal generative model with the evolutionary profile as the intermediate reasoning step. (A) Conceptual framework showing how intrinsic evolutionary reasoning in protein science inspired ProteinReasoner's design. Left: Protein science workflows inherently involve stepwise reasoning, where natural or directed evolution guides protein tasks. *WT* stands for wild-type; R_1 means round 1 of the optimizations; *ddG* stands for the protein stability measurement in terms of Gibbs Free Energy. Middle: Evolution is identified as the common intermediate reasoning

step, and the $L \times 21$ evolutionary profile is designed to explicitly model this reasoning step. L is the sequence length. Values in the profile represent the amino acid frequency. Right: ProteinReasoner is designed to explicitly reason through this intermediate step, connecting structure and sequence, and can be further extended to protein optimization tasks by appending rounds of directed evolutionary profile to the reasoning chain. *Profile-N* stands for the natural evolutionary profile; *Profile-Rn* stands for the directed evolutionary profile constructed from round *n* wet lab experimental scores. **(B)** Model architecture and multi-modal training process. ProteinReasoner processes sequence, evolutionary profile, and structure as separate token streams throughout the transformer layers. Decoder heads predict the next token in each modality. Training was performed in both folding (not shown) and inverse folding directions (shown). **(C)** Overview of pretraining datasets. Stage 1 used sequence-only datasets; Stage 2 incorporated structure datasets with MSAs for all proteins. Percentages in the plot indicate token counts. **(D)** Models were trained in two sizes, 150M and 650M parameters, using up to 189B tokens.

2. Results

2.1 ProteinReasoner: A Multi-Modal Model with Chain-of-Thought Reasoning

ProteinReasoner is a multi-modal generative protein language model that integrates sequence, structure, and evolutionary profile information as distinct and independently processed modalities (**Figure 1B**).

The input data for ProteinReasoner consists of three distinct modalities. The amino acid sequence is tokenized at the residue level into a sequence of length L. The 3D protein structure is discretized using the DPLM-2 structure tokenizer[8], which encodes atomic coordinates into structural tokens aligned to the same length. Central to ProteinReasoner's design is the evolutionary profile, which functions as a latent reasoning step. This profile is represented as an L×21 numerical matrix derived from multiple sequence alignments (MSAs), where each row corresponds to a residue position and each column denotes the frequency of the 20 standard amino acids plus one gap character. This profile captures conservation and covariation patterns across homologous sequences and serves as a biologically grounded prior reflecting natural evolutionary constraint (see **4.1** for details).

During pretraining, the model receives concatenated input in either the (*structure* → *profile* → *sequence*) or (*sequence* → *profile* → *structure*) order (**Supplementary Figure 1**). It is trained to simultaneously predict the next structure token, the next amino acid, and the evolutionary profile of the subsequent position. This training objective mirrors folding and inverse folding processes to structure the reasoning flow between modalities. In the forward direction, the model learns to predict structure from sequence and profile; In the reverse direction, it reconstructs sequence from structural and profile information. The evolutionary profile thus functions as a latent, intermediate reasoning step that links input and output modalities, forming a conceptual chain-of-thought that informs downstream predictions. In addition, this bidirectional formulation reinforces cross-modal associations and enhances the model's performance on both generative and predictive tasks. In contrast to models like ESM3, which fuse all modalities at the input embedding layer, ProteinReasoner preserves modality separation across the transformer layers. This design allows the attention mechanism to learn inter-modality dependencies through structured reasoning, enhancing modeling capacity, and aligns with recent trends in multi-modal architectures such as the Chain of Modality framework[22].

ProteinReasoner is pretrained using a two-stage strategy (**Figure 1C**). The first stage is a previously published sequence-only model[23] trained on 1 trillion amino acid tokens collected from diverse sources. The multi-modal pretraining stage (stage 2) initializes from this model and is trained on 9.45 million

AlphaFold Database[24] (AFDB) structures and 311k chains from the PDB-REDO[25] with refined X-ray structures from the Protein Data Bank[26] (PDB). We trained two model sizes, 150 million and 650 million parameters, with a maximum training budget of 1.89×10^{11} tokens. Evaluation on a held-out validation set (**Figure 1D**) shows that the 650M model achieves lower perplexity on both sequence and structure tokens, as well as lower KL divergence on the predicted evolutionary profile. These improvements indicate superior modeling capacity.

We next assessed how ProteinReasoner's chain-of-thought formulation supports zero-shot generalization across downstream protein modeling tasks, including structure prediction, inverse folding, and fitness prediction.

2.2 Zero-Shot Protein Tasks Performance

To evaluate the generalizability and modeling capacity of ProteinReasoner, we assessed its zero-shot performance on three representative protein modeling tasks: structure prediction[27], inverse protein folding[28], and fitness prediction[29]. We compared results against ESM3-Open 1.4B and DPLM-2, two strong multi-modal baselines, to contextualize ProteinReasoner's performance across these tasks.

The structure prediction task evaluated two modes of reasoning (**Figure 2A, B**): (1) an "externally guided" mode, where both the amino acid sequence and the evolutionary profile derived from MSAs were provided to the model, and (2) an "internally inferred" mode, where only the sequence was provided, requiring the model to generate the evolutionary profile before predicting structural tokens. These tokens were then converted into atom-level coordinates using the DPLM-2 structure decoder. Final performance is evaluated using the Root Mean Square Deviation (RMSD), and Template Modeling score (TM-score)[30], which quantify the structural similarity between predicted and ground-truth coordinates. We compared ProteinReasoner, ESM3, and DPLM-2 across four evaluation datasets: CAMEO, CASP14, CASP15, and a PDB date-based holdout split used in previous studies[31].

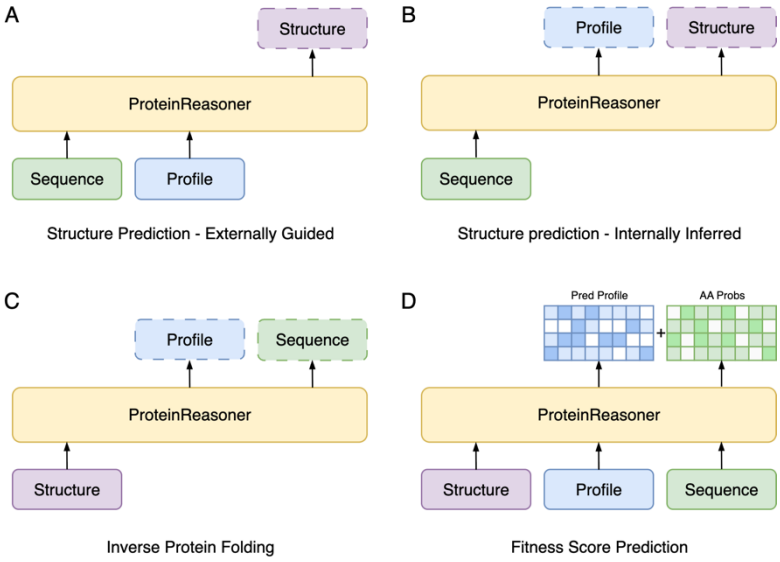


Figure 2. Overview of ProteinReasoner's downstream task configurations. (A) *Externally guided* structure prediction: The model predicts protein structure tokens using an externally provided evolutionary profile as a reasoning guide. **(B)** *Internally inferred* structure prediction: The model first internally generates the evolutionary profile and then uses it to predict structure tokens. **(C)** Inverse protein folding: Given a protein structure, the model first generates an evolutionary profile, which is then used to generate the amino acid sequence. **(D)** Fitness score prediction: In the ProteinGym benchmark, the model takes

structure, profile, and wild-type sequence as inputs. Final fitness scores are computed by summing the log-softmax outputs from the predicted profile and sequence head logits.

ProteinReasoner-650M in externally guided mode consistently outperformed all baseline models across four benchmark datasets, despite having a smaller parameter count than models like ESM3-Open-1.4B and DPLM-2-3B (**Table 1**). In terms of TM-score, it achieved relative improvements of up to 26.3% over DPLM-2-650M on CASP14, and 14.2% on CAMEO, while maintaining gains of 10.3% over ESM3 on CASP14 and 7.1% on the PDB Date Split. Corresponding RMSD improvements are also substantial, with absolute reductions of up to 0.81 Å (CASP14 vs. DPLM-2-650M) and 0.36 Å (PDB Split vs. ESM3). These gains are observed across all four datasets, highlighting the model's ability to generalize across both curated and challenging benchmarks. Interestingly, the internally inferred mode of ProteinReasoner performs comparably to other baselines even without access to external MSAs, demonstrating the effectiveness of its learned internal reasoning capabilities.

