FROM STRUCTURE TO FUNCTION: PREFERENCE ALIGNMENT FOR FUNCTION-AWARE PROTEIN INVERSE FOLDING

Tamatgar Nilufer[1*], **Soobin Park**[2*], **Yinghua Yao**[13†], **Xixian Chen**[4], **Yuangang Pan**[13]
[1]Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
[2]Applied Information Engineering, Yonsei University, South Korea
[3]Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
[4]Singapore Innovation of Food and Biotechnology, Agency for Science, Technology and Research, Singapore
{nilufer2703,clapkong}@gmail.com
{yao_yinghua,Xixian_Chen,pan_yuangang}@a-star.edu.sg

ABSTRACT

Protein inverse folding (IF) 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 (FPA) framework that avoids explicit function optimization by fine-tuning IF models to prefer function-preserving sequences over function-disrupting alternatives. 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. Our FPA framework guides protein sequence design models toward generating sequences that better preserve functional integrity, while remaining model-agnostic and compatible with existing inverse folding pipelines such as ProteinMPNN and ESM-IF. Extensive experiments on protein design benchmarks show that our fine-tuned models significantly outperform pretrained counterparts in preserving functional integrity during protein sequence design.

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 (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022) 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 (Xue et al., 2025). Protein function is often governed by a small subset of critical residues (Jayaraman et al., 2022), and perturbations at these sites[1] can abolish activity even when global structure is preserved (Derry et al., 2025).

A major challenge in function-aware protein design (Notin et al., 2024) is the lack of scalable supervision. Functional annotations are primarily obtained through wet-lab experiments, which are expensive and time-consuming (Zhou et al., 2024). 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 (Hu et al., 2024; Boadu et al., 2025), and no single unified function evaluator exists that generalizes across diverse protein families and biochemical tasks (Unsal et al., 2022; Xiao et al., 2025).

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

[*]Equal contribution. Work was done during Miss Soobin Park's internship at CFAR A*STAR.
[†]Corresponding author.
[1]In 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.

are likely to disrupt function. By discouraging such deleterious mutations, the model implicitly biases generation toward sequences that better preserve functional integrity, without requiring explicit functional labels (Amin et al., 2025; Huang et al., 2025). 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 (Reva et al., 2011; Studer et al., 2013) and model-consistent likelihood constraints (Bjerregaard et al., 2025; Smith et al., 2025), 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) (Ng & Henikoff, 2003; Vaser et al., 2016), 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 this signal, we generate function-aware preference pairs consisting of the original sequence and a negative variant produced by introducing deleterious substitutions at critical positions. 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 our 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 experiments on standard protein design benchmarks demonstrate that our fine-tuned models consistently outperform pretrained baselines and other fine-tuning methods in functional protein design.

2 Function-Aware Protein Inverse Folding

We first review protein inverse folding and formalize the problem of function-aware inverse folding. We then explain why standard preference-based fine-tuning methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) provide limited benefits in this setting. This motivates our Function-aware Preference Alignment (FPA) framework, designed to handle preference signals arising from diverse and mechanistically distinct functional perturbations.

2.1 Preliminaries

Protein inverse folding (IF) aims to generate an amino-acid sequence $y = (y_1, \ldots, y_L)$ conditioned on a backbone structure x, where $y_i \in \mathcal{A}$ and the amino acid set $\mathcal{A} = \{\mathrm{ACDEFGHIKLMNPQRSTVWY}\}$. Modern IF models (e.g., ProteinMPNN (Dauparas et al., 2022) and ESM IF (Hsu et al., 2022)) are trained to predict the sequence y based on the backbone structure x via an auto-regressive way:

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

Most protein IF models are trained to maximize overall sequence recovery, which primarily enforces global structural compatibility. However, protein function is typically governed by a small subset

of critical residues, while most positions are tolerant to mutation. Consequently, high sequence recovery does not guarantee functional competence, and IF–based design still requires extensive wet-lab screening to identify function-preserving variants.

2.2 Function-Aware Preferences via Critical Sites

To introduce functional awareness without relying on large-scale wet-lab annotations, we propose a computational strategy that constructs function-aware preference dataset consisting of positive and negative sequences generated via deleterious amino acid substitutions. This approach is grounded in the biological observation that protein function is often governed by a small set of critical residues, whose mutation can disproportionately impair structural stability, catalytic activity, or molecular recognition (Todd et al., 2002; Thibert et al., 2005).

