From Structure to Function: Preference Alignment for Function-aware Protein Inverse Folding

Tamatgar Nilufer*
Centre for Frontier AI Research, Institute of High Performance Computing, A*STAR
Singapore, Singapore
nilufer2703@gmail.com

Soobin Park*
Yonsei University
Seoul, Republic of Korea
clapkong@gmail.com

Yinghua Yao*
Centre for Frontier AI Research, Institute of High Performance Computing, A*STAR
Singapore, Singapore
yao_yinghua@a-star.edu.sg

Xixian Chen
Singapore Innovation of Food and Biotechnology, A*STAR
Singapore, Singapore
xixian_chen@a-star.edu.sg

Yuangang Pan
Centre for Frontier AI Research, Institute of High Performance Computing, A*STAR
Singapore, Singapore
yuangang.pan@gmail.com

Abstract

Protein inverse folding models conditioned on structure achieve high sequence recovery but often fail to preserve biological function due to the lack of functional supervision. We propose a function-aware preference alignment framework that improves functional preservation by fine-tuning models to favor function-preserving sequences over function-disrupting alternatives, avoiding the need for explicit function optimization. Our approach constructs reliable preference pairs in silico using hypothesis-driven perturbations of critical residues and model-consistent likelihood constraints, enabling scalable supervision without additional wet-lab measurements. The resulting framework guides protein sequence design models toward generating sequences that better preserve functional integrity, while remaining compatible with existing inverse folding pipelines such as ProteinMPNN and ESM-IF. Extensive experiments on protein design benchmarks and enzyme datasets with established wet-lab validation show that our fine-tuned models consistently outperform pretrained counterparts in preserving functional integrity during protein sequence design.

CCS Concepts

• **Applied computing** → **Computational biology; Protein design**; • **Computing methodologies** → **Deep learning**.

Keywords

Protein Inverse Folding, Preference Alignment, Function-aware Protein Design, Preference Optimization.

*Equal contribution.

KDD'26, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Tamatgar Nilufer, Soobin Park, Yinghua Yao, Xixian Chen, and Yuangang Pan. 2026. From Structure to Function: Preference Alignment for Function-aware Protein Inverse Folding. In *Proceedings of the 32th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'26).* ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent progress in protein foundation models has enabled direct protein sequence design conditioned on structural inputs, significantly accelerating in silico protein engineering. Models such as ProteinMPNN [5] and ESM-IF [11] take backbone geometry or structural representations as input and generate amino acid sequences that are statistically compatible with the given fold. These approaches have achieved success in sequence recovery; however, the resulting sequences are not guaranteed to be functionally active [35]. Protein function is often governed by a small subset of critical residues [14], and perturbations at these sites[1] can abolish activity even when global structure is preserved [6].

A major challenge in function-aware protein design [16] is the lack of scalable supervision. Functional annotations are primarily obtained through wet-lab experiments, which are expensive and time-consuming [36]. Consequently, large-scale datasets that associate protein sequences or structures with functional measurements are limited. While computational evaluators have been proposed to guide protein design, their use still requires experimental validation [3, 12], and no single unified function evaluator exists that generalizes across diverse protein families and biochemical tasks [27, 33].

In this work, we propose an alternative and scalable paradigm that avoids directly optimizing protein function. Instead, we fine-tune protein design models to move away from residue substitutions that are likely to disrupt function. By discouraging such deleterious mutations, the model implicitly biases generation toward

[1]Throughout this work, we use "critical sites" to denote sequence positions in a computational or modeling context, and "critical residues" to refer to their biological interpretation as functionally indispensable amino acids.

sequences that better preserve functional integrity, without requiring explicit functional labels [1, 13]. This formulation naturally reframes function optimization as a preference alignment problem, where function-preserving sequences are preferred over function-disrupting variants under the same structural context. To enable this, we leverage established bioinformatics tools and pretrained inverse folding models to construct function-aware supervision signals. These signals are derived from hypothesis-driven perturbations of critical residues [22, 24] and model-consistent likelihood constraints [2, 23], allowing us to synthesize reliable preference pairs in silico without requiring new wet-lab measurements during training.

Concretely, we use SIFT (Sorting Intolerant From Tolerant) [15, 28], a conservation-based mutation impact predictor, to identify functionally critical residues in protein sequences. SIFT exploits evolutionary conservation and amino acid similarity to distinguish tolerated substitutions from those likely to disrupt protein function. Based on SIFT, we generate function-aware preference pairs consisting of the original sequence and a negative variant produced by introducing deleterious substitutions at critical sites. Importantly, these negative sequences are constrained to remain statistically plausible under pretrained inverse folding models, ensuring that the supervision reflects functional degradation rather than trivial sequence implausibility. We incorporate the constructed function-aware preference pairs into a preference-based fine-tuning framework for ProteinMPNN and ESM-IF. By emphasizing critical-residues recovery and suppressing non-functional alternatives, the model is explicitly guided to prioritize functionally critical residues while preserving global structural compatibility.

The contributions of this work are summarized as follows:

- We reformulate function preservation in protein design as a function-aware preference alignment (FPA) framework, allowing inverse folding models to prefer function-preserving sequences over function-disrupting variants using in silico supervision.
- We propose a hypothesis-driven, in silico supervision strategy for constructing function-aware preference pairs, removing the reliance on explicit functional labels or additional wet-lab measurements.
- Our FPA framework provides a scalable and model-agnostic mechanism that encourages protein design models to generate function-preserving sequences, while remaining compatible with widely used inverse folding pipelines such as ProteinMPNN and ESM-IF.
- Extensive evaluations conducted both in silico and on enzyme datasets with established wet-lab measurements demonstrate that our FPA framework more effectively preserves functional properties during protein sequence design than pretrained baselines and existing fine-tuning methods.

2 Related Work

Protein Inverse Folding. Protein inverse folding aims to generate amino acid sequences compatible with a given target structure and has been substantially advanced by deep learning methods. Among existing approaches, ProteinMPNN [5] and ESM-IF [11] have become widely adopted backbones, leveraging message-passing architectures and pretrained protein language models, respectively, to learn structure-conditioned sequence distributions. These models serve as standard baselines across inverse folding benchmarks and are extensively used in downstream protein design pipelines. More recent methods, such as PiFold [8] and InstructPLM [18], further improve benchmark performance through enhanced geometric representations or instruction tuning.

Preference Optimization for Protein Inverse Folding. Preference-based optimization methods, including policy optimization and Direct Preference Optimization (DPO) [19], have recently emerged as effective tools for aligning protein inverse folding models with target properties beyond sequence recovery [9]. Early work such as ProteinDPO [30] applied DPO to fine-tune ESM-IF [11], encouraging preference for stabilizing over destabilizing variants. In parallel, similar paradigms have also been applied to ProteinMPNN [5]: Xu et al. [34] leveraged DPO with feedback from protein folding models, while Park et al. [17] explored DPO-based optimization to enhance sequence diversity in peptide design. A more recent work [35] further moved beyond sequence recovery by incorporating foldability signals, such as AlphaFold pLDDT, to directly optimize protein designability and improve in silico folding success. Beyond offline preference optimization, ProteinZero [29] introduced an online reinforcement learning framework for inverse folding, enabling scalable multi-objective optimization with efficient structural feedback and diversity regularization.

However, existing preference optimization methods largely focus on structural or stability-related objectives, leaving functional constraints underexplored.

3 Function-aware Protein Inverse Folding

In the following, we first introduce protein inverse folding. Building on this, we formulate the problem of function-aware inverse folding. We then discuss why commonly used direct preference optimization yields limited improvements in this setting. This motivates our function-aware preference alignment framework, designed to handle preference signals arising from diverse and mechanistically distinct functional perturbations.

3.1 Preliminaries

The goal of protein Inverse Folding (IF) is to generate an amino acid sequence $y = (y_1, y_2, \ldots, y_L)$ by taking a backbone structure x as input, where L is the sequence length, $y_i \in \mathcal{A}$ and the amino acid set $\mathcal{A} = \{\text{ACDEFGHIKLMNPQRSTVWY}\}$. Generally, the protein IF models [5, 11] are trained to predict the sequence y from the backbone structure x via an auto-regressive way:

$$\pi(y \mid x) = p(y_1 \mid x) \prod_{i=2}^{L} p(y_i \mid y_{<i}; x). \tag{1}$$

While most protein IF models are trained by maximizing sequence recovery, i.e., $\max \pi(y \mid x)$, this objective is inherently limited. In practice, it mainly promotes global structural compatibility and overall sequence recovery, with limited ability to account for functional constraints that are essential for biological activity beyond the given backbone geometry.

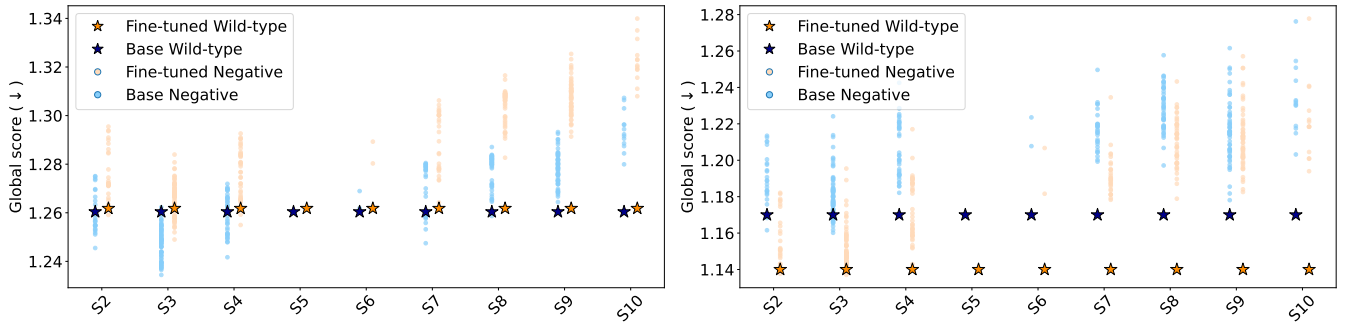

Figure 1: Stage-wise (Stage 2 (S2) to Stage 10 (S10)) discrimination of inactive mutations on the pMT enzyme data with established wet-lab validation for ProteinMPNN (Left) and ESM-IF (Right). The stars denote the wild-type sequence (Stage 1). The points represent experimentally validated inactive (negative) variants. The global score denotes the model's output negative log-likelihood (the lower the better). Compared with the base model, the FPA fine-tuned model has higher probability generating wild-type sequences than negative ones, thus avoiding generating inactive sequences. See Section 4.5 for more details.