Model scaling also contributed positively: the 650M model achieved higher (5.7% ~ 18.2%) TM-scores and lower (-0.06 Å ~ -0.68 Å) RMSD than the 150M counterpart under both externally guided and internally inferred settings. Finally, across both model sizes, the externally guided mode outperformed the internally inferred mode by 9.5% ~ 33.8% in TM-score and -0.18 Å ~ -0.82 Å in RMSD. This gap suggests that while the model can internally generate a useful evolutionary profile, the quality of these inferred profiles still lags behind externally provided ones. Improving profile generation remains an important direction for future work.

Model	Method	CAMEO (194)	CASP14 (50)	CASP15 (53)	PDB Date Split (449)
ESM3-Open-1.4B	Argmax	2.98/0.721	4.20/0.531	3.58/0.578	2.56/0.769
	Iterative	2.70/0.747	3.90/0.566	3.57/0.584	2.37/0.794
DPLM-2-650M	100 Iters	3.23/0.678	4.44/0.494	3.84/0.517	2.61/0.771
DPLM-2-3B	100 Iters	3.02/0.711	4.30/0.527	3.84/0.543	2.34/0.811
ProteinReasoner-150M	Guided	2.74/0.732	3.95/0.557	3.59/0.555	2.39/0.793
	Inferred	3.56/0.598	4.39/0.427	4.29/0.464	3.15/0.655
ProteinReasoner-650M	Guided	**2.50/0.774**	**3.63/0.624**	**3.53/0.593**	**2.01/0.850**
	Inferred	2.93/0.707	4.19/0.466	3.71/0.527	2.47/0.760

Table 1. Zero-shot structure prediction benchmark. Values represent RMSD/TM-Score. *Guided* refers to the externally guided mode where the evolutionary profile is provided, and *Inferred* refers to the internally inferred mode where the model generates the evolutionary profile. *Iters* means iterations.

The inverse protein folding task evaluated the model's ability to reconstruct an amino acid sequence from a given protein structure. Unlike structure prediction, we evaluated only the internally inferred mode for inverse protein folding (structure only, with the model generating its own evolutionary profile, **Figure 2C**), since providing profiles from MSAs would leak task-relevant information. Evaluation metrics included average amino acid recovery (AAR), which measures how accurately the model recovers the original sequence at each position, and self-consistency TM-score (scTM), which compares the original input structure with the structure predicted by AlphaFold2[9] using the model-generated sequence. This metric assesses how well the predicted sequence can recapitulate the original structure,

capturing the core objective of the inverse protein folding task—designing novel sequences that reliably fold into a specified structure. While AAR measures residue-level accuracy, scTM more directly evaluates the functional plausibility of the predicted sequence in a structural context.

ProteinReasoner-650M (internally inferred mode) achieved the highest structural consistency among all models, with scTM scores of 0.786 and 0.918 on CAMEO+CASP and the PDB date split respectively, alongside strong AAR scores of 47.77 and 59.22 (**Table 2**). Compared to ESM3-Open-1.4B, ProteinReasoner-650M achieved significant improvements in AAR and scTM on both datasets. Against DPLM-2-3B, despite having 5.1% lower AAR, it achieved a 1.9% higher scTM, suggesting that while DPLM-2-3B may better recover native residues, ProteinReasoner produces more structurally compatible yet sequence-diverse solutions.

We also observe benefits of model scaling: the 650M variant improved over ProteinReasoner-150M by 6.7% in AAR and 2.2% in scTM on CAMEO+CASP, and by 10.0% in AAR and 2.0% in scTM on the PDB Split. Notably, ProteinReasoner-150M already outperforms all baselines except DPLM-2-3B on both datasets in terms of scTM score, demonstrating the strong performance of the architecture even at a small scale.

Model	**Method**	**CAMEO+CASP**		**PDB Date Split**	
		AAR (297)	**scTM (50)**	**AAR (393)**	**scTM (50)**
ESM3-Open-1.4B	Argmax	38.17	0.689	40.76	0.838
	iterative	38.33	0.715	41.05	0.862
DPLM-2-650M	100 Iters	45.35	0.734	56.33	0.885
DPLM-2-3B	100 Iters	<u>46.85</u>	0.746	**62.39**	<u>0.901</u>
ProteinReasoner-150M	Inferred	44.78	<u>0.769</u>	53.84	0.900
ProteinReasoner-650M	Inferred	**47.77**	**0.786**	<u>59.22</u>	**0.918**

Table 2. Zero-shot inverse protein folding benchmark. *Inferred* refers to the internally inferred mode where the model generates the evolutionary profile. *Iters* means iterations.

In addition to generative tasks, we evaluated ProteinReasoner on protein understanding tasks, specifically focusing on predicting the effects of mutations. To this end, we benchmarked the model's performance using the ProteinGym DMS substitution dataset[4], which includes comprehensive mutational scanning data across diverse proteins. The dataset encompasses various types of mutations, capturing a broad landscape of possible protein variations. The primary objective is to predict the functional consequences of these mutations. Performance was assessed using Spearman correlation to quantify the rank-order agreement between predicted and experimentally observed fitness scores, and the area under the receiver operating characteristic curve (AUROC) to evaluate the model's ability to distinguish beneficial mutations from deleterious ones.

For fitness prediction on ProteinGym, we arranged all inputs following the inverse folding direction, providing the structure, evolutionary profile, and wild-type (WT) sequence (**Figure 2D**). We summed the log-softmax-normalized predicted profile logits and sequence logits from the model's output headers and calculated the odds ratio between the mutated amino acid and the WT amino acid at each mutated position to estimate fitness impact. For multiple mutations, we approximated the combined effect by summing the odds ratios of the individual mutations, assuming additive contributions (see **4.4** for details).

The 650M model outperformed ESM3-Open-1.4B according to benchmark results reported on the ProteinGym website, demonstrating the effectiveness of ProteinReasoner in fitness prediction despite its smaller size (**Table 3**). Interestingly, scaling from 150M to 650M yielded no improvements, suggesting that this mutation effect prediction task may be less sensitive to model scale compared to generative tasks such as structure prediction.

Model	Avg. Spearman	Avg. AUC
ESM3-Open-1.4B	0.466	0.755
ProteinReasoner-150M*	**0.489**	**0.767**
ProteinReasoner-650M*	0.482	0.765

Table 3. Zero-shot ProteinGym benchmark. *Evaluated with the checkpoint at the 3k step.

Together, these evaluations demonstrate that ProteinReasoner's chain-of-thought framework enables strong zero-shot performance across diverse protein modeling tasks. The model achieved superior structure prediction and inverse protein folding accuracy compared to ESM3-Open 1.4B and DPLM-2, despite its smaller size, highlighting the efficiency of its training design. Gains from scaling were evident in generative tasks, while fitness prediction appeared less dependent on model size. The performance gap between externally guided and internally inferred settings emphasizes the need to further improve the model's ability to generate high-quality evolutionary profiles. Overall, these results support reasoning-driven multi-modal integration as a promising approach for protein understanding and design.

2.3 Ablation Study: Validating the Chain-of-Thought Design and Training Strategies

To thoroughly assess the design choices of ProteinReasoner, we performed ablation studies focusing on the role of evolutionary profile reasoning and the impact of pretraining strategies. These studies elucidate the critical components that contribute to the model's capability across diverse protein tasks. To ensure a fair comparison, we trained all models using the same set of hyperparameter settings not involved in the ablation comparisons, except for learning rate adjustments in a few models to prevent early overfitting (**Table S5**). All 150M models were evaluated at the same training step (30k) using the identical training dataset, ensuring that observed performance differences can be attributed to the variables under investigation rather than confounding factors.

The evolutionary profile significantly enhances model performance in structure prediction and fitness prediction tasks. When comparing models trained with (Profile = Yes) and without (Profile = No) the profile as the intermediate reasoning, we observed consistent performance gains when the profile was included.

Direction	Profile	Initialization	CAMEO	CASP14	CASP15	PDB Date Split
Bi-direction	Yes	Pretrained	0.722/0.577	0.524/0.409	0.551/0.451	0.782/0.634
Bi-direction	No	Pretrained	0.543	0.399	0.445	0.576
Forward	Yes	Pretrained	0.760/0.607	0.579/0.432	0.577/0.469	0.821/0.663
Bi-direction	Yes	Scratch	0.650/0.388	0.471/0.325	0.507/0.315	0.715/0.380

Table 4. Ablation study on structure prediction. Sizes of all models listed here are 150M. Values represent Guided/Inferred TM-Scores for models trained with evolutionary profile (Profile = Yes). Forward means the model is pretrained with only

(sequence → profile → structure) order.

