ProteinOPD: Towards Effective and Efficient Preference Alignment for Protein Design

Yulin Zhang[1,2]*, **He Cao**[3]*, **Zihao Jiang**[1], **Chenyi Zi**[5], **Zhipeng Zhou**[6,7], **Zijing Liu**[4], **Yu Li**[4], **Jia Li**[5], **Ziqi Gao**[1]†

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University,
[2] Jilin University, [3] AbIntelli Bio, [4] International Digital Economy Academy,
[5] Hong Kong University of Science and Technology (Guangzhou),
[6] Nanyang Technological University,
[7] Jinan-NTU Green Technology Research Institute

Abstract

Designing proteins with desired functions or properties represents a core goal in synthetic biology, therapeutic development, and drug discovery. Recent advances in protein language models (PLMs) have enabled the generation of highly designable protein sequences, while preference alignment provides a promising way to steer designs toward desired functions and properties. Nevertheless, they often trigger catastrophic forgetting of pretrained knowledge, degrading basic designability and failing to balance multiple competing objectives. To address these issues, we draw inspiration from On-Policy Distillation (OPD), an advanced post-training method renowned for mitigating catastrophic forgetting through its mode-seeking nature. In this work, we propose ProteinOPD, a multi-objective preference alignment framework that can effectively balance multiple preference objectives while maintaining the inherent designability of PLMs. ProteinOPD adapts a pretrained PLM into preference-specific teachers and distills their knowledge into a shared student via token-level OPD on the student's own trajectories. During this process, the student is aligned to a unique normalized geometric consensus of weighted teachers while ensuring bounded optimization under conflicts. This bridges the gap for OPD in multi-objective/teacher alignment. Extensive experiments show that ProteinOPD achieves substantial gains on target preference objectives without compromising the designability, with an 8× training speedup over RL-based alignment competitors.

1 Introduction

Proteins are the central functional molecules of living systems and serve as key substrates for synthetic biology [1, 2], therapeutic development [3, 4], and drug discovery [5, 6]. However, natural functional proteins occupy only a minuscule portion of the exponentially vast sequence space. While recent advances in protein language models (PLMs) [7, 8, 9, 10] have enabled the generation of highly designable sequences, the *de novo* design of proteins with tailored, user-specified properties remains a fundamental and persistent challenge in biotechnology.

In response, protein preference alignment (PPA) methods have emerged, aiming to align PLMs to generate sequences with desired properties or functions. Advanced alignment methods can be broadly categorized into two classes: **post-training** and **test-time steering**. Post-training methods reshape a PLM's output distribution by directly updating its parameters. The most straightforward approach

*Equal contribution.
†Corresponding author: Ziqi Gao (ziqigao@sz.tsinghua.edu.cn).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

is supervised fine-tuning (SFT) [11, 12, 13], which adapts the PLM on a curated set of sequences pre-screened to satisfy the target objective. Despite its simplicity, mode-covering objective of SFT tends to constrain generation to the training distribution. These constraints prevent the model from simultaneously achieving high novelty and optimal target properties. Another prevalent class of post-training methods is reinforcement learning (RL) [14, 15, 16, 17, 18, 19], which updates the PLM's policy using reward signals corresponding to the property scores of generated outputs. Although RL achieves effective alignment with desired objectives, its sequence-level supervision requires extensive rollouts, making training computationally expensive. Furthermore, when faced with new preferences, an RL policy can quickly become suboptimal, and drastic policy drift may occur, undermining the designability inherent in the PLM. Here, we use designability to refer to a model's ability to generate sequences that remain consistent with the natural protein distribution while retaining structural plausibility. The second major class, test-time steering [20, 21], refers to methods that guide a PLM to generate high-quality sequences during inference without modifying its parameters. Although it offers remarkable computational efficiency, its reliance on linear combinations of steering vectors can be insufficient for navigating the complex, non-linear trade-offs when optimizing multiple competing objectives. Hence, we raise a central question—*how can we efficiently align a PLM with multiple competing preferences without compromising its designability?*