It is well established that not all residues contribute equally to protein function. A small subset of functionally critical residues, e.g., active-site or binding residues, largely determines functional activity (See Fig. 1), whereas the majority of residues are more tolerant to substitutions and have minimal impact on function.

As a result, optimizing inverse folding models solely for overall sequence recovery often fails to produce functionally competent sequences, even when high sequence recovery scores are achieved. This explains why protein sequence design based on inverse folding models still relies heavily on extensive wet-lab screening to identify functional variants, leading to substantial experimental cost.

3.2 Identification of Functionally Critical Sites

While large-scale functional annotation via wet-lab experiments is desirable but cost-prohibitive, we introduce a lightweight computational strategy to derive function-aware supervision from biological priors.

In enzymology, it is well established that protein function is often governed by a small subset of functionally important residues, whose perturbation can disproportionately impair structural stability, catalytic activity, or molecular recognition [25, 26]. Motivated by this observation, we formalize such positions at the sequence level as critical sites, which serve as the basis for our function-aware preference construction.

Definition 1 (Functionally Critical Sites)**.** Given a protein sequence y of length L, the critical sites are defined as a subset of residue positions $\mathcal{F}(y) \subseteq \{1, 2, \ldots, L\}$ such that, $\forall i \in \mathcal{F}(y)$, substitutions at position i are expected to result in a substantial degradation of protein function—as quantified by a scoring function ϕ. Namely,

$$\mathcal{F}(y) = \{i \in \{1, 2, \ldots, L\} \mid \phi(i \mid y, x) > \tau\},$$

where $\phi(i \mid y, x) \in \mathbb{R}$ is a residue-level scoring function that aggregates mutation-induced functional effects at residue i, and τ is a predefined threshold.

SIFT-based functional impact scoring. To instantiate the scoring function ϕ in a biologically grounded yet scalable manner, we adopt **SIFT** (Sorting Intolerant From Tolerant) [15, 28], a widely used evolutionary conservation–based predictor of mutation impact. Formally, given a protein sequence y, SIFT provides substitution-level tolerance scores $s(i, a)$ for mutating the wild-type residue y_i at position i to amino acid $a \in \mathcal{A} \setminus \{y_i\}$. Lower scores indicate a higher likelihood that the substitution is functionally deleterious.

To obtain a residue-level functional impact score, we aggregate substitution-level intolerance across all possible mutations at each position. Specifically, we define the SIFT-based functional impact score as

$$\phi_{\text{SIFT}}(i \mid y) = \frac{1}{|\mathcal{A} \setminus \{y_i\}|} \sum_{a \in \mathcal{A} \setminus \{y_i\}} \mathbb{I}\big[s(i, a) \leq \epsilon\big],$$

where $\mathbb{I}[\cdot]$ denotes the indicator function and ϵ is set to 0.05 following the standard SIFT criterion for deleterious substitutions.

This formulation measures the fraction of possible amino acid substitutions at position i that are predicted to be functionally intolerant, thereby capturing the overall sensitivity of that residue to mutation. Accordingly, critical sites for sequence y are identified as

$$\mathcal{F}(y) = \{i \in \{1, 2, \ldots, L\} \mid \phi_{\text{SIFT}}(i \mid y) > \tau\}, \tag{2}$$

where τ controls the stringency of functional intolerance.

3.3 Function-aware Preference Pairs

Building on the identified functionally critical sites as biological priors, we construct synthetic function-aware preference pairs that can be directly used for model training.

Assumption 2 (Hypothetical Function-aware Preference Pair)**.** Given a protein sequence y, let $\mathcal{F}(y)$ denote its functionally critical sites. We hypothesize that introducing deleterious substitutions at one or more functionally critical sites $\mathcal{F}(y)$ is likely to result in a degradation of protein function[2]. Accordingly, a synthetically perturbed sequence y^- can be constructed as follows:

$$y_i^- = \begin{cases} a, \ a \in \mathcal{A} \setminus \{y_i\}, & i \in \mathcal{F}(y), \\ y_i, & i \notin \mathcal{F}(y), \end{cases} \tag{3}$$

[2]This hypothesis is consistent with established observations in protein biology that perturbations at conserved or functionally critical residues can disproportionately affect functional activity.

Then, we define a hypothetical function-aware preference pair (y^+, y^-), where the wild-type sequence y is treated as the positive sequence y^+ that is known to be functionally competent, and y^- is hypothesized to be function-disrupting due to perturbations at critical residues.

Synthetic perturbations introduced at functionally critical sites $\mathcal{F}(y)$ provide a natural way to construct negative sequences. However, not all such synthetically perturbed sequences are informative for fine-tunings IF models. In practice, many negative sequences are trivially implausible, as they fail to remain compatible with the given backbone structure and can be easily rejected by the IF model itself. Including such trivial negatives provides limited training signal and does not meaningfully improve functional discrimination.

This observation motivates the construction of hard negative sequences, defined as negative variants that remain structurally plausible under the given backbone yet are incorrectly favored by the base IF model. To identify such hard negatives, we leverage the log-likelihood scores produced by the base IF model as a proxy for backbone compatibility. Since IF models are trained to assign higher likelihood to sequences consistent with a given backbone, the model likelihood naturally reflects structural plausibility under the design objective.

Definition 3 (Hard Negative Sequence Mining)**.** Given a backbone structure x, a positive sequence y^+, and a set of synthetically perturbed sequences $\mathcal{Y}^- = \{y^-\}$ generated by mutating residues at critical sites $\mathcal{F}(y^+)$, we define a sequence $y^- \in \mathcal{Y}^-$ as a hard negative if it satisfies

$$\log \pi(y^- \mid x) \;\geq\; \log \pi(y^+ \mid x) + \upsilon, \tag{4}$$

where $\pi(y \mid x)$ denotes the conditional likelihood assigned by the base IF model, and υ is a tolerance threshold.

By emphasizing hard negative sequences that remain structurally plausible yet are incorrectly favored by the model, this mining strategy concentrates training on biologically meaningful failure cases. The resulting preference pairs (y^+, y^-) therefore provide a weak but biologically grounded supervision signal for protein function, enabling effective fine-tuning of protein sequence design models without relying on explicit functional annotations, structure prediction pipelines, or additional wet-lab experiments.

Detailed descriptions of the data construction process are provided in Section A of the Appendix.

3.4 Function-aware Preference Alignment

According to Definition 3, for each backbone x, we obtain a preference pair (y^+, y^-), where y^+ is a function-preserving sequence (the ground-truth/wild-type) and y^- is a synthetically perturbed variant that is hypothesized to disrupt protein function. We now describe how to use these signals to fine-tune protein IF models.

Let $\pi_\theta(y \mid x)$ denote a parameterized IF model that generates a protein sequence y conditioned on a backbone structure x, and let $\pi_{\text{ref}}(y \mid x)$ denote a fixed reference model initialized from the pretrained IF model.

3.4.1 Direct Preference Optimization (DPO). A widely adopted paradigm for preference-based fine-tuning is DPO, which increases the likelihood of outputs preferred according to a predefined target criterion (e.g., stability, foldability, or diversity), while suppressing dispreferred ones relative to a reference model [19]. Under a Bradley–Terry assumption, the DPO loss for a preference pair (y^+, y^-) conditioned on x is given by

$$\ell_{\text{DPO}}(x, y^+, y^-) = -\log \sigma(a - b), \tag{5}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid. a, b refer to the relative log-likelihood ratios, i.e.,

$$a = \beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)}, \quad b = \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}, \tag{6}$$

and β is a scaling factor, controlling the alignment strength.

Recent studies have shown that the majority of DPO variants suffer from the squeezing effect [21], whereby the likelihoods of both preferred and dispreferred samples are reduced, with probability mass shifted toward unobserved argmax candidates. In protein IF models, such likelihood displacement can be undesirable, as positive sequences typically correspond to ground-truth or wild-type proteins and should not be penalized during fine-tuning.

Moreover, DPO implicitly assumes that preference data follow a single, transitive ranking criterion [31]. This assumption may not hold for our function-aware preferences, where function disruption can arise from diverse and non-comparable mechanisms, including impaired structural stability, catalytic activity, or molecular recognition. As a result, directly applying DPO may fail to achieve the desired alignment and can even degrade performance.

3.4.2 Preference Optimization Based on SPPO. To address the limitations of DPO, which adjusts only the likelihood gap (i.e., the log-likelihood difference) between preferred and dispreferred sequences, Wu et al. [32] proposed self-play preference optimization (SPPO). SPPO frames preference learning as an alignment problem with respect to the model's own distribution. Formally, given a preference oracle $P(y \succ y' \mid x)$, SPPO fits the log-likelihood ratio of a candidate sequence to the preference signal via a quadratic regression objective:

$$\ell_{\text{SPPO}}(x, y \mid \pi_t) = \Big[\beta \log \frac{\pi_\theta(y \mid x)}{\pi_t(y \mid x)} - \big(P(y \succ \pi_t \mid x) - \frac{1}{2}\big)\Big]^2,$$

where $P(y \succ \pi_t \mid x) = \mathbb{E}_{y' \sim \pi_t(\cdot|x)}[P(y \succ y' \mid x)]$ and π_t denotes the current policy.