Definition 1 (Critical Sites)**.** Given a protein sequence y of length L, the critical sites are a subset of residue positions $\mathcal{F}(y) \subseteq \{1, 2, \ldots, L\}$, whose substitutions are expected to substantially disrupt protein function, as quantified by a residue-level scoring function ϕ:

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

where τ is a predefined threshold.

We instantiate ϕ using SIFT, an evolutionary conservation–based predictor of mutational intolerance. Residues with a high fraction of deleterious substitutions are treated as functionally critical, without experimental labels.

2.3 Hypothetical Function-Aware Preference Pairs

Given a wild-type sequence y, we construct a function-disrupting variant y^- by introducing substitutions at functionally critical sites $\mathcal{F}(y)$, while keeping all other positions unchanged. This yields a hypothetical preference pair (y^+, y^-), where $y^+ = y$ is function-preserving and y^- is hypothesized to be function-disrupting.

To avoid trivial negatives that are incompatible with the backbone, we identify hard negatives using the base inverse folding model. Given a backbone x and a set of perturbed variants $\mathcal{Y}^-$, a sequence $y^- \in \mathcal{Y}^-$ is selected as the hard negative if

$$\log \pi(y^- \mid x) \geq \log \pi(y^+ \mid x) + \upsilon,$$

where $\pi(y \mid x)$ denotes the model likelihood and υ is a tolerance threshold.

Focusing on such hard negatives directs training toward biologically meaningful errors, enabling effective fine-tuning without explicit functional labels or additional wet-lab experiments.

2.4 Function-Aware Preference Alignment

Let $\pi_\theta(y \mid x)$ denote a trainable IF model and $\pi_{\text{ref}}(y \mid x)$ a frozen reference initialized from the pretrained backbone. Standard preference optimization methods such as DPO (Widatalla et al., 2024; Gasser et al., 2025; Xu et al., 2025) focus on increasing the likelihood gap between preferred and dispreferred sequences, which can inadvertently suppress the likelihood of native-like sequences under the assumation of a single, globally consistent preference criterion (Ren & Sutherland, 2025).

In contrast, protein function degradation can arise from diverse and mechanistically distinct perturbations (e.g., stability loss, catalytic disruption, impaired recognition), making such assumptions unsuitable. We therefore propose Function-aware Preference Alignment (FPA), which directly aligns the likelihoods of positive and negative sequences relative to the reference model. Please refer to Appendix for a detailed derivation process.

For each preference pair (y^+, y^-), FPA minimizes the symmetric alignment objective

$$\ell_{\text{Seq}}(x, y^+, y^-) = \left(a - \tfrac{1}{2}\right)^2 + \left(b + \tfrac{1}{2}\right)^2, \tag{3}$$

where $a = \beta \log \frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)}$ and $b = \beta \log \frac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}$. $\beta > 0$ controls the strength of alignment. Equation 3 explicitly encourages increasing the likelihood of the function-preserving sequence y^+ while suppressing that of the function-disrupting variant y^-, relative to a frozen reference model.

2.5 CRITICAL-SITE-AWARE ALIGNMENT

However, equation 3 does not fully account for the biological observation that functional constraints are typically localized to a small subset of residues, and most positions primarily support structural stability or sequence background. To reflect this locality, we restrict preference-driven updates to functionally critical sites, leaving functionally neutral regions largely unconstrained.

Concretely, we define a residue-level preference optimization objective by applying the alignment loss only at critical sites $\mathcal{F}(y^+)$:

$$\ell_{\mathrm{Res}}(y^+, y^-, x, \mathcal{F}) = \left(a_{\mathcal{F}} - \tfrac{1}{2}\right)^2 + \left(b_{\mathcal{F}} + \tfrac{1}{2}\right)^2, \tag{4}$$

where $a_{\mathcal{F}} = \beta \log \frac{\pi_\theta(y^+_{\mathcal{F}}|x)}{\pi_{\mathrm{ref}}(y^+_{\mathcal{F}}|x)}$ and $b_{\mathcal{F}} = \beta \log \frac{\pi_\theta(y^-_{\mathcal{F}}|x)}{\pi_{\mathrm{ref}}(y^-_{\mathcal{F}}|x)}$.