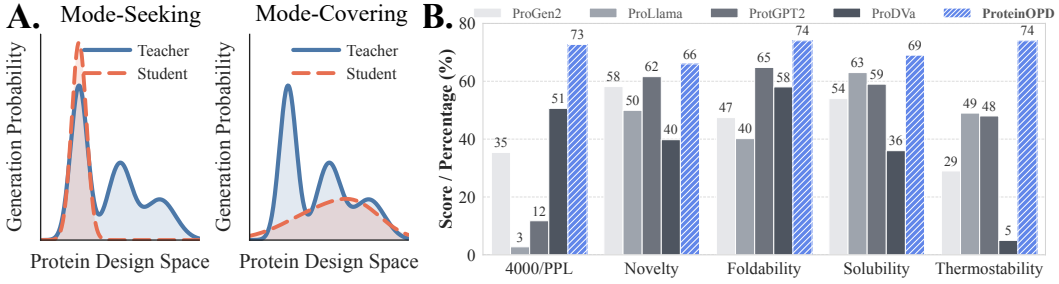

Figure 1: **A.** We show the superiority of the mode-seeking nature inherent to OPD. **B.** Performance comparison with advanced protein design and alignment methods. Our proposed ProteinOPD method achieves optimal performance in both designability and multi-property objectives.

On-policy Distillation (OPD) is valued in large language model (LLM) alignment due to its dual capability: **efficiently** adapting to new preferences while **effectively** resisting catastrophic forgetting [22, 23, 24, 25]. Specifically, OPD enables a student to learn from token-level supervision provided by a teacher model on the student's own generated trajectories. Unlike mode-covering methods such as offline distillation or SFT [26, 27, 28], OPD exhibits a mode-seeking nature [22]. As shown in Figure 1(a), OPD guides the student model to converge to the sharper and higher-reward modes of the teacher, rather than spreading probability mass across suboptimal options. This helps mitigate exposure bias and, in multi-preference scenarios, promotes clearer Pareto-optimal solutions. Moreover, compared to the sparse sequence-level rewards in conventional RL, OPD's dense token-level supervision enables more efficient and stable optimization. Consequently, the advantages of OPD align well with the key challenges of PPA. Motivated by this, we explore OPD for protein design and leverage it as an algorithmic backbone for effective and efficient PPA.

In this work, we propose ProteinOPD, a multi-objective preference alignment framework that effectively balances multiple preference objectives while preserving the intrinsic designability of pretrained PLMs. ProteinOPD first performs the SFT, which adapts a pretrained PLM into different preference-specific teachers and then distills their preference signals into a shared student through on-policy distillation. To address the research gap in multi-objective (multi-teacher) OPD, we propose the generalized OPD that aligns the student with a geometric consensus distribution, formulated as a normalized product-of-experts over multiple teachers. Theoretically, the resulting consensus distribution yields bounded optimization signals even when teachers conflict, improving training stability under noisy and antagonistic preference objectives. Empirically, ProteinOPD demonstrates the best alignment performance under both single- and multi-objective scenarios, while causing the minimal harm to designability. When using ProtGPT2 as the base model, ProteinOPD reduces PPL by 83.7%, and enhances its foldability, solubility, and thermostability by 14.8%, 16.9%, and 54.2%, respectively. ProteinOPD matches the alignment performance of RL at $1/8$ of the training time.