Unlike DPO, which optimizes only the log-likelihood difference, SPPO applies separate updates to the preferred and dispreferred samples, thereby mitigating the squeezing effect [21]. Such decoupled updates are particularly important in protein sequence design, where preference supervision is biologically heterogeneous.

Moreover, for a given backbone structure x, we construct a single function-aware preference pair (y^+, y^-), where the ground-truth sequence y^+ is functionally competent and the perturbed variant y^- is hypothesized to be function-disrupting, which induces a deterministic preference label:

$$\mathbb{P}(y^+ \succ y^- \mid x) = 1, \quad \mathbb{P}(y^- \succ y^+ \mid x) = 0.$$

Under this hard-label, single-pair regime, the self-play preference score reduces to $P(y^+ \succ \pi_t \mid x) = 1$ and $P(y^- \succ \pi_t \mid x) = 0$, and the SPPO objective simplifies to a symmetric pairwise alignment loss. Building on this observation, we propose our Function-aware Preference Alignment (FPA) framework tailored to hypothesis-driven

functional signals in protein inverse folding (See Appendix Section B for details.). Let a, b denote the relative log-likelihood ratios defined in Eq. (6). The FPA loss is given by

$$\ell_{\text{Seq}}(x, y^+, y^-) = \left(a - \frac{1}{2}\right)^2 + \left(b + \frac{1}{2}\right)^2, \tag{7}$$

which explicitly encourages increasing the likelihood of the function-preserving sequence y^+ while suppressing that of the function-disrupting variant y^-, both measured relative to a frozen reference IF model.

Intuitively, Eq. (7) reshapes the local likelihood landscape around functionally critical residues, biasing sequence generation toward functional integrity while preserving the structural compatibility and background statistics learned by the pretrained model. This makes our FPA well suited for fine-tuning inverse folding models under biologically motivated, non-transitive, and in silico preference supervision.

3.4.3 Critical-site-aware Preference Alignment. A key biological property exploited by FPA is that functional constraints are typically localized to a small subset of residues, while the majority of positions primarily contribute to structural stability or sequence background. To reflect this locality, FPA supports residue-level preference alignment by restricting preference-driven updates to functionally critical sites, while leaving functionally neutral regions largely unconstrained.

Therefore, we define a residue-level preference objective by applying the preference alignment loss exclusively to the functionally critical sites indexed by $\mathcal{F}(y^+)$:

$$\ell_{\text{Residue}}(y^+, y^-, x, \mathcal{F}) = \left(a_{\mathcal{F}} - \frac{1}{2}\right)^2 + \left(b_{\mathcal{F}} + \frac{1}{2}\right)^2, \tag{8}$$

where the relative log-likelihood ratios are also calculated at functionally critical sites, i.e.,

$$a_{\mathcal{F}} = \beta \log \frac{\pi_\theta(y^+_{\mathcal{F}} \mid x)}{\pi_{\text{ref}}(y^+_{\mathcal{F}} \mid x)}, \quad b_{\mathcal{F}} = \beta \log \frac{\pi_\theta(y^-_{\mathcal{F}} \mid x)}{\pi_{\text{ref}}(y^-_{\mathcal{F}} \mid x)}.$$

This formulation localizes preference gradients to functionally critical residues, directly promoting accurate recovery and discrimination at residues most relevant to protein function, while avoiding unnecessary perturbations to the global sequence distribution.

While residue-level preference alignment focuses learning on critical sites, optimizing only these positions can still induce drift in non-critical regions through shared model parameters. To preserve the generative behavior and structural compatibility learned by the pretrained inverse folding model, we introduce an auxiliary cross-entropy (CE) objective over the positive sequences only:

$$\ell_{\text{CE}}(y^+, x) = -\sum_{i=1}^{L} \log \pi_\theta(y_i^+ \mid x). \tag{9}$$

Rather than encoding functional preference, this term acts as a stabilizing anchor that preserves the pretrained generative prior, allowing controlled deviations at critical residues while limiting undesired degradation of sequence plausibility or structural compatibility elsewhere.

The final training objective for our FPA framework combines localized preference alignment with global generative regularization:

$$\mathcal{L}_{\text{FPA}} = \sum_{(y^+, y^-, x)} \left[\ell_{\text{Res}}(y^+, y^-, x, \mathcal{F}) + \lambda \ell_{\text{CE}}(y^+, x)\right], \tag{10}$$

where $\lambda \geq 0$ controls the strength of the generative regularization. Together, Eq. (10) enables FPA to integrate localized biological priors into preference-based optimization, achieving function-aware fine-tuning of protein sequence design models entirely through in silico supervision.

The overall training procedure focuses on preference-driven updates at positions where positive and negative sequences differ, encouraging the model to increase the likelihood of function-preserving variants while suppressing function-disrupting alternatives relative to a frozen reference model. The scaling factor β modulates the contribution of each differing residue to the total FPA loss, allowing the strength of preference alignment to be tuned according to the desired sensitivity to functionally critical perturbations.

4 Experiment

In this work, we build upon ProteinMPNN [5] and ESM-IF [11] as representative base models to study function-aware inverse folding fine-tuning.

4.1 Experimental Setup

Dataset. All experiments are conducted on the CATH 4.2 40% non-redundant dataset [5], which serves as the pretrained data source for ProteinMPNN, and we follow the original data splitting protocol. Based on this dataset, we construct our function-aware preference dataset as described in Section A.1 in Appendix. To prevent data leakage due to partial overlap between the pretrained datasets of ProteinMPNN and ESM-IF, we apply an additional filtering step that enforces mutual exclusivity among the training, validation, and test splits. Dataset statistics after filtering are summarized in Table 4 in Appendix.

Evaluation Metrics. We evaluate our function-aware inverse folding using the following four metrics.

- **Perplexity (PPL)** measures the model's uncertainty in predicting the protein sequence y^+ given its backbone structure x. $\text{PPL}(y \mid x) = \mathbb{E}_{(y^+, y^-, x) \sim \mathcal{D}_{\text{test}}}\left[\exp\left(-\frac{1}{L}\sum_{i=1}^{L} \log \pi_\theta(y_i^+ \mid x)\right)\right]$, where $\pi_\theta(y_i^+ \mid x)$ is the model's probability for residue y_i^+ at position i.
- **Sequence recovery (Rec)** measures the average fraction of residues in generated sequences that match the ground-truth sequence under a given backbone structure: $\text{Rec} = \mathbb{E}_{(y^+, y^-, x) \sim \mathcal{D}_{\text{test}}}\left[\max_{k=1,\ldots,K} \frac{1}{L}\sum_{i=1}^{L} \mathbb{I}[\hat{y}_i^{(k)} = y_i^+]\right]$.
- **Critical Site Recovery (CSR)** measures the average recovery restricted to perturbed (critical) residues: $\text{CSR} = \mathbb{E}_{(y^+, y^-, x)}\left[\max_{k=1,\ldots,K} \frac{1}{|C|}\sum_{i \in C} \mathbb{I}[\hat{y}_i^{(k)} = y_i^+]\right]$.
- **Preference accuracy (ACC)** measures the fraction of preference pairs in which the likelihood of the positive sequence is higher than that of the negative variant under

Table 1: Evaluation of FPA for fine-tuning ProteinMPNN and ESM-IF on single-chain (Single) and multi-chain (Multi) test sets. Best results are marked in bold, and second-best results are underlined.

	Perplexity ↓		Sequence Recovery (%) ↑		Critical Site Recovery (%) ↑	
	Single	Multi	Single	Multi	Single	Multi
ProteinMPNN	4.55	4.03	50.40	51.04	16.93	16.73
+DPO	7.19	6.27	37.47	37.45	**21.64**	**22.78**
+SPPO	<u>4.44</u>	<u>3.94</u>	<u>50.64</u>	<u>51.33</u>	17.50	17.17
+FPA (ours)	**4.38**	**3.92**	**51.39**	**51.87**	<u>20.08</u>	<u>21.45</u>
ESM-IF	3.80	3.77	60.63	60.46	20.59	20.59
+DPO	7.69	7.80	42.52	42.04	**34.29**	**32.91**
+SPPO	<u>3.76</u>	<u>3.71</u>	<u>61.11</u>	<u>60.87</u>	23.90	23.04
+FPA (ours)	**3.41**	**3.50**	**62.63**	**61.81**	<u>31.22</u>	<u>30.50</u>

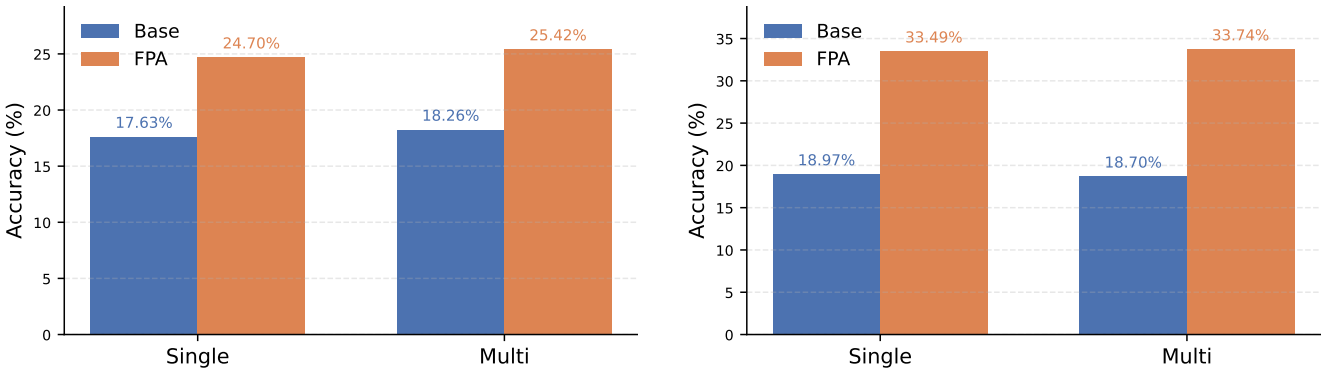

Figure 2: Pairwise preference accuracy of FPA fine-tuned vs. base models on single-chain and multi-chain test sets for ProteinMPNN (Left) and ESM-IF (Right), respectively.

the same backbone: $\mathrm{ACC} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\Big[\mathbb{I}\big[\log \pi_\theta(y^+ \mid x) > \log \pi_\theta(y^- \mid x)\big]\Big]$.