While equation 4 concentrates preference gradients on critical sites $\mathcal{F}(y^+)$, optimizing only these positions can still induce drift in non-critical regions through shared model parameters. To preserve the structural compatibility learned by the pretrained IF model, we introduce an auxiliary cross-entropy (CE) objective over the positive sequences only, i.e., $\ell_{\mathrm{CE}}(y^+, x) = -\sum_{i=1}^{L} \log \pi_\theta(y_i^+ \mid x)$. This term preserves the pretrained generative prior, enabling controlled adaptation at critical residues without compromising overall sequence plausibility or structural compatibility.

The final FPA objective combines localized preference alignment with global regularization:

$$\mathcal{L}_{\mathrm{FPA}} = \sum_{(y^+, y^-, x)} \Big[\ell_{\mathrm{Res}}(y^+, y^-, x, \mathcal{F}) + \lambda\, \ell_{\mathrm{CE}}(y^+, x)\Big], \tag{5}$$

where $\lambda \geq 0$ controls the regularization strength. Together, this formulation 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.

3 EXPERIMENT

In this work, we build upon ProteinMPNN (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022) as representative base models for studying function-aware inverse folding fine-tuning.

Dataset. All experiments are conducted on the CATH 4.2 40% non-redundant dataset (Dauparas et al., 2022), 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. 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. This ensures evaluation on sequences unseen during pretraining and precludes performance gains from reused validation or test data. Please refer to the Appendix for more details about the dataset construction.

Evaluation Metrics. We evaluate function-aware inverse folding using sequence recovery and critical-site recovery to assess overall sequence fidelity and targeted recovery at functionally perturbed residues. Complementary likelihood-based metrics measure whether the model consistently prefers function-preserving sequences over function-disrupting variants under the same backbone. Formal definitions are provided in the Appendix.

3.1 BENCHMARK EVALUATION

Table 1 reports the quantitative evaluation of FPA for fine-tuning ProteinMPNN and ESM-IF on test sets, using perplexity, sequence recovery, and critical site recovery. The results show that (1) FPA-based fine-tuning consistently achieves notable improvements over both base models across all three metrics; (2) FPA delivers the strongest and most balanced performance, substantially outperforming other fine-tuning strategies DPO and SPPO. While the overall sequence recovery rate slightly improves after fine-tuning, FPA consistently improves the recovery rate at functionally critical sites across both models. This highlights the effect of function-aware preference alignment:

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>

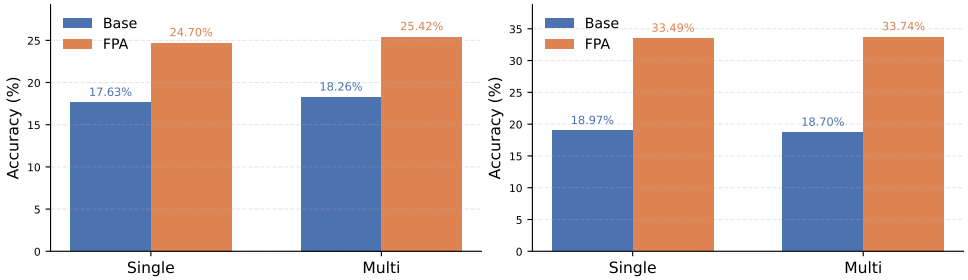

Figure 1: Pairwise preference accuracy of FPA-fine-tuned vs. base models on single-chain and multi-chain test sets for ProteinMPNN (**Left**) and ESM-IF (**Right**).

learning signals are concentrated on functionally relevant residues, rather than uniformly increasing similarity across all positions, thereby improving fidelity at critical sites without sacrificing global sequence recovery or structural plausibility. (3) 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.

3.2 Evaluation of Protein Function Discrimination via Preference Accuracy.

Preference accuracy is defined as the fraction of sequence pairs for which the protein design model assigns a higher likelihood to the positive sequence than to its corresponding negative variant, reflecting the model's ability to prioritize the generation of function-preserving sequences, thereby reducing the burden of wet-lab screening. Fig. 1 respectively presents the Preference accuracy of the base models ProteinMPNN and ESM-IF, as well as our FPA fine-tuned models, on the test set. It 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 negative sequence than to the positive one.