Specifically, in structure prediction (**Table 4**), bi-directional models pretrained with the profile consistently improved TM-scores by 23.8%-35.8% in the externally guided setting and by 1.0%-6.3% in the internally inferred setting compared to two-modality models that excluded the reasoning step. These results demonstrate the critical value of incorporating the evolutionary profile, regardless of whether it is externally provided or internally generated.

In the ProteinGym fitness prediction task (**Table 5**), models using the full three-modality setup with evolutionary profiles (Profile = Yes) consistently outperformed their two-modality counterparts (Profile = No). For example, the bi-directional model with the profile achieved a Spearman correlation of 0.469, which is a 10.9% improvement compared to 0.423 when the profile was omitted. These results highlight the strong contribution of the evolutionary profile in tasks that require reasoning about evolution and mutation effects.

Direction	Profile	Initialization	ProteinGym (217)	CAMEO+CASP		PDB Date Split	
				AAR (297)	scTM (50)	AAR (393)	scTM (50)
Bi-direction	Yes	Pretrained	0.469	44.11	0.761	52.88	0.899
Bi-direction	No	Pretrained	0.423	46.13	0.782	56.60	0.924
Inverse	Yes	Pretrained	0.448	43.34	0.765	51.28	0.915
Bi-direction	Yes	Scratch	0.463	40.32	0.729	46.82	0.886

Table 5. Ablation study on ProteinGym and inverse protein folding. Sizes of all models listed here are 150M. Inverse means the model is pretrained with only (structure → profile → sequence) order.

Interestingly, in the inverse protein folding task (**Table 5**), the two-modality model (Profile = No) achieved the highest average amino acid recovery (AAR) and self-consistency TM-score (scTM), and unexpectedly outperformed the three-modality model (Profile = Yes) using the internally generated evolutionary profile. A possible explanation is that this task requires the model to generate the evolutionary profile internally, and the current quality of these generated profiles may not yet be sufficient. These profiles may introduce systematic noise that propagates through the reasoning process, ultimately reducing the model's ability to accurately reconstruct sequences. This observation aligns with previous findings in the structure prediction task, further indicating that improving the quality of internally inferred profiles is essential to enhancing performance in future iterations of the model.

Training strategies include the directional flow of reasoning (*Direction*) and the two-stage pretraining process (*Initialization*).

Directional training flow influenced performance in a task-dependent manner. The forward-only model (Direction = Forward), pretrained with data in only (sequence → profile → structure) order, consistently outperformed bi-directional models in structure prediction (**Table 4**), and the inverse-only model (Direction = Inverse) also outperformed their bi-directional counterparts in most cases (**Table 5**). These patterns indicate that a unidirectional reasoning path is more efficient and better suited for tasks with a single modality focus. In contrast, bi-directional models achieved the best performance on the ProteinGym benchmark, suggesting that the cross-modal generalization provided by bi-directional flows is particularly advantageous for protein understanding. Overall, these observations suggest that while unidirectional reasoning suffices for tasks with a single modality focus, bi-directional flows may

offer superior performance in tasks requiring broader optimization and generalization capabilities.

Next, we evaluated the role of two-stage pretraining. Across all tasks, models initialized from pretrained sequence-only backbones (Initialization = Pretrained) significantly outperformed those trained from scratch (Initialization = Scratch). For example, on the PDB Date Split set, the bi-directional, profile-augmented model trained from scratch achieved TM-scores of only 0.715/0.380, whereas the same architecture with 2-stage pretraining achieved 0.782/0.634 (**Table 4**). The performance gap was especially pronounced in the internally inferred mode, where training from scratch led to a severe drop, highlighting the critical importance of pretraining for supporting effective internal profile generation. This trend was consistent across structure prediction and inverse folding tasks. For fitness prediction in the ProteinGym benchmark, the benefit from two-stage pretraining was relatively modest. This may be because the task emphasizes ranking mutation effects over precise sequence recovery, allowing scratch-trained models to capture sufficient sequence-structure compatibility for reasonable performance. This explanation is also supported by our scaling comparisons, where 150M and 650M model gives similar ProteinGym performance. Nonetheless, two-stage pretraining still provides consistent improvements, indicating its general value across tasks.

Collectively, these ablation studies provide strong evidence supporting the design principles of ProteinReasoner. Incorporating the evolutionary profile as a reasoning step consistently enhances model performance across diverse protein tasks, establishing it as a critical component of the framework. Nevertheless, improving the generation quality of the internally inferred evolutionary profiles remains a key area for future work to further strengthen the model's reasoning process and overall performance.

2.4 Applying Chain-of-Thought to Protein Optimization

Protein optimization is a central task in protein engineering that involves systematically improving the properties of proteins, such as stability, activity, specificity, or binding affinity[12]. This process typically requires generating and evaluating large libraries of protein variants through cycles of mutagenesis, selection, and characterization to identify sequences with enhanced functional performance. It is often framed as an iterative search for improved functional variants, where each round of experimentation informs the design of subsequent variants.

To streamline and accelerate this process, machine learning (ML) and generative methods have been increasingly adopted to predict promising mutations and guide sequence selection based on past experimental data[19,32]. Active learning[33] (AL), in particular, has become a popular strategy in this context (**Figure 3A**). In AL pipelines, ML models iteratively select the most informative or high-potential sequences for experimental validation, then use the fitness measurements from these sequences to iteratively update the models. This feedback loop aims to maximize efficiency by focusing experimental efforts on the most impactful sequence variants, thus reducing the total number of experiments required to achieve optimization goals.

Despite the promise of active learning strategies, they present critical drawbacks in protein optimization workflows. Fine-tuning machine learning models between rounds to generate improved sequences is not an optimal use of the limited labeled data obtained from iterative experimental validation. Moreover, fine-tuning often demands extensive hyperparameter tuning and manual oversight to avoid performance degradation, which hampers scalability and limits automation.

Given ProteinReasoner's demonstrated capability in fitness prediction and structured reasoning, we sought to leverage its chain-of-thought (CoT) design to enable protein optimization via in-context learning (ICL)[34]. This strategy provides a more efficient protein optimization paradigm that learns from prior examples presented in context, without modifying model weights. This approach fully exploits the generative capacity of foundation models to guide sequence design through successive rounds of directed evolution. Furthermore, it significantly reduces both computational overhead and experimental cost by eliminating the need for model retraining.

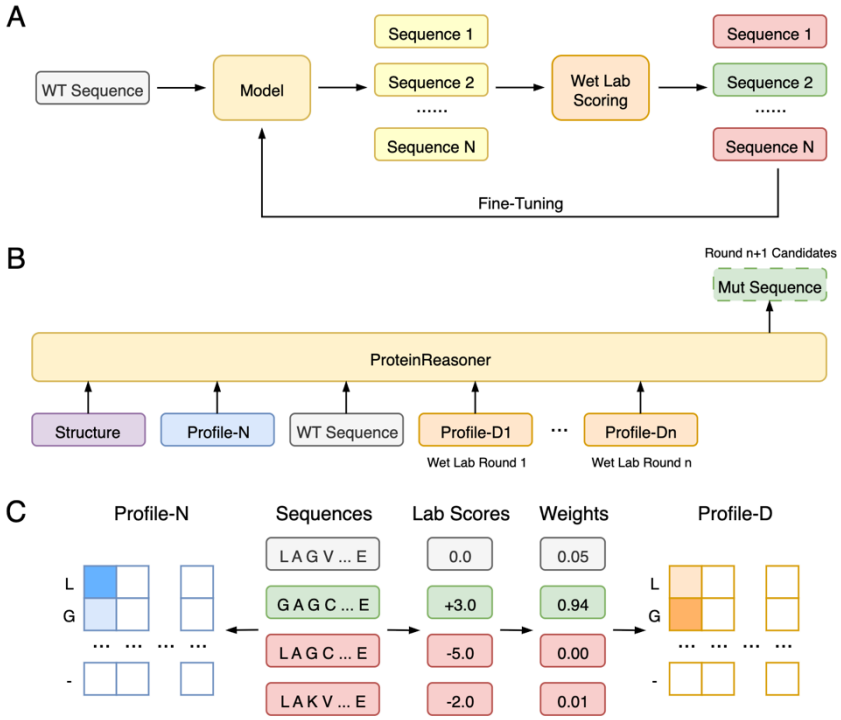


Figure 3. ProteinReasoner's in-context learning paradigm for protein optimization. (A) Visualization of the conventional active learning (AL) pipeline. The model generates sequence variants from a wild-type (WT) sequence, which are experimentally scored and then used to fine-tune the model for subsequent rounds. **(B)** Visualization of ProteinReasoner's in-context learning (ICL) paradigm. ProteinReasoner integrates structure, natural evolutionary profile (*Profile-N*), wild-type sequence, and directed evolution profiles from prior rounds (*Profile-D1*, *Profile-D2*, etc.) to directly generate optimized mutant sequences, to better guide the model with prior examples without fine-tuning. **(C)** Construction of the score matrix (directed evolutionary profile). Unlike natural profiles, the directed profile aggregates frequencies of amino acids weighted by experimental fitness scores, forming a score-informed evolutionary profile that guides the next round of sequence generation.