For notational convenience, we express all expectations as being taken uniformly over D_{test}, although in practice some metrics depend only on the ground-truth sequence y^+. To simplify notation, we denote the sequence length uniformly as L, despite variations across different sequences. For each test backbone structure x with a corresponding preference pair (y^+, y^-), all sequence-level metrics are computed using the full sequences without masking. For residue-level preference evaluation, we restrict the analysis to the set of perturbed (critical) positions $C = \{i \mid y_i^+ \neq y_i^-\}$, where function-aware mutations are introduced.

4.2 Benchmark Evaluation

Table 1 presents the quantitative evaluation results of FPA when fine-tuning ProteinMPNN and ESM-IF on the test sets, in comparison with baseline methods.

Table 1 show that: (1) FPA-based fine-tuning consistently achieves notable improvements over both base models across all three metrics. (2) While sequence recovery improves only modestly after fine-tuning, FPA consistently boosts recovery at functionally critical sites across both models, indicating that function-aware preference alignment focuses learning on biologically relevant residues without sacrificing global sequence fidelity or structural plausibility. (3) SPPO-based method yields consistent yet modest gains over the base model, this result underscores that our residue-level preference optimization better directs the model toward function critical sites, thereby enabling more efficient function alignment. (4) In contrast, DPO-based methods exhibit a clear trade-off, achieving relatively high critical site recovery at the cost of degraded perplexity and overall sequence recovery, which indicates reduced structural plausibility.

Preference accuracy reflects the model's ability to prioritize the generation of function-preserving sequences, thereby reducing the burden of wet-lab screening. Fig. 2 respectively presents the pairwise preference accuracy of the base models (ProteinMPNN and ESM-IF), as well as our FPA fine-tuned models, on the test set.

Fig. 2 shows that: (1) After fine-tuning, compared to the two base models, our FPA framework can significantly enhance the model's ability to perceive protein function, thereby increasing the likelihood of generating function-preserving sequences. (2) Both ProteinMPNN and ESM-IF exhibit relatively low pairwise preference accuracy. This is because only hard preference cases were selected, where both base models are prone to assign a higher likelihood to the function-disrupting sequence than to the function-preserving one.

Table 2: Ablation studies of key design choices in Eq. (10). "CE" indicates the use of the cross-entropy loss, whereas "Mask" indicates the adoption of the residue-level loss in Eq. (8).

Base Model	CE	Mask	Sequence Recovery (%) ↑		Critical Site Recovery (%) ↑	
			Single	Multi	Single	Multi
ProteinMPNN	✓		50.91	51.57	18.32	17.59
		✓	34.60	34.48	24.28	24.43
	✓	✓	51.39	51.87	20.08	21.45
ESM-IF	✓		63.16	62.35	27.85	26.50
		✓	41.81	41.40	35.51	35.63
	✓	✓	62.63	61.81	31.22	30.50

4.3 Ablation Study

We conduct ablation studies to examine the contribution of different components and the sensitivity to key hyperparameters. In particular, we analyze the effects of the cross entropy loss, the residue mask, and the hyperparameter λ.

From Table 2, we can find that: (1) using either the cross-entropy (CE) loss or the residue-level mask alone yields limited and unbalanced improvements: CE mainly benefits full sequence recovery but performs poorly on critical-site recovery, while the mask-only variant substantially improves critical-site recovery but sacrifices overall sequence recovery. (2) In contrast, combining CE with the residue-level mask leads to large and consistent gains on both sequence recovery and critical-site recovery in both single-chain and multi-chain settings, demonstrating strong complementarity between global sequence modeling and residue-level functional supervision.

Table 3 further studies the influence of λ, which balances sequence-level fidelity and residue-level preference optimization. We observe consistent trends across both backbones. (1) Setting $\lambda = 0$ (i.e., without cross-entropy) leads to a substantial improvement in critical-site recovery but scarifies sequence recovery. (2) Increasing λ improves sequence recovery but generally degrades critical-site recovery, suggesting that overly emphasizing the cross-entropy loss can weaken functional alignment. (3) Importantly, the optimal trade-off differs across backbones: $\lambda = 0.2$ provides the best balance for ProteinMPNN, whereas $\lambda = 1.0$ achieves the most favorable compromise for ESM-IF. Accordingly, we adopt $\lambda = 0.2$ for ProteinMPNN-based experiments and $\lambda = 1.0$ for ESM-IF-based experiments for comparison with other baselines.

4.4 Residue-level Predictive Distributions

To qualitatively examine how FPA alters residue-level modeling behavior, we visualize sequence logos derived from the output logits of the ProteinMPNN base model and the FPA fine-tuned model on a wild-type (WT) protein sequence in Fig. 3 (ESM-IF's result put in Appendix Fig. 4). Each logo summarizes the per-position amino-acid distribution predicted by the model. At each residue position, logos are constructed from the model output logits ($L \times 20$) converted into information-content representations.

Table 3: Ablation studies on λ in Eq. (10). We adopted $\lambda = 0.2$ for proteinMPNN and $\lambda = 1$ for ESM-IF as the setting we used for comparison with other baselines.

Base Model	λ	Sequence Recovery (%) ↑		Critical Site Recovery (%) ↑	
		Single	Multi	Single	Multi
ProteinMPNN	0.0	34.60	34.48	24.28	24.43
	0.2	51.39	51.87	20.08	21.45
	0.5	52.06	52.59	18.84	19.56
	1.0	52.30	52.78	18.69	19.42
	2.0	52.34	52.86	18.88	19.24
ESM-IF	0.0	41.81	41.40	35.51	35.63
	0.2	59.60	58.79	35.11	35.51
	0.5	62.89	62.15	29.89	29.58
	1.0	62.63	61.81	31.22	30.50
	2.0	63.00	62.13	29.91	29.38

Fig. 3 and Fig. 4 show that: (1) The base model distributes probability mass broadly across many amino acids, including at functionally critical sites, reflecting its objective of maximizing overall sequence recovery rather than prioritizing residues essential for function. Consequently, although the base model can achieve high global recovery, it may substitute critical residues and generate function-disrupting designs. (2) In contrast, the FPA fine-tuned model produces much sharper and more confident distributions at critical sites, assigns substantially higher probability to the WT residues, and corrects many mispredictions made by the base model, while remaining relatively flexible at non-critical sites. (3) Overall, these results demonstrate that FPA shifts the model's focus toward function-determining residues, enabling selective preservation of critical sites without sacrificing sequence diversity. This provides direct qualitative evidence that residue-level preference optimization promotes function-preserving protein sequence design.

4.5 Case Study with Wet-lab Validation

To evaluate whether FPA enhances the ability of inverse folding models to distinguish function-disrupting mutations from the wild-type sequence, we perform a stage-wise analysis on an in-house dataset on the enzyme promiscuous methyltransferase (pMT) [4, 20]. Starting from a WT enzyme (Stage 1), the dataset consists of nine rounds of expert-guided mutagenesis, where the best-performing variant from each round serves as the seed for the next. All variants are experimentally characterized by wet-lab assays, with few or no inactive variants observed in Stages 5–6.

For each stage, we treat the WT sequence as the reference point and all experimentally confirmed inactive variants as negative points. Each sequence–structure pair is evaluated by both the base model and the FPA fine-tuned model, where lower scores indicate better structural compatibility. Fig. 1 shows that: (1) Both FPA fine-tuned models consistently assign higher scores to negative variants than to the WT sequence across almost all stages, indicating an improved ability to recognize function-disrupting mutations. Moreover, the score gap increases in later stages, where more mutations