2 Related Work

Protein Sequence Design. Existing protein sequence design methods can be broadly divided into unconditional and condition-controlled generation. Unconditional methods, such as ProtGPT2 [7] and RITA [29], aim to generate protein sequences without explicit design constraints, thus exploring previously uncharted protein sequence space. In contrast, condition-controlled methods introduce extra signals to guide generation. ProGen [8] uses biological tags that are defined in the pre-training stage for annotation-conditioned generation, while ProLLaMA [9] extends protein language modeling to both generation and understanding tasks through continued pre-training and instruction tuning, supporting both unconditional and family-conditional generation. More recent text-guided methods, such as Pinal [30], ProDVa [31], and PAAG [32], further use natural-language descriptions or functional annotations to guide protein design. Collectively, these generative models have substantially advanced protein design by enabling the generation of novel and biologically plausible protein sequences. However, they are primarily trained to model natural protein distributions or to follow predefined conditioning signals, which do not necessarily align their outputs with the specific objectives required in protein engineering [33, 34, 35], highlighting the significance of protein preference alignment.

Protein Preference Alignment. Recent studies have investigated protein preference alignment for steering PLMs toward property- or function-aware sequence generation. Existing approaches mainly fall into two categories. Post-training methods modify model parameters after pretraining: SFT [11, 12] trains PLMs on curated sequences satisfying target criteria, whereas RL [14, 15, 16] optimizes generation policies using property-based rewards. Test-time steering [20, 21] provides an alternative by guiding generation during inference while keeping model parameters fixed, typically through steering vectors constructed from positive and negative datasets associated with the target preference. Despite their effectiveness in certain settings, these methods still face several limitations. SFT may provide limited alignment when such data are scarce or biased [13]. RL allows for direct optimization of property-based rewards. However, its training process is often inefficient and may induce catastrophic forgetting, thereby weakening the pre-acquired designability and degrading the structural and evolutionary quality of the generated proteins [17, 18]. Conversely, test-time steering avoids parameter updates, but its effectiveness depends heavily on the quality of steering directions and is difficult to align with multi-objective preferences. To address these limitations, we introduce a multi-objective on-policy distillation framework that aligns PLMs with multiple preferences in an effective and efficient manner while preserving their pretrained designability.

3 Preliminaries

3.1 On-Policy Distillation

On-Policy Distillation (OPD) computes supervision provided by teacher policy p_T on trajectories sampled from the current student policy p_S. Given a conditioning problem $x \sim \mathcal{D}_x$, the student samples a trajectory $y = (y_1, \ldots, y_N) \sim p_S(\cdot \mid x)$, where $N \triangleq |y|$ denotes the rollout length. Both policies are then evaluated on the student-generated trajectory, yielding two next-token distributions at each step n: $p_S(\cdot \mid x, y_{<n})$ and $p_T(\cdot \mid x, y_{<n})$.

A standard formulation minimizes the trajectory-averaged, token-wise divergence

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, y \sim p_S(\cdot|x)} \left[\sum_{t=1}^{N} D(p_S(\cdot \mid x, y_{<t}) \,\|\, p_T(\cdot \mid x, y_{<t})) \right], \tag{1}$$

where D can be any distribution divergence measure. Unlike offline supervised fine-tuning (SFT), which frequently suffers from exposure bias and distribution shifts, this token-level decomposition ensures that the student receives corrective signals directly on its actively visited state space. This property establishes a robust foundation for transferring generative behaviors, which we subsequently extend to address multi-objective preference alignment.

3.2 Preliminary Analysis

Before introducing our method, we empirically analyze the efficiency and efficacy of OPD. For comparison, we investigate two other representative paradigms for protein preference alignment: SFT