To achieve this, we extended the inverse folding input chain (*structure* → *evolutionary profile* → *wild-type sequence*) by appending a series of evolutionary profiles that represent the directed evolution[35] trajectory at each round of optimization with wet lab feedback (**Figure 3B**). To construct these directed evolutionary profiles, we designed the score matrix which utilized experimentally scored sequences and calculated a weighted amino acid frequency matrix at each residue position (**Figure 3C**). This score matrix encodes the evolutionary trajectory by emphasizing the enrichment of beneficial mutations across optimization rounds. In this ICL setup, the model implicitly reasons about potential future beneficial mutations along these trajectories and generates new candidate sequences with improved properties. This design allows ProteinReasoner to internalize and extend the optimization process without requiring explicit parameter updates, enabling efficient, flexible, generalizable, and automated

protein optimization.

To transition from theoretical design to practical implementation, we developed a prototype to empirically validate the feasibility of our in-context learning (ICL) framework. Given the current lack of publicly available multi-round protein optimization datasets, we constrained our implementation to a two-round optimization setting, utilizing a single round-1 score matrix to encode directed evolution improvements, and predicting the round-2 sequences. This design allowed us to isolate and rigorously assess the core reasoning capabilities of the model within the practical limits of existing data resources.

We selected the ProteinReasoner-150M pretrained model as the initialization for our prototype. However, the model had not been previously exposed to the five-modality input chain required for this task (*structure → evolutionary profile → wild-type sequence → evolutionary profile → mutant sequence*). To bridge this distributional gap, we employed supervised fine-tuning (SFT) to condition the model on the new multi-modal input format, enabling it to effectively parse and integrate the five-modality chain during inference. Following this, we applied direct preference optimization (DPO)[36] to align the model's generative policies with optimization-specific reward signals, reinforcing its capacity to preferentially generate beneficial sequence variants from directed evolution prompts. (see **Methods** for more details)

In the following section, we present evaluations that substantiate the effectiveness of this in-context protein optimization framework. Through comparative experiments, we assess the model's performance against traditional active learning pipelines, demonstrating the practical advantages of ProteinReasoner's reasoning-driven ICL optimization approach.

2.5 Efficient Protein Optimization via In-Context Learning

To evaluate the practical benefits of ProteinReasoner's reasoning-driven in-context learning (ICL) framework for protein optimization, we conducted a series of comparative experiments using the Megascale dataset[37]. Megascale is a large-scale protein fitness dataset encompassing 479 proteins, each with extensive mutational scans, including single and combination mutations, totaling more than 776k sequences. For evaluation, we adopted standard dataset splits that separated training, validation, and held-out test sets based on sequence clustering to ensure minimal sequence similarity between training and testing proteins.

Prior to evaluating the ICL framework, we benchmarked ProteinReasoner against existing stability prediction models, including ProteinDPO[38] and ThermoMPNN[39]. In these experiments, ProteinReasoner was fine-tuned using only three input modalities (structure → profile → sequence) via SFT and DPO, with identical dataset splits reported in ThermoMPNN. Our results (**Table S1**) demonstrate that ProteinReasoner consistently outperforms both baselines in terms of Spearman correlation, evaluated separately on all mutant sequences and on double-mutant sequences.

A key experimental design decision during training and inference was how to select sequences to construct the score matrix. We experimented with several selection strategies and ultimately chose to use the top 50 single-mutation sequences with the highest predicted probabilities from the pretrained model (see details in **Table S2-S3**). This approach ensured that the prior examples provided to the model were highly probable sequences according to the pretrained distribution, and it mimics the real-world scenario where round 0 sequence generation can only depend on the initial model. During evaluation,

the top 50 mutant sequences served as the prior examples to form the score matrix (**Figure 3C**), and evaluation metrics were calculated on the remaining sequences. Our evaluation metrics included Spearman correlation to assess the rank-order agreement between model-predicted sequence probabilities and experimentally measured stability changes ($\Delta\Delta G$), and AUROC to quantify the model's ability to distinguish between more stable and less stable variants.

To ensure a fair comparison, we benchmarked ProteinReasoner's ICL-based optimization approach against a conventional active learning (AL) pipeline. For the AL baseline, we fine-tuned ProteinReasoner-150M in the inverse folding direction (*structure* → *profile* → *sequence*), so that more stable sequences were assigned higher probabilities by the sequence head output. For each test protein, we incorporated the same 50 prior sequences used in the ICL model into the AL training set and conducted the fine-tuning with SFT and DPO.

The test set consisted of 63 proteins, and we trained 63 individual AL models, each corresponding to a test protein. These models were used to predict sequence probabilities and were evaluated using the same metrics applied to the ICL model. We maintained consistent hyperparameter settings across all ICL and AL models. For the 63 AL models, we selected the best-performing epoch based on validation set performance, whereas for the only one ICL model, we consistently used the final training epoch without any checkpoint selection or hyperparameter tuning, preserving the simplicity and generality of the ICL approach.

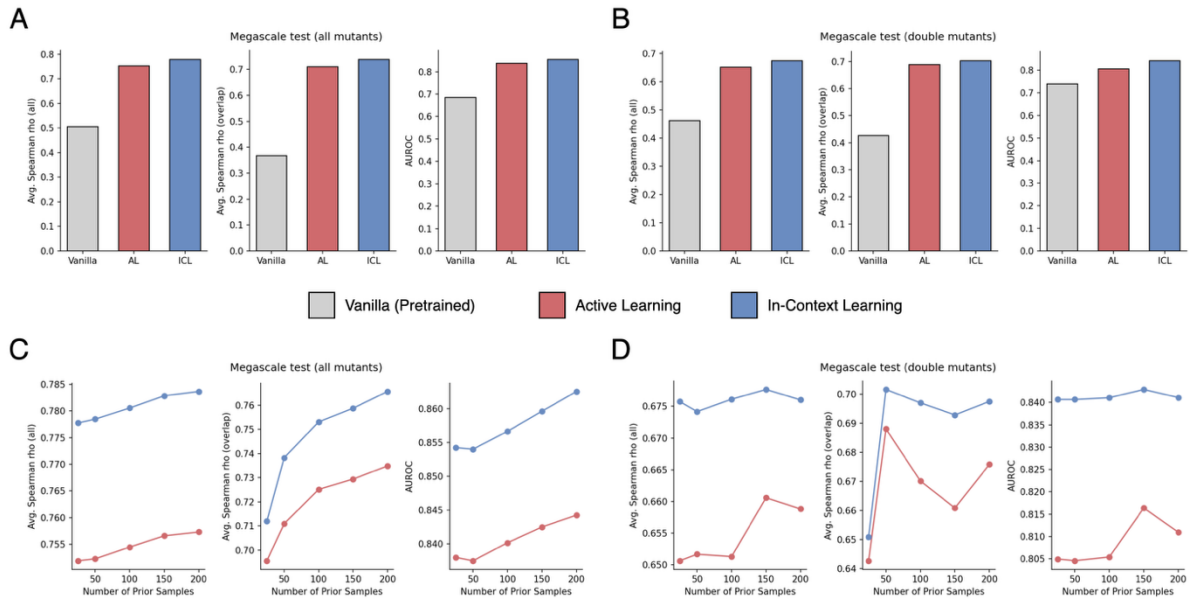


Figure 4. ProteinReasoner's in-context learning enables efficient protein optimization on the Megascale benchmark. (A) Performance comparison on the Megascale test set (all mutants). Metrics include average Spearman correlation (all mutants), Spearman correlation (mutants overlapping prior examples), and AUROC. **(B)** Performance comparison on double mutant subsets of the Megascale test set, using the same evaluation metrics as in A. **(C)** Impact of increasing the number of prior samples on ICL and AL performance for all mutants. Metrics are the same as in A. **(D)** Impact of increasing the number of prior samples on ICL and AL performance for double mutants. Metrics are the same as in A.