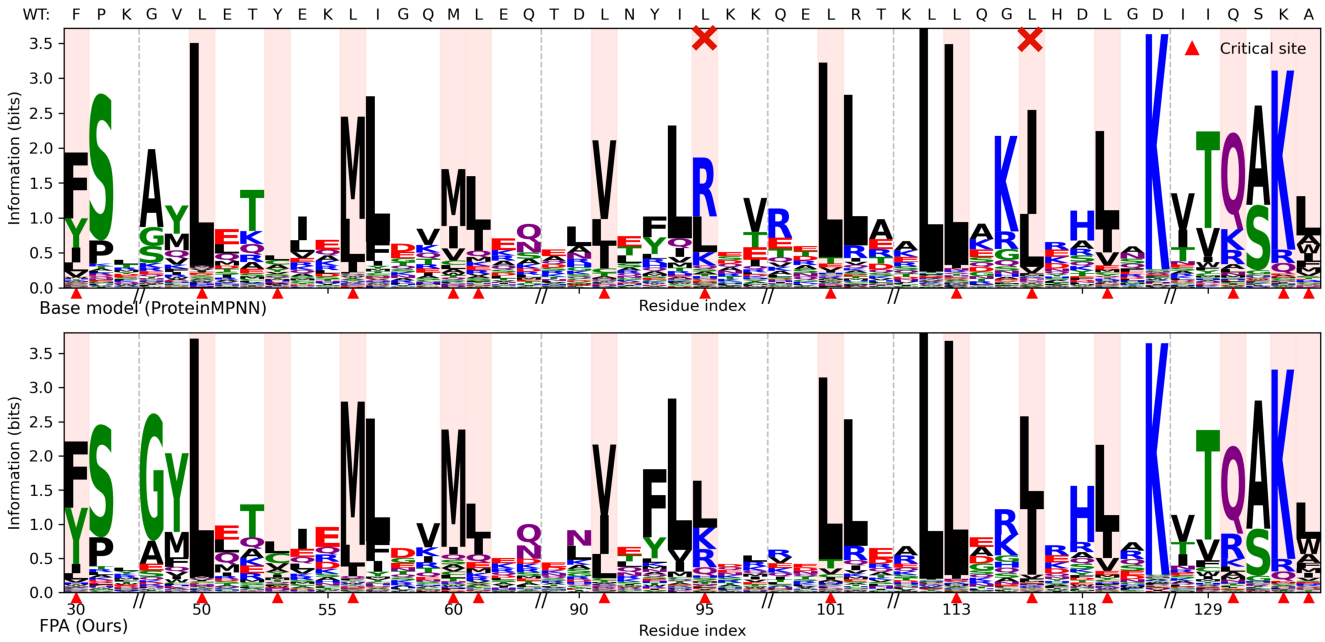

Figure 3: Sequence-logo visualization of residue-level predictive distributions for the wild-type (WT) sequence from the ProteinMPNN base model (top) and our FPA fine-tuned model (bottom). The WT residues are displayed above each position, while functionally critical sites are highlighted by shaded background columns and red triangle markers. Residues mispredicted by ProteinMPNN but correctly recovered by our model are marked with "×".

accumulate, suggesting strengthened discriminative power under higher mutational complexity. (2) The base models exhibit a similar but much weaker trend, with substantial overlap between negative and the WT sequence in early stages (Stages 2–4), and even up to Stage 7 for the ProteinMPNN base model. This behavior reflects their emphasis on global sequence recovery rather than explicitly penalizing mutations at function-critical residues, which limits their ability to guarantee function-preserving designs. (3) For both base and FPA fine-tuned models, ESM-IF consistently outperforms ProteinMPNN, which is consistent with the results reported in Table 1. This advantage is likely attributable to the larger training corpus and higher model capacity of ESM-IF. (4) Overall, these results provide wet-lab–grounded evidence that our function-aware preference alignment improves the identification of function-disrupting mutations and enhances the reliability of functional protein design.

5 Conclusion

We presented a function-aware preference alignment framework for protein inverse folding that improves functional robustness by guiding models to favor function-preserving residues, without requiring explicit functional optimization or additional wet-lab experiments. By constructing function-aware preference pairs entirely in silico, our approach enables scalable fine-tuning of existing inverse folding models while remaining compatible with widely used pipelines such as ProteinMPNN and ESM-IF. Importantly, wet-lab validation on enzyme datasets demonstrates that the fine-tuned models can explicitly distinguish functionally critical residues from non-critical positions. This behavior reflects a biologically meaningful shift in model focus from global sequence plausibility toward residue-level functional constraints, aligning protein sequence generation more closely with known principles of protein function. As such, our framework provides a practical step toward protein design models that are not only structure-aware, but also sensitive to the molecular determinants of biological activity.

6 Limitations and Ethical Considerations

Our preference supervision is constructed in silico using bioinformatics signals (e.g., SIFT-guided critical-site perturbations) and model-based plausibility constraints. These proxies provide only an approximation to biochemical function and may miss context-dependent effects (e.g., epistasis, conformational dynamics, or assay-specific conditions). Importantly, while our FPA does not require additional wet-lab measurements, we validate the approach using benchmark experiments on an in-house enzyme dataset with wet-lab readouts; nevertheless, broader generalization across protein families, assays, and operating conditions remains to be established.

This work uses publicly available protein sequence–structure resources and does not involve human subjects or personal data. Protein design methods can raise dual-use concerns; our approach does not introduce new capabilities beyond existing inverse folding backbones, but rather improves functional consistency under structural constraints. Practical deployment should follow standard biosafety and responsible research practices, including appropriate experimental screening, documentation, and biosecurity oversight.

7 GenAI Disclosure

Generative AI tools were used only for language editing, code prototyping, and formatting assistance. They were not used to generate scientific hypotheses, design experiments, analyze results, or draw conclusions. All core intellectual contributions are solely attributable to the authors.

Acknowledgments

This paper was supported by the Singapore Manufacturing Trade and Connectivity (MTC) Individual Research Grant (IRG) (Grant No. M24N7c0091).

References

[1] Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Yucen Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. 2025. Bayesian Optimization of Antibodies Informed by a Generative Model of Evolving Sequences. In *The Thirteenth International Conference on Learning Representations*.

[2] Andreas Bjerregaard, Peter Mørch Groth, Søren Hauberg, Anders Krogh, and Wouter Boomsma. 2025. Foundation models of protein sequences: A brief overview. *Current Opinion in Structural Biology* 91 (2025), 103004.

[3] Frimpong Boadu, Yanli Wang, and Jianlin Cheng. 2025. A unified multimodal model for generalizable zero-shot and supervised protein function prediction. *bioRxiv* (2025), 2025–05.

[4] Xixian Chen, Rehka T, Jérémy Esque, Congqiang Zhang, Sudha Shukal, Chin Chin Lim, Leonard Ong, Derek Smith, and Isabelle André. 2022. Total enzymatic synthesis of cis-α-irone from a simple carbon source. *Nature Communications* 13, 1 (2022), 7421.

[5] 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. 2022. Robust deep learning–based protein sequence design using ProteinMPNN. *Science* 378, 6615 (2022), 49–56.

[6] Alexander Derry, Alp Tartici, and Russ B Altman. 2025. Protein functional site annotation using local structure embeddings. *Proceedings of the National Academy of Sciences* 122, 34 (2025), e2513219122.

[7] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. 2022. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications* 13 (2022). https://api.semanticscholar.org/CorpusID:247439606

[8] Zhangyang Gao, Cheng Tan, and Stan Z. Li. 2023. PiFold: Toward effective and efficient protein inverse folding. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=oMsN9TYwJ0j

[9] Hans-Christof Gasser, Diego A Oyarzún, Javier Antonio Alfaro, and Ajitha Rajan. 2025. Tuning ProteinMPNN to reduce protein visibility via MHC Class I through direct preference optimization. *Protein Engineering, Design and Selection* 38 (2025), gzaf003.

[10] Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hötzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. 2023. Protein Design with Guided Discrete Diffusion. In *Advances in Neural Information Processing Systems*.

[11] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. 2022. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*. PMLR, 8946–8970.

[12] Bozhen Hu, Cheng Tan, Yongjie Xu, Zhangyang Gao, Jun Xia, Lirong Wu, and Stan Z Li. 2024. Protgo: Function-guided protein modeling for unified representation learning. *Advances in Neural Information Processing Systems* 37 (2024), 88581–88604.

[13] Mingyu Huang, Shasha Zhou, and Ke Li. 2025. Augmenting Biological Fitness Prediction Benchmarks with Landscapes Features from GraphFLA. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[14] Vijay Jayaraman, Saacnicteh Toledo-Patiño, Lianet Noda-García, and Paola Laurino. 2022. Mechanisms of protein evolution. *Protein Science* 31, 7 (2022), e4362.

[15] Pauline C Ng and Steven Henikoff. 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research* 31, 13 (2003), 3812–3814.

[16] Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. 2024. Machine learning for functional protein design. *Nature biotechnology* 42, 2 (2024), 216–228.

[17] Ryan Park, Darren J Hsu, C Brian Roland, Maria Korshunova, Chen Tessler, Shie Mannor, Olivia Viessmann, and Bruno Trentini. 2024. Improving inverse folding for peptide design with diversity-regularized direct preference optimization. *arXiv preprint arXiv:2410.19471* (2024).

[18] Jiezhong Qiu, Junde Xu, Jie Hu, Hanqun Cao, Liya Hou, Zijun Gao, Xinyi Zhou, Anni Li, Xiujuan Li, Bin Cui, et al. 2024. Instructplm: Aligning protein language models to follow protein structure instructions. *bioRxiv* (2024), 2024–04.

[19] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems* 36 (2023), 53728–53741.

[20] T Rehka, Xin Li, Jing Sen Ong, Jérémy Esque, Congqiang Zhang, Qingsong Lin, Isabelle André, and Xixian Chen. 2023. Mutagenesis of dimer interfacial residues improves the activity and specificity of methyltransferase for cis-α-irone biosynthesis. *Journal of Agricultural and Food Chemistry* 71, 22 (2023), 8497–8507.

[21] Yi Ren and Danica J. Sutherland. 2025. Learning Dynamics of LLM Finetuning. In *The Thirteenth International Conference on Learning Representations*.

[22] Boris Reva, Yevgeniy Antipin, and Chris Sander. 2011. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic acids research* 39, 17 (2011), e118–e118.

[23] Henry D Smith, Nathaniel L Diamant, and Brian L Trippe. 2025. Calibrating Generative Models. *arXiv preprint arXiv:2510.10020* (2025).

[24] Romain A Studer, Benoit H Dessailly, and Christine A Orengo. 2013. Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. *Biochemical journal* 449, 3 (2013), 581–594.