4 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, our experiment results demonstrate 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.

REFERENCES

Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Yucen Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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

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

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

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

Hans-Christof Gasser, Diego A Oyarzún, Javier Antonio Alfaro, and Ajitha Rajan. Tuning protein-mpnn to reduce protein visibility via mhc class i through direct preference optimization. *Protein Engineering, Design and Selection*, 38:gzaf003, 2025.

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

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

Mingyu Huang, Shasha Zhou, and Ke Li. 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*, 2025.

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

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

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

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

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

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

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

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

Henry D Smith, Nathaniel L Diamant, and Brian L Trippe. Calibrating generative models. *arXiv preprint arXiv:2510.10020*, 2025.

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

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

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

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

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

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

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, 2024. doi: 10.1101/2024.05.20.595026.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=a3PmRgAB5T.

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

Junde Xu, Zijun Gao, Xinyi Zhou, hujie, Xingyi Cheng, Le Song, Guangyong Chen, Pheng-Ann Heng, and Jiezhong Qiu. Protein inverse folding from structure feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Fanglei Xue, Andrew Kubaney, Zhichun Guo, Joseph K Min, Ge Liu, Yi Yang, and David Baker. Improving protein sequence design through designability preference optimization. *arXiv preprint arXiv:2506.00297*, 2025.

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

A Background

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 (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022) 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 (Gao et al., 2023) and InstructPLM (Qiu et al., 2024), 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) (Rafailov et al., 2023), have recently emerged as effective tools for aligning protein inverse folding models with target properties beyond sequence recovery (Gasser et al., 2025). Early work such as ProteinDPO (Widatalla et al., 2024) applied DPO to fine-tune ESM-IF (Hsu et al., 2022), encouraging preference for stabilizing over destabilizing variants. In parallel, similar paradigms have also been applied to ProteinMPNN (Dauparas et al., 2022): Xu et al. (2025) leveraged DPO with feedback from protein folding models, while Park et al. (2024) explored DPO-based optimization to enhance sequence diversity in peptide design. A more recent work (Xue et al., 2025) 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 (Wang et al., 2025) introduced an online reinforcement learning framework for inverse folding, enabling scalable multi-objective optimization with efficient structural feedback and diversity regularization.

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

B Data Preparation and 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.

B.1 Dataset Preparation

We directly build our dataset upon the ProteinMPNN dataset (PDB release 2021-08-02) (Dauparas et al., 2022) 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 in a manner consistent with ProteinMPNN. The ESM-IF base model uses a vocabulary of size 35, producing a logit matrix of size $L \times 35$. To ensure comparability of magnitude of log-probability with PrtoeinMPNN, we restricted the output distribution to the 21 tokens of ProteinMPNN. This is done 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. The 14 tokens are B, O, U, Z, ⟨unk⟩, ⟨pad⟩, ⟨null_1⟩, ⟨null_0⟩, ⟨mask⟩, ⟨eos⟩, ⟨cath⟩, ⟨af2⟩, ., $-$.

The tolerance threshold v used in generation of our preference pairs were 0.1, and the maximum negative sequences generated for one sequence was 20.

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.

B.2 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-IF 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 2.

Table 2: 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 2.3 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 B).

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%)

B.3 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.

C DETAILED DERIVATION FROM THE SPPO LOSS TO THE FPA LOSS

We build our fine-tuning objective on Self-Play Preference Optimization (SPPO) (Wu et al., 2025), 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:

$$P(y \succ \pi_t \mid x) = \mathbb{E}_{y' \sim \pi_t(\cdot|x)}\big[P(y \succ y' \mid x)\big]. \tag{6}$$

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

$$\ell_{\text{SPPO}}(x, y \mid \pi_t) = \Big[\beta \log \frac{\pi_\theta(y \mid x)}{\pi_t(y \mid x)} - \big(P(y \succ \pi_t \mid x) - \frac{1}{2}\big)\Big]^2, \tag{7}$$

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).