As shown in **Figure 4A**, both the AL baseline and the ICL model significantly outperformed the pretrained model (Vanilla), demonstrating the effectiveness of fine-tuning for protein optimization. In this context, "Spearman rho (all)" refers to the correlation between sequence probability and $\Delta\Delta G$

across all sequences, while "Spearman rho (overlap)" measures the correlation between $\Delta\Delta G$ and sequences that share the same mutation position(s) as the prior examples included in the score matrix. Despite being trained without hyperparameter tuning or checkpoint selection, the ICL model (Spearman rho = 0.778 / 0.738, AUROC = 0.854) consistently surpassed the AL baseline (Spearman rho = 0.752 / 0.711, AUROC = 0.838) across all evaluation metrics. This result highlights the ICL model's ability to generalize from provided prior sequences to other mutations and, importantly, to extrapolate from mutation positions present in the priors to entirely different positions within the sequence.

We further assessed model performance specifically for the prediction of stability in combination mutations. As shown in **Figure 4B**, we evaluated all models exclusively on double mutant sequences across 21 proteins. In this more challenging setting, the ICL model (Spearman rho = 0.674 / 0.702, AUROC = 0.841) continued to outperform both the AL (Spearman rho = 0.652 / 0.688, AUROC = 0.805) and Vanilla models (Spearman rho = 0.463 / 0.427, AUROC = 0.740), with particularly substantial gains observed in AUROC. This result underscores the strength of the chain-of-thought reasoning embedded within the ICL framework, which is effective in capturing and generalizing the complex, non-linear interactions that characterize combinatorial mutation landscapes.

We also evaluated the impact of support set size on model performance by varying the number of prior examples provided to the ICL and AL models. Across support set sizes ranging from 25 to 200 prior examples, performance steadily improved for both the ICL and AL models as more prior examples were provided (**Figure 4C**). This observation aligns with expectations that additional context enhances model generalization. Importantly, at each support set size, the ICL model consistently outperformed the AL model, indicating that the ICL framework is more effective in leveraging increasing amounts of prior information for protein optimization. When evaluating the models specifically on double mutant sequences (**Figure 4D**), performance trends exhibited greater variability. Spearman rho fluctuated across support set sizes, and in some instances, the AL model temporarily outperformed the ICL model in rank correlation. However, across all settings, the ICL model consistently achieved higher AUROC scores than the AL model, suggesting superior classification performance even in combinatorial mutation scenarios.

These findings indicate that ProteinReasoner's ICL framework scales effectively with increasing support set sizes and consistently outperforms AL baselines across full mutation sets and double mutant combinations. Importantly, the ICL model achieved this superior performance without requiring hyperparameter tuning or checkpoint selection. This highlights the ICL model's ability to generalize from limited prior examples and effectively handle complex combinatorial mutation landscapes. ProteinReasoner's reasoning-driven ICL framework offers a practical, efficient, and scalable solution for accelerating protein engineering workflows. It better leverages the capabilities of foundation models by explicitly reasoning through prior examples to support protein optimization tasks.

3. Discussion

ProteinReasoner introduces a novel reasoning-driven approach to multi-modal protein language modeling by explicitly integrating evolutionary profiles as an intermediate step in a chain-of-thought (CoT) framework. This design enables the model to capture biologically meaningful reasoning flows across sequence and structure modalities. Through a comprehensive suite of evaluations, we demonstrate that ProteinReasoner achieves strong zero-shot performance across structure prediction,

inverse folding, and fitness prediction tasks, consistently outperforming the larger ESM3-Open 1.4B and DPLM-2 baselines despite using substantially fewer parameters. Additionally, ProteinReasoner enables efficient protein optimization via In-context learning (ICL), offering a practical alternative to computationally intensive active learning pipelines. Together, these results validate the core design principles of ProteinReasoner and highlight its potential for advancing protein modeling and engineering workflows.

The key conceptual innovation of ProteinReasoner is the explicit use of evolutionary profiles as a latent reasoning bridge between sequence and structure modalities. Unlike prior models that treat evolutionary information as auxiliary input features, ProteinReasoner centralizes the evolutionary profile within the reasoning chain. This design strengthens multi-modal integration and enhances the model's performance across diverse tasks by leveraging structured reasoning steps.

Compared to existing multi-modal PLMs, ProteinReasoner offers several distinct advantages. First, it consistently achieves strong zero-shot performance across multiple protein tasks despite having significantly fewer parameters than competing models like ESM3-Open 1.4B. This efficiency arises from ProteinReasoner's reasoning-driven architecture, which structures modality interactions in a task-relevant manner rather than simply fusing inputs at the embedding layer. Second, the model's flexible chain-of-thought framework extends naturally to protein optimization tasks via In-context learning, offering a more efficient way to learn from prior examples than the active learning paradigm. Third, ProteinReasoner is uniquely positioned to reason over numerical matrices, as the evolutionary profile is inherently a structured numerical input. This capacity to handle quantitative intermediate representations distinguishes it from models optimized solely for token-based modalities.

ProteinReasoner's reasoning-centric framework has the potential to significantly streamline protein engineering workflows by supporting generalizable, efficient, and scalable protein optimization through In-context learning. While this study validated the ICL-based optimization framework in a single-round setting, extending the model to support multi-round protein optimization remains a critical future direction. Developing datasets that capture multi-round directed evolution trajectories will allow for more comprehensive evaluations of ProteinReasoner's capacity to reason across iterative optimization cycles and predict beneficial mutations over extended experimental horizons. Future research should also explore test-time scaling[40,41] strategies to maximize the practical utility of ProteinReasoner's ICL framework. Specifically, forming an internal loop within the computational modeling step that iteratively refines the generated mutant sequences before selecting candidates for wet lab validation could significantly improve efficiency and design quality. Together, these advancements could enable effective, scalable, and automated protein engineering solutions.

4. Methods

4.1 ProteinReasoner Model Architecture

Input and tokenizers

ProteinReasoner processes three modalities as input sequences: structure, evolutionary profile, and amino acid sequence, concatenated in either folding or inverse folding order. The 3D structure, represented by residue-level atom coordinates extracted from PDB files, is tokenized into discrete tokens (vocabulary size = 8192) using the DPLM-2 structure tokenizer, with the checkpoint sourced

from the Hugging Face *airkingbd/struct_tokenizer* model. The amino acid sequence is tokenized at the residue level using a 26-token amino acid vocabulary plus 11 special tokens. Multiple sequence alignments (MSAs) in A3M format are processed to compute per-position amino acid frequencies, with an additional column representing the gap character. To signal modality boundaries to the model, we introduce modality-specific <beginning of sequence> and <end of sequence> tokens at the start and end of each modality segment.

Natural Evolutionary Profile (Pretraining Phase). For each target protein sequence, we construct an evolutionary profile from its homologous sequences identified via multiple sequence alignment (MSA). Specifically, we use an A3M file containing n aligned sequences $\{X_1, X_2, \ldots, X_n\}$, each of length L. The evolutionary profile is represented as a matrix $\boldsymbol{P} \in \mathbb{R}^{L\times 21}$, where each entry $\boldsymbol{P}_{ij}$ denotes the frequency of amino acid A_j (including one special gap character "-") at position i across the aligned sequences:

$$\boldsymbol{P}_{ij} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = A_j)$$

Here, $\mathbb{I}(\cdot)$ is the indicator function, $A_j \in A \cup \{-\}$ and A denotes the set of 20 standard amino acids.

Directed Evolutionary Profile (Protein Optimization). During protein optimization, we compute a directed evolutionary profile to encode mutational preferences informed by prior experimental feedback.

Let $\{X_1, X_2, \ldots, X_m\}$ denote a set of mutant sequences (including the wild-type sequence) with associated softmax-normalized stability scores $\{s_1, s, \ldots, s_m\}$. The directed profile $\boldsymbol{D} \in \mathbb{R}^{L\times 21}$ is defined as a weighted frequency matrix, where each entry $\boldsymbol{D}_{ij}$ reflects the stability-weighted presence of amino acid A_j at position i:

$$\boldsymbol{D_{ij}} = \frac{1}{Z_i}\sum_{k=1}^{m} s_k \cdot \mathbb{I}(X_{k,i} = A_j), Z_i = \sum_{k=1}^{m} s_k$$

The WT sequence is included in this set to ensure that baseline information about the unmutated protein is preserved. This profile encodes an empirical evolutionary trajectory shaped by selective pressure, and serves as an intermediate reasoning modality that guides the generation of improved variants in ProteinReasoner's in-context learning framework.