[25] Boris Thibert, Dale E Bredesen, and Gabriel del Rio. 2005. Improved prediction of critical residues for protein function based on network and phylogenetic analyses. *BMC bioinformatics* 6, 1 (2005), 213.

[26] Annabel E Todd, Christine A Orengo, and Janet M Thornton. 2002. Sequence and structural differences between enzyme and nonenzyme homologs. *Structure* 10, 10 (2002), 1435–1451.

[27] Serbulent Unsal, Heval Atas, Muammer Albayrak, Kemal Turhan, Aybar C Acar, and Tunca Doğan. 2022. Learning functional properties of proteins with language models. *Nature Machine Intelligence* 4, 3 (2022), 227–245.

[28] Robert Vaser, Swarnaseetha Adusumalli, Sim Ngak Leng, Mile Sikic, and Pauline C Ng. 2016. SIFT missense predictions for genomes. *Nature protocols* 11, 1 (2016), 1–9.

[29] Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. 2025. ProteinZero: Self-Improving Protein Generation via Online Reinforcement Learning. *arXiv preprint arXiv:2506.07459* (2025).

[30] Talal Widatalla, Rafael Rafailov, and Brian Hie. 2024. Aligning protein generative models with experimental fitness via Direct Preference Optimization. *bioRxiv* (2024). doi:10.1101/2024.05.20.595026

[31] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2025. Self-Play Preference Optimization for Language Model Alignment. In *The Thirteenth International Conference on Learning Representations*.

[32] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2025. Self-Play Preference Optimization for Language Model Alignment. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=a3PmRgAB5T

[33] Yijia Xiao, Wanjia Zhao, Junkai Zhang, Yiqiao Jin, Han Zhang, Zhicheng Ren, Renliang Sun, Haixin Wang, Guancheng Wan, Pan Lu, et al. 2025. Protein large language models: A comprehensive survey. *arXiv preprint arXiv:2502.17504* (2025).

[34] Junde Xu, Zijun Gao, Xinyi Zhou, hujie, Xingyi Cheng, Le Song, Guangyong Chen, Pheng-Ann Heng, and Jiezhong Qiu. 2025. Protein Inverse Folding From Structure Feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[35] Fanglei Xue, Andrew Kubaney, Zhichun Guo, Joseph K Min, Ge Liu, Yi Yang, and David Baker. 2025. Improving Protein Sequence Design through Designability Preference Optimization. *arXiv preprint arXiv:2506.00297* (2025).

[36] Ziyi Zhou, Liang Zhang, Yuanxi Yu, Banghao Wu, Mingchen Li, Liang Hong, and Pan Tan. 2024. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications* 15, 1 (2024), 5566.

A Data Preparation and Function-aware Pairwise Preference Construction

In this section, we describe the datasets used in this work, including their sources, sizes, and train–validation–test splits. All sequences were filtered to remove incomplete or low-quality samples prior to training.

A.1 Dataset Preparation

We directly build our dataset upon the ProteinMPNN dataset (PDB release 2021-08-02) [5] by constructing function-aware preference

pairs from its curated protein sequences, which includes PDB chain identifiers, amino acid sequences, resolution, and cluster assignments. To ensure high-quality data, we retained only structures with resolution ≤ 3.5 Å and sequence lengths below 10,000 residues. Dataset splits (training, validation, test) were determined using precomputed non-overlapping structural clusters.

For each backbone structure x, ProteinMPNN in inference mode produces a log-probability matrix of size $L \times 21$, where L is the sequence length and the 21 columns represent the 20 standard amino acids plus a token for unknown residues. These matrices were stored while preserving the amino acid order, facilitating uniform tabular representation for large-scale analyses such as sequence scoring, likelihood computation, and comparative evaluation across splits.

ESM-IF inference is modified during dataset preparation to produce a log-probability matrix to ensure comparability of magnitude of log-probability with ProteinMPNN. The ESM-IF base model uses a vocabulary of size 35, producing a logit matrix of size $L \times 35$. We restricted the output distribution to the 21 tokens of ProteinMPNN by removing logits that correspond to non-standard amino acid or auxiliary tokens prior to softmax operation. The remaining logits are renormalized during softmax and produce $L \times 21$ log probability matrix that is directly comparable across models.

A.2 SIFT-Based Mutational Analysis

To assess residue-level mutational sensitivity, we applied SIFT (Sorting Intolerant From Tolerant) to all ground-truth protein sequences. For each protein, we enumerated all possible single–amino-acid substitutions by mutating each residue to the 19 alternative standard amino acids, excluding identity substitutions. Residue positions containing ambiguous amino acids (e.g., "X") were excluded from mutation analysis to avoid introducing uncertainty. All variants were generated using a preprocessing script compatible with SIFT input requirements.

SIFT predicts the functional impact of each amino acid substitution based on evolutionary conservation and amino acid similarity, producing a score in the range $[0, 1]$, where scores ≤ 0.05 indicate substitutions likely to be deleterious. Additional outputs include alignment-derived statistics such as effective sequence diversity and conservation scores. Variants labeled as `NOT SCORED` by SIFT were removed during post-processing. Following standard practice, substitutions initially classified as `DELETERIOUS` but associated with low conservation (median conservation score > 3.5) were reclassified as `TOLERATED`, reflecting reduced selective constraint at the corresponding positions.

For each residue position, we aggregated the sets of tolerated and non-tolerated amino acids, always including the wild-type residue in the tolerated set. This procedure yields a residue-level mutational tolerance profile for each protein, which serves as a biologically grounded signal for identifying functionally critical sites across the dataset.

A.3 SIFT-Guided Function-aware Preference Construction

We refer to the pairwise preferences constructed as hypothetical function-aware preferences. Here, "hypothetical" indicates that these preference pairs are synthetically generated through function-aware perturbations, rather than derived from direct experimental functional annotations.

To generate negative sequences that are biologically constrained yet statistically plausible, we integrate residue-level mutational tolerance information from **SIFT** with sequence likelihoods provided by **ProteinMPNN** and **ESM-IF**. For each ground-truth (GT) sequence, we consider only residue positions at which SIFT predicts a unique tolerated amino acid that matches the GT residue, while multiple non-tolerated substitutions exist. Positions containing ambiguous amino acids (e.g., `X`) or lacking reliable SIFT annotations are excluded from mutation and retain the GT residue, in order to avoid introducing uncertainty or disrupting backbone compatibility.

Formally, let T_i and N_i denote the SIFT-predicted tolerated and non-tolerated amino acid sets at position i, respectively. We define the set of SIFT-identified critical sites as

$$\mathcal{F}(y) = \{i \in \{1, 2, \ldots, L\} \mid (T_i = \{y_i\}) \wedge (|N_i| > 10)\}. \tag{11}$$

Candidate substitutions are then drawn from the corresponding *Predict Not Tolerated* set N_i provided by SIFT, reflecting mutations that are likely to impair protein function.

Furthermore, for each candidate residue $a \in \mathcal{A}$ at position i, we compute per-residue log-likelihoods $\pi_i^{\mathcal{P}}(a \mid x)$ and $\pi_i^{\mathcal{E}}(a \mid x)$ under ProteinMPNN and ESM-IF, respectively. To ensure that candidate mutations remain compatible with the given backbone structure, we introduce a tolerable log-likelihood deviation parameter $\tau = -0.1$, which controls how much higher a candidate's likelihood must be relative to that of the ground-truth residue. A candidate residue $a \in \mathcal{A}$ at position i is accepted only if both models satisfy the following criteria with respect to the ground-truth residue y_i. Namely

$$C_h(i \mid x) = \left\{ a \in N_i \;\middle|\; \bigwedge_{* \in \{\mathcal{P}, \mathcal{E}\}} \left[\pi_i^*(a \mid x) \geq \pi_i^*(y_i \mid x) + \tau\right] \right\}. \tag{12}$$

We define the hard-filtered critical-site set as the subset of SIFT critical sites for which at least one hard candidate substitution exists:

$$\mathcal{F}_h(y \mid x) = \{i \in \mathcal{F}(y) \mid C_h(i \mid x) \cap N_i \neq \varnothing\}. \tag{13}$$

All valid candidate substitutions are then aggregated across positions, and up to k multi-site substitution combinations are sampled using a permutation-based strategy. Specifically, Let $\mathcal{F}_h(y \mid x)$ and $\{C_h(i \mid x)\}_{i \in \mathcal{F}_h(y|x)}$ be defined as above. We sample up to k distinct multi-site mutation sets $\mathcal{M} = \{\mathcal{M}_j\}_{j=1}^{k}$ such that

$$\mathcal{M}_j \subseteq \bigcup_{i \in \mathcal{F}_h(y|x)} \{(i, a) | a \in C_h(i \mid x) \cap N_i\}, \qquad j = 1, \ldots, k, \tag{14}$$

where each combination $\mathcal{M}_j$ induces a distinct negative sequence by applying all substitutions $(i, a) \in \mathcal{M}_j$ to y, allowing us to construct diverse function-disrupting variants while maintaining overall structural plausibility.

In addition to model-likelihood-based hard negatives, a small fraction of additional negative sequences were generated by sampling from SIFT non-tolerated amino acids set at critical sites without other constraint.

Overall, this procedure combines biologically motivated mutational constraints derived from SIFT with model-based structural

likelihoods from ProteinMPNN and ESM-IF. The τ-relaxation ensures that negative sequences remain statistically consistent with strong inverse folding models, while permutation-based sampling yields diverse yet minimally perturbed variants suitable for downstream preference-based fine-tuning and evaluation.

A.4 Preventing Data Leakage

To prevent data leakage from ESM-IF pre-training dataset, we explicitly filtered out overlapping splits across training, validation and test sets. For instance, any samples that appear in validation or test splits of the ESM-IF1 pre-training dataset were removed from our training set. Similarly, samples included in other splits in pre-training were excluded from the test set and validation set with the same filtering strategies. The resulting dataset sizes after filtering are in Table 4.