C.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 native (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:

$$P(y^+ \succ y^- \mid x) = 1, \quad P(y^- \succ y^+ \mid x) = 0. \tag{8}$$

Under this hard-label regime, the self-play preference score in equation 6 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

$$P(y^+ \succ \pi_t \mid x) = 1, \quad P(y^- \succ \pi_t \mid x) = 0. \tag{9}$$

C.2 DERIVING THE FPA OBJECTIVE

Crucially, SPPO is defined per candidate sequence y. Therefore, when we have a preference pair (y^+, y^-), we apply the SPPO loss to each member of the pair and aggregate them. Plugging equation 9 into equation 7, we obtain two quadratic terms:

$$\ell_{\mathrm{SPPO}}(x, y^+ \mid \pi_t) = \Big[\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_t(y^+ \mid x)} - \Big(1 - \frac{1}{2}\Big)\Big]^2 = \Big[\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_t(y^+ \mid x)} - \frac{1}{2}\Big]^2, \tag{10a}$$

$$\ell_{\mathrm{SPPO}}(x, y^- \mid \pi_t) = \Big[\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_t(y^- \mid x)} - \Big(0 - \frac{1}{2}\Big)\Big]^2 = \Big[\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_t(y^- \mid x)} + \frac{1}{2}\Big]^2. \tag{10b}$$

Summing the two terms yields a symmetric pairwise alignment objective:

$$\ell_{\mathrm{Seq}}(x, y^+, y^-) = \ell_{\mathrm{SPPO}}(x, y^+ \mid \pi_t) + \ell_{\mathrm{SPPO}}(x, y^- \mid \pi_t). \tag{11}$$

Finally, instantiating π_t as the frozen reference inverse folding model π_{ref} and adopting the relative log-likelihood ratios a, b defined in Section 2.4, we obtain the FPA loss:

$$\ell_{\mathrm{Seq}}(x, y^+, y^-) = \Big(a - \frac{1}{2}\Big)^2 + \Big(b + \frac{1}{2}\Big)^2. \tag{12}$$

The two quadratic terms arise in equation 12 because SPPO is applied independently to the positive (native-like, 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.

D 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

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

Algorithm 1 Function-Aware Preference Alignment (FPA)

Require: Preference dataset $\mathcal{D}$, reference model π_{ref}, scale β, trade-off coefficient λ, critical-site mask $\mathcal{F}$, learning rate η, epochs T.
Ensure: Function-aware inverse folding model π_θ.
Initialize $\pi_\theta \leftarrow \pi_{\mathrm{ref}}$.
for epoch $t = 1, 2, \ldots, T$ **do**
  for mini-batch $\mathcal{D}_{\text{sub}} \subseteq \mathcal{D}$ **do**
    Compute $\mathcal{L}_{\text{Res}} = \sum_{(y^+, y^-, x) \in \mathcal{D}_{\text{sub}}} \ell_{\text{Res}}$.
    Compute $\mathcal{L}_{\text{CE}} = \sum_{(y^+, y^-, x) \in \mathcal{D}_{\text{sub}}} \ell_{\text{CE}}$.
    $\mathcal{L}_{\text{FPA}} \leftarrow \mathcal{L}_{\text{Res}} + \lambda \mathcal{L}_{\text{CE}}$.
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{FPA}}$.
  end for
end for

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

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

  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,

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

E Model Architecture Details and Training Procedure

E.1 Base Model

We use ProteinMPNN (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022) as base inverse folding models, adopting their official implementations and pretrained weights without any architectural modifications[2][3]. 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 (Widatalla et al., 2024). 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 3.

E.4 Regularization

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

[2] https://github.com/dauparas/ProteinMPNN

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

Table 3: 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	β_1=0.9, β_2=0.98, ϵ=10^{-9}	β_1=0.9, β_2=0.98, ϵ=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 4).

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 $\mathcal{C} = \{i \mid y_i^+ \neq y_i^-\}$, 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.

- **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:

$$\text{PPL}(y \mid x) = \mathbb{E}_{(y^+, y^-, x) \sim \mathcal{D}_{\text{test}}} \left[\exp\Big(-\frac{1}{L} \sum_{i=1}^{L} \log \pi_\theta(y_i^+ \mid x) \Big) \right], \tag{13}$$

  where $\pi_\theta(y_i^+ \mid x)$ is the model's probability for residue y_i^+ at position i. .
- **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:

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

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

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

- **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:

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

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

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

Table 4: 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