Input Embeddings

Prior to entering the Transformer layers, tokenized or preprocessed inputs are projected into $\mathrm{L} \times \mathrm{D}$ embedding representations. Discrete modalities (structure and sequence) are embedded using *nn.Embedding* layers, which are *PyTorch* modules designed to map discrete tokens to dense vector representations. The numerical matrix (evolutionary profile) is embedded using *nn.Linear* projections. The sequence modality embedding is initialized from the pretrained sequence-only model, while all other input and output embedding layers are initialized from scratch.

Transformer Layers

The modality-specific embeddings are concatenated into a $3\mathrm{L} \times \mathrm{D}$ embedding sequence and passed through N Transformer layers. No attention masking is applied between modalities, allowing the

model to freely learn cross-modal dependencies. We apply rotary positional encoding, reusing position indices within each modality. The positional encodings are shared across all modalities, meaning the same position indices are used regardless of modality type. Model architecture and hyperparameter details are provided in **Supplementary Table S4**.

Output Decoders

The output embeddings from the last Transformer layer are segmented by modality and passed through separate modality-specific decoder heads to predict structural tokens, amino acid tokens, and evolutionary profiles.

Protein Optimization Models

For the ProteinReasoner model adapted for the In-context Learning (ICL) protein optimization, the input sequence is extended to include an additional directed evolutionary profile and a subsequent sequence, each with their own <beginning> and <end> tokens to mark their boundaries. All other components of the model architecture remain consistent with the pretrained ProteinReasoner.

4.2 Model Training

Generative Pretraining Task

We trained ProteinReasoner and its derivative models using the autoregressive framework that predicts the next token within the same modality until the <*EOS*> token is reached. The pretraining loss combines Cross-Entropy Loss for the structure and sequence modalities with KL Divergence for the evolutionary profile, encouraging the predicted profiles to closely match the target distributions derived from MSAs. The detailed loss formulation is provided below:

$$L = w_s \times L_{CE}(Structure) + w_r \times L_{CE}(Sequence) + w_p \times L_{KL}(Profile)$$

Where w_s, w_r, w_p are the weighting coefficients for the structure, sequence, and profile losses, respectively. We set $w_s = 1.0, w_r = 1.0, w_p = 2.0$ for all models.

Pretraining Details

Models were trained using our customized *Megatron-Deepspeed* framework to enable large-scale parallel pretraining. We employed data parallelism across GPUs to distribute training efficiently. We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, a weight decay of 0.1, and applied dropout with a rate of 0.1 to stabilize training and prevent overfitting. The maximum learning rate was set to $2e^{-3}$ and the minimum learning rate to $2e^{-5}$. The learning rate was linearly increased during the initial 0.62% of the total steps to the maximum learning rate, then gradually decreased to the minimum learning rate using a cosine scheduler. Training was conducted over 600k steps with a batch size of 1024. All training was performed using BF16 mixed precision to improve computational efficiency. We trained ProteinReasoner-150M using 32 Nvidia A800 GPUs for approximately 3 days; ProteinReasoner-650M using 32 Nvidia A800 GPUs for approximately 6 days.

Protein Optimization: Supervised Fine-Tuning (SFT)

For the protein optimization tasks, we fine-tuned all ProteinReasoner models (models aligned with ThermoMPNN, Active Learning (AL), and In-context Learning (ICL) models) for 10 epochs. Each

epoch consisted of randomly selecting 128 positive ($\Delta\Delta G > 0$) single-mutation sequences for each wild-type protein screening in the training set, resulting in 128 × n samples per epoch, where n is the number of wild-type proteins. We used a batch size of 64 and a learning rate of $1e^{-5}$ for fine-tuning.

Protein Optimization: Direct Preference Optimization (DPO)

We employed Direct Preference Optimization[36] (DPO) to further align the SFT model. We hypothesized that compared to traditional supervised fine-tuning, the relative framing of positive-negative mutation pairs can allow our model to learn a more generalizable fitness landscape and smoothen potential noise in $\Delta\Delta G$ measurements.

Specifically, given a preference pair (y_w, y_l), where y_w is the winner sequence and y_l is the loser sequence, DPO updates the model π_θ using the following loss function:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)}[log\sigma\,(\beta log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}) - log\sigma\,(\beta log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)})]$$

This loss function effectively guides the model to maximize the likelihood of stabilizing mutations; while making sure the model does not deviate too much from the reference model which is itself.

During DPO training of both AL and ICL models, we randomly selected 128 sequence pairs for each wild-type protein screening, resulting in 128×n pairs per epoch. Both single and double mutants were included in the pair selection to ensure diversity across mutation types. Each pair included one preferred and one less preferred sample, with a minimum $\Delta\Delta G$ difference of 0.001 to ensure meaningful preference distinctions. All DPO training sessions were conducted for 10 epochs with a batch size of 64, a learning rate of $1e^{-5}$, and a beta value of 0.1.

4.3 Datasets

Pretraining Datasets

The Stage 1 pretraining dataset is detailed in the original publication[23]. For Stage 2 pretraining, we used a combined dataset consisting of 9,450,396 proteins from the AlphaFold Database (AFDB) and 310,862 chains from the PDB-REDO with experimentally resolved structures. AFDB proteins were sourced from 2.3 million FoldSeek[42] non-singleton clusters. Proteins from the AFDB with an average pLDDT score below 70 were excluded to ensure structural reliability. Additionally, any protein present in any of the downstream testing sets was removed from the training data to prevent data leakage. Multiple sequence alignments (MSAs) were generated using HHblits[43] (version 3.3.0) with the following parameters: -n 2 -cov 70 -id 90 -diff 2000, aligning protein sequences to the UniRef30 database. Proteins with fewer than 10 aligned sequences were filtered out to ensure sufficient evolutionary context.

CASP and CAMEO Datasets

We curated a test set comprising 194 single-chain proteins from CAMEO, 50 from CASP14, and 53 from CASP15 for evaluation in folding and inverse folding tasks. Structural data for these proteins were obtained from experimentally resolved PDB files, and MSAs were generated using MMSeqs2[44] against the UniRef30 database, following the same procedure as ColabFold[45] (https://github.com/sokrypton/ColabFold/blob/64c8b2fe2ecf3199bc2d3c60b5da4b929a41086e/colabfold_search.sh).

PDB Date Split Datasets

We obtained 449 single-chain protein IDs from the MultiFlow[31] study's GitHub repository and downloaded the corresponding PDB files and sequences from the Protein Data Bank. MSAs were generated using MMSeqs2 alignments against the UniRef30 database. For the inverse folding task, we successfully converted the PDB files of 393 proteins into ESM3 structure tokens, which constituted the final dataset used for this task.

ProteinGym Datasets

ProteinGym datasets were downloaded from the official ProteinGym website, including PDB structures, MSA query results, wild-type sequences, and mutational screening CSV files. We computed the evolutionary profiles using RSALOR[46] without combining Relative Solvent Accessibility from the structure data.

Megascale Datasets

The Megascale dataset was pre-clustered based on wild-type protein sequence similarity. These clusters were then split into training (411), validation (4), and test (63) sets to minimize sequence similarity between training and testing samples. MSAs were generated using MMSeqs2 alignments against the UniRef30 database. For 159 out of 478 proteins lacking MSA coverage, we generated the natural evolutionary profiles using the ProteinReasoner-650M model. Protein structures for this dataset were predicted using AlphaFold2. The score matrix used in the In-context Learning prompts were weighted position-wise amino acid frequency matrix, with weights derived from the softmax-normalized $\Delta\Delta G$ labels to reflect mutation preferences.

To ensure a fair comparison with ThermoMPNN and ProteinDPO, we trained our models using the same data split (training: 238 proteins, validation: 31, test: 27) as reported in the official ThermoMPNN GitHub repository, which was also adopted by ProteinDPO.

4.4 Evaluation

ESM3 Baseline

We used the ESM3-Open-1.4B model, downloaded from the Hugging Face repository (esm3-sm-open-v1). All evaluations were performed following the official examples provided in the ESM GitHub repository README. For the structure prediction task, we provided the sequence track to generate the structure and saved the output as a PDB file. For the inverse protein folding task, we tokenized the provided atomic coordinates into structure tokens and used the model to generate the corresponding sequence track. For both tasks, we evaluated ESM3 under two generating modes: num_steps = 1 (Iterative) and num_steps = sequence length (Argmax).