This issue does not arise for ProteinMPNN, as its dataset explicitly provides non-overlapping training, validation, and test splits. These precautions ensure that downstream evaluation reflects genuine generalization to unseen protein sequences.

B Detailed derivation from the SPPO loss to the FPA loss

We build our fine-tuning objective on Self-Play Preference Optimization (SPPO) [32], which frames preference learning as an alignment problem with respect to the model's own distribution. Let $\pi_\theta(\cdot \mid x)$ denote the current inverse folding model and $\pi_t(\cdot \mid x)$ the current policy (in practice, we instantiate π_t by a fixed reference model, i.e., $\pi_t = \pi_{\text{ref}}$). SPPO assumes access to a pairwise preference oracle $P(y \succ y' \mid x)$ and lifts it to a self-play preference score against π_t by averaging over candidates sampled from π_t:

$$P(y \succ \pi_t \mid x) = \mathbb{E}_{y' \sim \pi_t(\cdot|x)}\big[P(y \succ y' \mid x)\big]. \tag{15}$$

SPPO then fits the log-likelihood ratio of a candidate sequence y to this preference signal via a quadratic regression objective:

$$\ell_{\text{SPPO}}(x, y \mid \pi_t) = \left[\beta \log \frac{\pi_\theta(y \mid x)}{\pi_t(y \mid x)} - \big(P(y \succ \pi_t \mid x) - \frac{1}{2}\big)\right]^2, \tag{16}$$

where $\beta > 0$ controls the strength of alignment. Unlike DPO, which optimizes only the positive–negative gap, SPPO applies separate calibration targets to individual samples. This decoupling is particularly desirable in protein sequence design, where preference supervision is typically sparse and biologically heterogeneous (e.g., functional disruption may arise from stability, catalysis, or binding defects).

B.1 Hard-label single-pair preferences

In our setting, each backbone x is associated with a single function-aware preference pair (y^+, y^-), where y^+ is the wild-type (ground-truth) sequence assumed to be functionally competent, and y^- is a synthetically perturbed variant (e.g., deleterious substitutions at critical sites) hypothesized to be function-disrupting. This induces a deterministic preference label:

$$P(y^+ \succ y^- \mid x) = 1, \quad P(y^- \succ y^+ \mid x) = 0. \tag{17}$$

Under this hard-label regime, the self-play preference score in Eq. (15) simplifies as follows. Since π_t is anchored on the same backbone x and the preference label is deterministic for the constructed pair, we obtain

$$P(y^+ \succ \pi_t \mid x) = 1, \quad P(y^- \succ \pi_t \mid x) = 0. \tag{18}$$

B.2 Deriving the FPA objective

Crucially, SPPO is defined per candidate sequence y (not per pair). Therefore, when we have a preference pair (y^+, y^-), we apply the SPPO loss to each member of the pair and aggregate them. Plugging Eq. (18) into Eq. (16), we obtain two quadratic terms:

$$\begin{aligned}
\ell_{\text{SPPO}}(x, y^+ \mid \pi_t) &= \left[\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_t(y^+ \mid x)} - \Big(1 - \frac{1}{2}\Big)\right]^2 && \text{(19a)}\\
&= \left[\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_t(y^+ \mid x)} - \frac{1}{2}\right]^2, \\
\ell_{\text{SPPO}}(x, y^- \mid \pi_t) &= \left[\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_t(y^- \mid x)} - \Big(0 - \frac{1}{2}\Big)\right]^2 && \text{(19b)}\\
&= \left[\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_t(y^- \mid x)} + \frac{1}{2}\right]^2.
\end{aligned}$$

Summing the two terms yields a symmetric pairwise alignment objective:

$$\ell_{\text{Seq}}(x, y^+, y^-) = \ell_{\text{SPPO}}(x, y^+ \mid \pi_t) + \ell_{\text{SPPO}}(x, y^- \mid \pi_t). \tag{20}$$

Finally, instantiating π_t as the frozen reference inverse folding model π_{ref} and adopting the relative log-likelihood ratios a, b defined in Eq. (6), we obtain the FPA loss:

$$\ell_{\text{Seq}}(x, y^+, y^-) = \Big(a - \frac{1}{2}\Big)^2 + \Big(b + \frac{1}{2}\Big)^2. \tag{21}$$

The two quadratic terms arise in Eq. (21) because SPPO is applied independently to the positive (wild-type, function-preserving) sequence y^+ and the negative (function-disrupting) variant y^-. This yields separate and directionally consistent updates: the first term explicitly encourages increasing the likelihood of y^+ relative to the reference model, while the second term suppresses y^- relative to the same reference. Compared with gap-only objectives (e.g., DPO), this decoupled form is particularly suitable for sparse, hypothesis-driven preference pairs in protein inverse folding, where the preferred sequence is ground-truth and should not be inadvertently penalized during fine-tuning.

C Alternative Scoring Functions for Computational Functional Assessment

The scoring function ϕ can be derived from one of the following sources:

- **Evolutionary Conservation:** Let $p(y_i = a)$ denote the empirical frequency of amino acid a at position i in the multiple sequence alignment associated with y. The evolutionary conservation score is defined as

$$\phi_{\text{evol}}(i \mid y) = \log_2 20 - \sum_{a \in \mathcal{A}} p(y_i = a) \log_2 \frac{1}{p(y_i = a)}.$$

- **Thermodynamic Stability:** The maximum change in Gibbs free energy upon mutation,

$$\phi_{\text{stab}}(i \mid y, x) = \max_{a \in \mathcal{A}} \big|\Delta G(y \mid x) - \Delta G\big(y^{i \to a} \mid x\big)\big|,$$

Table 4: Dataset size. `Total Unique FASTA` corresponds to the number of unique FASTA sequences in the ProteinMPNN CATH 4.2 40% non-redundant set. `No Negative Seq` denotes sequences that did not have any hard negative substitution during Section 3.3 and were removed from the dataset. `ESM-IF Dataset Overlap` indicates samples excluded due to overlap with other splits of ESM-IF pretrained dataset (Section A.4).

Split	Total Unique FASTA	No Negative Seq	ESM-IF Dataset Overlap	Final Dataset
Train	113,095	68,690	1,853	42,552 (37.6%)
Val	4,727	1,445	687	2,595 (54.9%)
Test	4,518	1,539	665	2,314 (51.2%)

where $y^{i\to a}$ denotes the sequence obtained by mutating residue y_i to amino acid a at position i, and $\Delta G(\cdot \mid x)$ denotes the folding free energy under the fixed backbone structure x.

- **Computational Probabilistic Influence:** In a masked protein language model (PLM) parameterized by θ, the negative log-likelihood of the observed residue,

$$\phi_{\mathrm{PLM}}(i \mid y, \pi_\theta) = -\log \pi_\theta(y_i \mid y_{\setminus i}).$$

D Additional Results

ProtGPT for sequence quality. We follow [10] to apply ProtGPT [7] to evaluate the naturalness of the sampled sequences from the protein inverse folding models. ProtGPT outputs the average log likelihood over residues for a given sequence. As shown in Table 5, the fine-tuned model exhibits a likelihood for its generated sequences that is similar to that of the pre-trained model, demonstrating that our FPA does not compromise the naturalness of the produced sequences.

Table 5: Average likelihood scores (mean ± standard deviation) for different methods.

Method	Single	Multi
ProteinMPNN	−8.77 ± 0.99	−8.82 ± 0.82
+FPA (ours)	−8.79 ± 0.96	−8.86 ± 0.80
ESM-IF	−8.45 ± 1.05	−8.41 ± 1.18
+FPA (ours)	−8.32 ± 1.15	−8.32 ± 1.04

Entropy for critical and non-critical sites. We compute the entropy of the predicted amino-acid distributions at critical and non-critical sites for each protein, and report the mean and standard deviation (std) in Table 6. The results show that FPA consistently reduces entropy at critical sites for both ProteinMPNN and ESM-IF, indicating more confident and deterministic predictions at functionally important positions. In contrast, the entropy at non-critical sites remains largely unchanged, suggesting that FPA selectively sharpens predictions at critical residues without reducing diversity elsewhere.

Different base models in ProteinMPNN. ProteinMPNN [5] has different setups for the backbone noise and released different pretrained base models (v_48_020 and v_48_002). We apply our FPA to these models, and both show improvements as shown in Table 8.

Table 6: Entropy comparison between critical and non-critical sites for different methods.

Method	Critical	Non-critical
ProteinMPNN	1.925 ± 0.350	2.481 ± 0.220
+FPA (ours)	1.795 ± 0.412	2.443 ± 0.252
ESM-IF	0.832 ± 0.445	1.611 ± 0.304
+FPA (ours)	0.796 ± 0.473	1.613 ± 0.336

E Model Architecture Details and Training Procedure

E.1 Base Model

We use ProteinMPNN [5] and ESM-IF [11] as base inverse folding models, adopting their official implementations and pretrained weights without any architectural modifications[3][4]. As our method focuses solely on preference-based fine-tuning, all reported performance gains arise from the proposed function-aware preference alignment objective rather than changes to model architecture or capacity.

E.2 Fine-Tuned Variants

We directly adopt the official implementations and pretrained weights of ProteinMPNN and ESM-IF, initializing all fine-tuned models from the base checkpoints. Modified training objectives are applied without architectural changes.

E.3 Optimization

ESM-IF models were trained with the AdamW optimizer with learning rate 1×10^{-7} following the ProteinDPO paper [30]. ProteinMPNN models were trained with AdamW optimizer with learning rate 5×10^{-6} .Training was performed for a fixed number of epochs with early stopping based on validation performance. We collect the non-default hyperparameters in Table 7.