DPLM-2 Baseline

We used the DPLM-2 650M and 3B models, downloaded from the Hugging Face repository (*airkingbd/dplm2_650m*, *airkingbd/dplm2_3b*). All evaluations were performed using the official GitHub repository script *generate_dplm2.py* with default settings. For both structure prediction and inverse protein folding tasks, we evaluated DPLM-2 with argmax decoding for 100 sampling iterations.

Structure Prediction Evaluation

Predicted structure tokens from each benchmarked models were decoded back into 3D atomic coordinates using their corresponding structure token decoder. The predicted structures were compared to the ground truth PDB structures using *US-align* (Version 20220924) with the "-outfmt 2" option to calculate both the Root Mean Square Deviation (RMSD) and the TM-score.

Inverse Folding Evaluation

We calculated the amino acid recovery rate (AAR) using the following formula:

$$AAR = \frac{1}{L}\sum_{i=1}^{L} I(a_i^{pred} = a_i^{true})$$

where L is the sequence length, a_i^{pred} is the predicted amino acid at position i, a_i^{true} is the true amino acid at position i, and $\mathbb{I}(\cdot)$ is the indicator function that equals 1 when the predicted and true amino acids match.

Due to computational resources and time constraints, we randomly selected 50 samples from each evaluation dataset. For each sample, we used AlphaFold2 to predict the structure from the model-generated sequence and assessed structural similarity to the original structure using the TM-score, calculated by *US-align* (Version 20220924) with the "-outfmt 2" option.

ProteinGym Evaluation

To quantify the fitness impact of mutations, we computed a log-odds score based on the model's predicted distributions over amino acids at each residue position. Let $\ell_i^{prof}(a)$ and $\ell_i^{seq}(a)$ denote the predicted logits for amino acid a at position i from the profile and sequence heads, respectively. We first normalized the logits using a log softmax function and summed the two heads to obtain a combined score:

$$s_i(a) = log\left(\frac{\ell_i^{prof}(a)}{\sum_{a'} \ell_i^{prof}(a')}\right) + log\left(\frac{\ell_i^{seq}(a)}{\sum_{a'} \ell_i^{seq}(a')}\right)$$

Given a mutation from the wild-type amino acid a^{WT} to a mutant amino acid a^{mut} at position i, the fitness score $F_i^{a^{mut}}$ for the mutation is defined as the log-odds ratio:

$$F_i^{a^{mut}} = s_i(a^{mut}) - s_i(a^{WT})$$

For multiple mutations (e.g., double mutants), we assumed additive contributions and computed the total fitness score as the sum over individual positions. We generated prediction CSV files in accordance with the ProteinGym GitHub repository's required format and used the official ProteinGym evaluation script to ensure that our results were directly comparable to other models listed on the official leaderboard.

Protein Optimization Evaluation

For protein optimization tasks, we selected the top N sequences with the highest predicted probability from the pretrained model to construct the score matrix. The remaining mutants in each protein screening were used to compute Spearman's correlation and AUROC, and we reported the mean Spearman's rho and AUROC across all test set proteins. To ensure fair comparison between Active Learning (AL) and In-Context Learning (ICL) models, we maintained consistent evaluation sets across all experimental settings.

Author Contributions

C.L., L.C., and X.Z. conceived the research. M.Y., S.J., H.W., Z.G., C.L, and L.C. built the pretraining infrastructure. C.L., and L.C. performed model pretraining, fine-tuning and evaluation. C.L., L.C., and Y.G. performed dataset preprocessing and curation. T.J. implemented the DPO algorithm. X.Z. supervised the research. C.L. wrote the initial version of the manuscript. All authors wrote the final manuscript.

References

1. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021).
2. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science (1979)* **379**, 1123–1130 (2023).
3. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods* **16**, 1315–1322 (2019).
4. Notin, P. *et al.* ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. in *Advances in Neural Information Processing Systems* (eds. Oh, A. et al.) vol. 36 64331–64379 (Curran Associates, Inc., 2023).
5. Hayes, T. *et al.* Simulating 500 million years of evolution with a language model. *Science (1979)* **387**, 850–858 (2025).
6. Li, M. *et al.* ProSST: Protein Language Modeling with Quantized Structure and Disentangled Attention. in *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (2024).
7. Su, J. *et al.* SaProt: Protein Language Modeling with Structure-aware Vocabulary. *bioRxiv* (2023) doi:10.1101/2023.10.01.560349.
8. Wang, X. *et al.* DPLM-2: A Multimodal Diffusion Protein Language Model. Preprint at https://arxiv.org/abs/2410.13782 (2024).
9. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
10. Marks Debora S. AND Colwell, L. J. A. N. D. S. R. A. N. D. H. T. A. A. N. D. P. A. A. N. D. Z. R. A. N. D. S. C. Protein 3D Structure Computed from Evolutionary Sequence Variation. *PLoS One* **6**, 1–20 (2011).
11. Morcos, F. *et al.* Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences* **108**, E1293–E1301 (2011).
12. Listov, D., Goverde, C. A., Correia, B. E. & Fleishman, S. J. Opportunities and challenges in design and optimization of protein function. *Nat Rev Mol Cell Biol* **25**, 639–653 (2024).
13. Fang, X. *et al.* A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nat Mach Intell* **5**, 1087–1096 (2023).
14. Wu, R. *et al.* High-resolution de novo structure prediction from primary sequence. *bioRxiv* (2022) doi:10.1101/2022.07.21.500999.
15. Wang, W., Peng, Z. & Yang, J. Single-sequence protein structure prediction using supervised transformer protein language models. *Nat Comput Sci* **2**, 804–814 (2022).
16. Gong, J. *et al.* Steering Protein Family Design through Profile Bayesian Flow. Preprint at https://arxiv.org/abs/2502.07671 (2025).
17. Wei, J. *et al.* Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. in

Advances in Neural Information Processing Systems (eds. Koyejo, S. et al.) vol. 35 24824–24837 (Curran Associates, Inc., 2022).

18. Brown, T. *et al.* Language Models are Few-Shot Learners. in *Advances in Neural Information Processing Systems* (eds. Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F. & Lin, H.) vol. 33 1877–1901 (Curran Associates, Inc., 2020).
19. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst* **12**, 1026-1045.e7 (2021).
20. Henning, N. J. *et al.* Deubiquitinase-targeting chimeras for targeted protein stabilization. *Nat Chem Biol* **18**, 412–421 (2022).
21. Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M. & Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nat Methods* **18**, 389–396 (2021).
22. Wang, C. *et al.* Chain-of-Modality: Learning Manipulation Programs from Multimodal Human Videos with Vision-Language-Models. Preprint at https://arxiv.org/abs/2504.13351 (2025).
23. Cheng, X. *et al.* Training Compute-Optimal Protein Language Models. Preprint at https://arxiv.org/abs/2411.02142 (2024).
24. Varadi, M. *et al.* AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Res* **52**, D368–D375 (2024).
25. Joosten, R. P., Long, F., Murshudov, G. N. & Perrakis, A. The PDB_REDO server for macromolecular structure model optimization. *IUCrJ* **1**, 213–220 (2014).
26. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res* **28**, 235–242 (2000).
27. Jänes, J. & Beltrao, P. Deep learning for protein structure prediction and design—progress and applications. *Mol Syst Biol* **20**, 162–169 (2024).
28. Dauparas, J. *et al.* Robust deep learning–based protein sequence design using ProteinMPNN. *Science (1979)* **378**, 49–56 (2022).
29. Chen, L. *et al.* Learning protein fitness landscapes with deep mutational scanning data from multiple sources. *Cell Syst* **14**, 706-721.e5 (2023).
30. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics* **57**, 702–710 (2004).
31. Campbell, A., Yim, J., Barzilay, R., Rainforth, T. & Jaakkola, T. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design. Preprint at https://arxiv.org/abs/2402.04997 (2024).
32. Frey, N. C. *et al.* Lab-in-the-loop therapeutic antibody design with deep learning. *bioRxiv* (2025) doi:10.1101/2025.02.19.639050.
33. Tharwat, A. & Schenck, W. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics* **11**, 820 (2023).
34. Dong, Q. *et al.* A Survey on In-context Learning. Preprint at https://arxiv.org/abs/2301.00234 (2024).
35. Cobb, R. E., Chao, R. & Zhao, H. Directed evolution: Past, present, and future. *AIChE Journal* **59**, 1432–1440 (2013).

36. Rafailov, R. *et al.* Direct Preference Optimization: Your Language Model is Secretly a Reward Model. Preprint at https://arxiv.org/abs/2305.18290 (2024).

37. Tsuboyama, K. *et al.* Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 434–444 (2023).

38. Widatalla, T., Rafailov, R. & Hie, B. Aligning protein generative models with experimental fitness via Direct Preference Optimization. *bioRxiv* (2024) doi:10.1101/2024.05.20.595026.

39. Dieckhaus, H., Brocidiacono, M., Randolph, N. Z. & Kuhlman, B. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the National Academy of Sciences* **121**, e2314853121 (2024).

40. Muennighoff, N. *et al.* s1: Simple test-time scaling. Preprint at https://arxiv.org/abs/2501.19393 (2025).

41. Bio, N. & Biswas, S. De novo design of epitope-specific antibodies against soluble and multipass membrane proteins with high specificity, developability, and function. *bioRxiv* (2025) doi:10.1101/2025.01.21.633066.

42. van Kempen, M. *et al.* Fast and accurate protein structure search with Foldseek. *Nat Biotechnol* **42**, 243–246 (2024).

43. Remmert, M., Biegert, A., Hauser, A. & Söding, J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods* **9**, 173–175 (2012).

44. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* **35**, 1026–1028 (2017).

45. Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nat Methods* **19**, 679–682 (2022).

46. Tsishyn, M., Hermans, P., Pucci, F. & Rooman, M. Residue conservation and solvent accessibility are (almost) all you need for predicting mutational effects in proteins. *bioRxiv* (2025) doi:10.1101/2025.02.03.636212.

Supplementary Materials

Model	Spearman – all mutants	Spearman – double mutants
ProteinReasoner-150M	0.727	0.492
ProteinReasoner-650M	**0.738**	**0.540**
ThermoMPNN	0.725	0.438
ProteinDPO	\	0.468

Supplementary Table S1. Comparison between stability prediction models. All models are trained and evaluated using the same Megascale data split. We got the performance of ThermoMPNN and ProteinDPO from the corresponding paper, and ProteinDPO didn't report the Spearman's correlation on all mutant sequences.

Model	Megascale – All Mutants (n = 63)		
	Spearman - all	Spearman - overlap	AUROC
Pretrain	0.505	0.368	0.684
Baseline (AL)	0.752	0.711	0.838
ICL (Random)	**0.780**	**0.741**	**0.855**
ICL (Positive)	0.777	0.738	0.850
ICL (Logits)	0.778	0.738	0.854

Supplementary Table S2. Benchmark on Megascale Holdout – All Mutants. We evaluated three different design choices of the score matrix. *Random* means we randomly select N sequences as the prior examples to form the score matrix; *Positive* means we randomly choose N sequences that have $ddG > 0$; *Logits* means we choose the top N sequences that have the highest predicted probability by the pretrained ProteinReasoner-150M model. During inference, all ICL models used the logits mode since this simulates the real case scenario, and we can align the comparison between different models.

Model	Megascale – Double Mutants (n = 21)		
	Spearman - all	Spearman - overlap	AUROC
Pretrain	0.463	0.427	0.740
Baseline (AL)	0.652	0.688	0.805
ICL (Random)	**0.677**	0.706	0.833
ICL (Positive)	0.671	**0.709**	0.823
ICL (Logits)	0.674	0.702	**0.841**

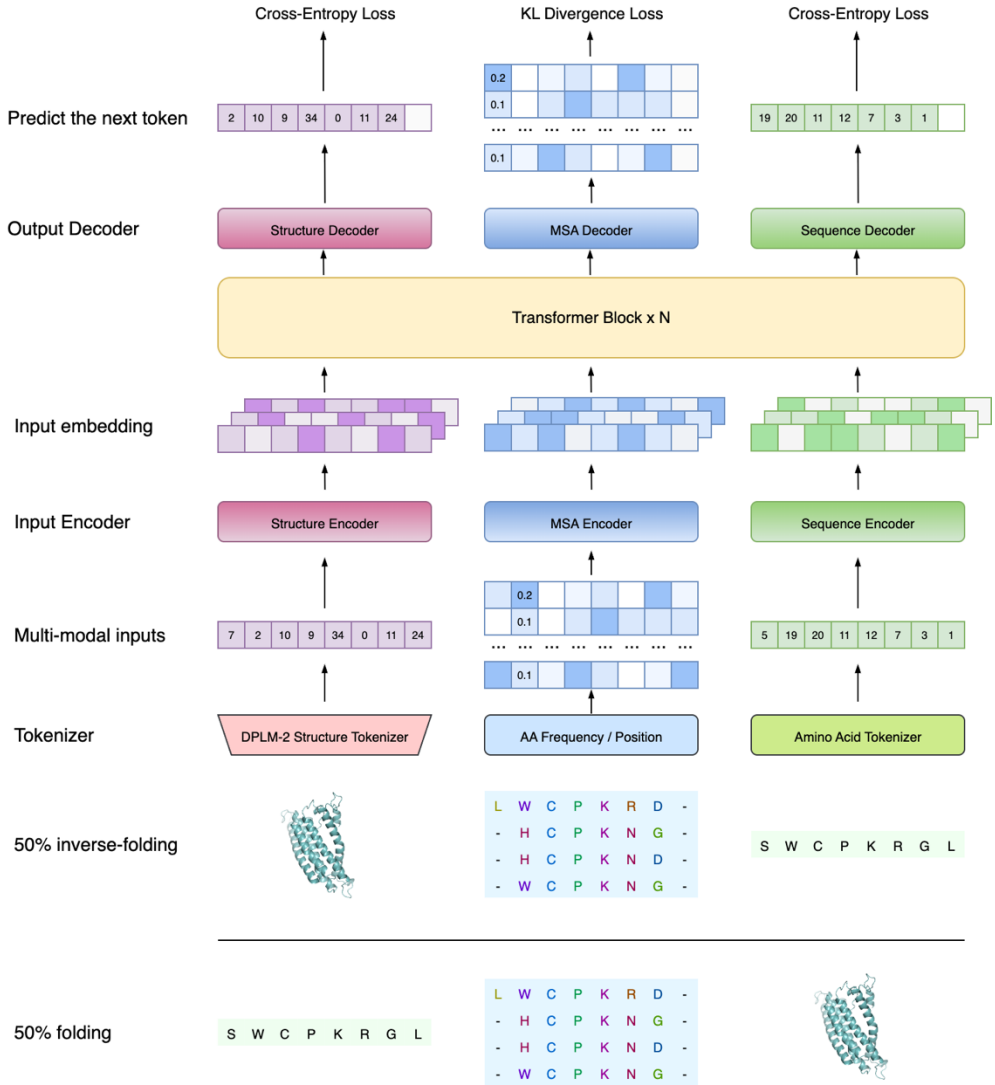
Supplementary Table S3. Benchmark on Megascale Holdout – Double Mutants.

Model	Layers	Dim	FFN Dim	Heads	Dropout
ProteinReasoner-150M	30	640	1706	20	0.1
ProteinReasoner-650M	33	1280	3424	20	0.1

Supplementary Table S4. Model architecture details. *Layers*: number of transformer layers; *Dim*: Hidden embedding dimension; *FFN Dim*: Feedforward layer intermediate dimension; *Heads*: number of attention heads.

Model	Loss Wts	Max LR	Min LR	Warmup	Steps	Batch	Seq Len
ProteinReasoner-150M	1,2,1	2e-3	2e-5	1.25%	60k	1024	3072
ProteinReasoner-650M	1,2,1	2e-3	2e-5	1.25%	60k	1024	3072
No Profile Models	1,1	2e-3	2e-5	2.5%	30k	1024	2048
150M from Scratch	1,2,1	4e-4	1e-5	2.5%	30k	1024	3072

Supplementary Table S5. Pretraining hyperparameters. *Loss Wts:* Weights for structure, profile, and sequence losses; *Warmup:* Percentage of steps used for learning rate warmup; *Steps:* Total number of training steps.



Supplementary Figure 1. ProteinReasoner model architecture. ProteinReasoner processes three input modalities: structure, evolutionary profile (MSA), and sequence, each tokenized and encoded independently. The model maintains modality separation throughout the transformer layers. Separate decoders predict the next token for each modality, using cross-entropy loss for structure and sequence predictions, and KL divergence loss for the evolutionary profile prediction. Training is balanced between folding and inverse folding directions, each comprising 50% of training batches.