E.4 Regularization

Dropout was applied during training to mitigate overfitting. Weight decay was used where indicated. No regularization was applied during evaluation.

[3]https://github.com/dauparas/ProteinMPNN
[4]https://github.com/facebookresearch/esm/tree/main/examples/inverse_folding

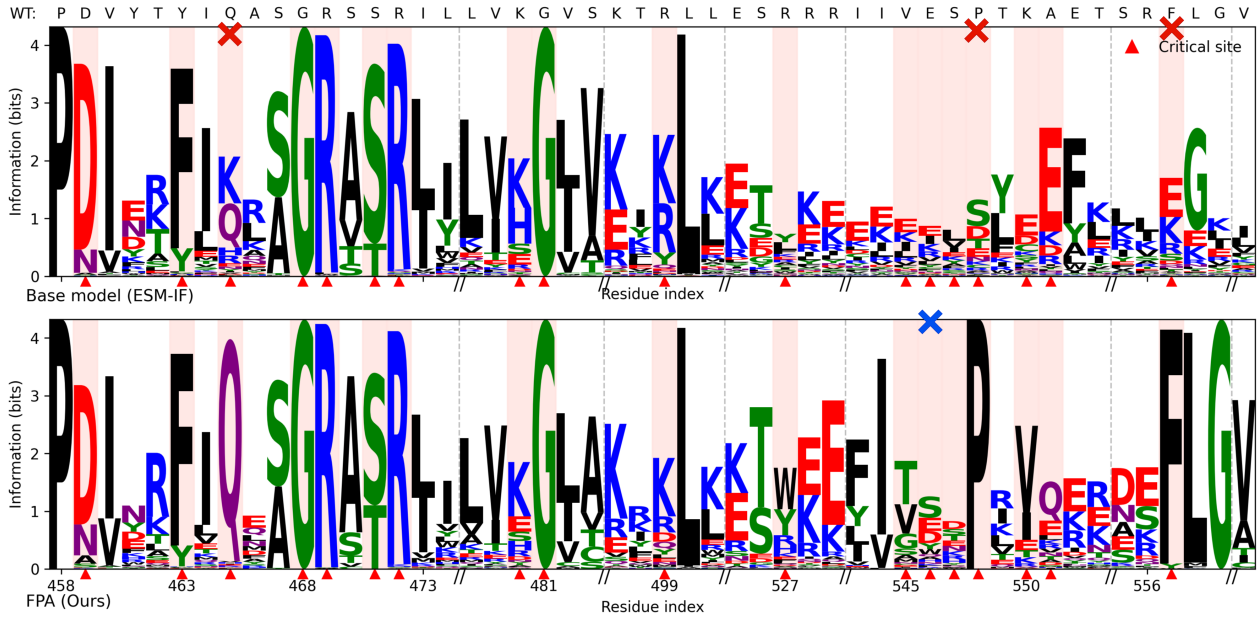

Figure 4: Sequence-logo visualization of residue-level predictive distributions for the wild-type (WT) sequence obtained from the ESM-IF base model (top) and our FPA fine-tuned model (bottom). The WT residues are shown at the top, and functionally critical sites are highlighted by shaded background columns and red triangle markers. Residues mispredicted by ProteinMPNN but correctly recovered by our model are marked with "×". Residues that is correct for the base model but mispredicted by our model are marked with "×".

Table 7: Training hyperparameters for our FPA.

Hyperparameter	ProteinMPNN	ESM-IF
FPA scaling factor β	0.2	0.1
FPA recovery loss λ	0.2	1
Optimizer	AdamW	AdamW
Optimizer params	$\beta_1 = 0.9, \beta_2 = 0.98,$ $\epsilon = 10^{-9}$	$\beta_1 = 0.9, \beta_2 = 0.98,$ $\epsilon = 10^{-8}$
Weight decay	0.01	0.1
Learning rate	5×10^{-6}	1×10^{-7}
Epochs	20	40
Batch size	10,000 tokens	1

E.5 Hardware and Runtime

Experiments were conducted on GPUs with CUDA acceleration. ESM-IF models were trained on RTX Pro 6000 with 96GB memory. ProteinMPNN models were trained on RTX 6000 with 48GB memory.

E.6 Software and Libraries

All experiments were implemented in Python using PyTorch. Library versions are listed for reproducibility (Table 9).

F Evaluation Protocol and Metrics

We evaluate function-aware inverse folding using a combination of sequence-level and residue-level metrics defined over function-aware preference pairs $\mathcal{D}_{\text{test}}$. Sequence-level metrics assess global likelihood and generation fidelity under a given backbone structure, while residue-level metrics focus specifically on functionally perturbed (critical) positions where positive and negative sequences differ.

Evaluation Protocol. For each test backbone structure x with a corresponding preference pair (y^+, y^-), all sequence-level metrics are computed using the full sequences without masking. For residue-level preference evaluation, we restrict analysis to the set of perturbed (critical) positions $C = \{i \mid y_i^+ \neq y_i^-\}$, where function-aware mutations are introduced. At these positions, log-probabilities are extracted directly from the model output and compared between the ground-truth and negative residues.

For sequence generation metrics, we adopt Best-of-K sampling: multiple sequences are sampled per backbone, and the sample with the highest recovery score is used for evaluation.

We use the following metrics for in silico evaluation in our experiments. For consistency, we compute expectations uniformly over D_{test} for all metrics, even though some depend only on the ground truth y^+. Furthermore, to simplify notation, we denote the sequence length uniformly as L, despite actual variation across different sequences.

Table 8: Evaluation of FPA for fine-tuning ProteinMPNN with different base models (backbone noise as 0.2 v_48_020 and backbone noise as 0.02 v_48_002) on single-chain (Single) and multi-chain (Multi) test sets.

	Perplexity ↓		Sequence Recovery (%) ↑		Critical Site Recovery (%) ↑	
	Single	Multi	Single	Multi	Single	Multi
Base (0.02)	3.78	3.39	56.97	57.70	25.35	26.46
+FPA (ours)	**3.74**	**3.39**	**57.13**	**57.65**	**28.17**	**29.56**
Base (0.2)	4.55	4.03	50.40	51.04	16.93	16.73
+FPA (ours)	**4.38**	**3.92**	**51.39**	**51.87**	**20.08**	**21.45**

Table 9: Software libraries and versions used in ESM-IF and ProteinMPNN experiments.

Software / Library	ProteinMPNN	ESM-IF
Python	Python 3.12.12	Python 3.10
PyTorch	2.5.1	2.9.0
Model implementation	Original ProteinMPNN	fair-esm (commit 2b36991)
GPU acceleration	CUDA 12.1	CUDA 12.8
GPU primitives	cuDNN 9.10	cuDNN 9.10
NumPy	2.4.1	1.26.4
Biotite	–	0.40.0
torch-geometric	–	2.7.0
torch-scatter	–	2.1.2
torch-sparse	–	0.6.18
torch-cluster	–	1.6.3
torch-spline-conv	–	1.2.2

- **Perplexity (PPL)** measures the model's uncertainty in predicting a protein sequence y given its backbone structure x. It is defined as the exponentiated average negative log-likelihood per residue:

$$\mathrm{PPL}(y \mid x) = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log \pi_\theta(y_i^+ \mid x)\right)\right],$$

  where $\pi_\theta(y_i^+ \mid x)$ is the model's probability for residue y_i^+ at position i. .
- **Average Log-Likelihood (LL)** reports the average per-residue log-likelihood assigned by the model to the positive and negative sequences under the same backbone:

$$\mathrm{LL}^+ = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\frac{1}{L}\sum_{i=1}^{L}\log \pi_\theta(y_i^+ \mid x)\right],$$

$$\mathrm{LL}^- = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\frac{1}{L}\sum_{i=1}^{L}\log \pi_\theta(y_i^- \mid x)\right].$$

- **Preference Accuracy (ACC)** measures the fraction of preference pairs in which the model assigns higher likelihood to the positive sequence than to the negative variant under the same backbone:

$$\mathrm{ACC} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\mathbb{I}\left[\log \pi_\theta(y^+ \mid x) > \log \pi_\theta(y^- \mid x)\right]\right].$$

- **Sequence Recovery (Rec)** measures the average fraction of residues in generated sequences that match the ground-truth sequence:

$$\mathrm{Rec} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\max_{k=1,\dots,K}\frac{1}{L}\sum_{i=1}^{L}\mathbb{I}[\hat{y}_i^{(k)} = y_i^+]\right].$$

- **Delta Log-Likelihood (ΔLL)** quantifies the average log-likelihood gap between positive and negative variants at perturbed positions:

$$\Delta\mathrm{LL} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\frac{1}{|C|}\sum_{i\in C}\left(\log \pi_\theta(y_i^+ \mid x) - \log \pi_\theta(y_i^- \mid x)\right)\right].$$

- **Residue Preference Accuracy (Res-ACC)** measures the fraction of perturbed positions at which the model assigns higher likelihood to the ground-truth residue than to the corresponding negative mutation:

$$\mathrm{Res}_{\mathrm{ACC}} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\frac{1}{|C|}\sum_{i\in C}\mathbb{I}\left[\log \pi_\theta(y_i^+ \mid x) > \log \pi_\theta(y_i^- \mid x)\right]\right].$$

- **Critical Sequence Recovery (CSR)** measures recovery restricted to perturbed (critical) positions:

$$\mathrm{CSR} = \mathbb{E}_{(y^+,y^-,x)\sim\mathcal{D}_{\text{test}}}\left[\max_{k=1,\dots,K}\frac{1}{|C|}\sum_{i\in C}\mathbb{I}[\hat{y}_i^{(k)} = y_i^+]\right].$$

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009