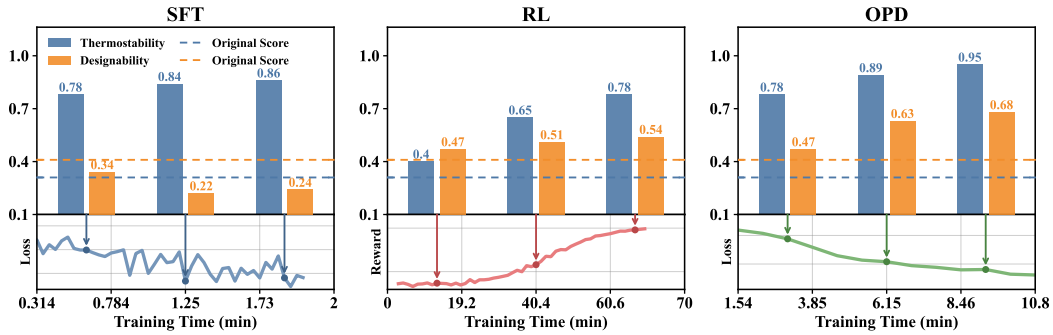

Figure 2: Preliminary comparative analysis of three popular alignment methods for improving thermostability. We track the loss/reward trend and evaluate three intermediate checkpoints on thermostability (alignment target) and designability. Dashed horizontal lines denote the capability of the base model (i.e., ProLLaMA) before alignment.

and RL. For all three methods, following MoMPNN [14], we employ TemBERTure [36] as the oracle to assess the sequence's thermostability. For SFT, we sample protein sequences from UniRef50 and use TemBERTure to estimate their thermostability. We select 100 high-quality data entries for SFT, and **the resulting model (after SFT) serves as the teacher in OPD**. For RL, we employ GRPO [37] with ProLLaMA [9] as the base model and use TemBERTure as the reward function. We take designability and thermostability as the primary metrics, and report the training time accordingly.

These exploratory experiments reveal complementary limitations of SFT and RL. As shown in Figure 2, SFT achieves strong thermostability alignment at the cost of severe degradation in designability, suggesting catastrophic forgetting of pretrained protein knowledge. This also aligns with the mode-covering nature of SFT. In contrast, RL preserves the pretrained model's designability, but the improvement of thermostability quickly saturates. For efficiency, RL is clearly the most time-consuming, due to repeated oracle evaluations and its sequence-level (sparse) supervision. Since SFT requires minimal training time and aligns effectively with new preferences, the resulting SFT model naturally serves as the teacher in OPD. Although SFT has limitations in preserving designability, OPD's resistance to forgetting ensures that the distillation retains both the student's designability and the teacher's preference scores.

Overall, our preliminary results show that OPD achieves stronger thermostability improvement effectively while mitigating catastrophic forgetting in a time-efficient way. This motivates us to further extend OPD from single-objective alignment to the multi-objective alignment setting.

4 Method

Overview. In this section, we introduce ProteinOPD, a method for multi-objective preference alignment in protein design, as illustrated in Figure 3. The method first adapts a pre-trained protein language model into multiple preference-specific teachers. **For single-objective alignment**, we apply the standard on-policy distillation framework, distilling a teacher model into the student via token-level distributional alignment on the student's own rollout trajectories. **For multi-objective alignment**, the student is aligned with a geometric consensus distribution defined as a normalized product-of-experts (PoE) over all teachers. Moreover, we emphasize the general applicability of this multi-teacher OPD, which extends the standard OPD framework to the practical scenario of multi-teacher distillation.

Teacher Model Construction. For each target preference, we construct a specialized teacher through an efficient oracle-guided adaptation process. First, we sample sequences from UniRef50 and score them using a property-specific oracle. Then, the pre-trained PLM is adapted via supervised fine-tuning (SFT) on a small high-scoring subset $\mathcal{D}_p$ of only a few hundred sequences:

$$\phi^* = \arg\min_{\phi} \sum_{(x,y)\in\mathcal{D}_p} \mathcal{L}_{\text{SFT}}\big(f(x;\phi), y\big), \tag{2}$$

where the pretrained PLM is parameterized by ϕ. In this work, we choose ProtGPT2 [7] and ProLLaMA [9] as the pretrained PLMs for unconditional and conditional generation, respectively. This process requires minimal oracle annotations and short training time, avoiding the costly online queries in RL. The adapted PLM is then used as a preference-specific teacher, providing preference-biased token-level guidance for OPD.

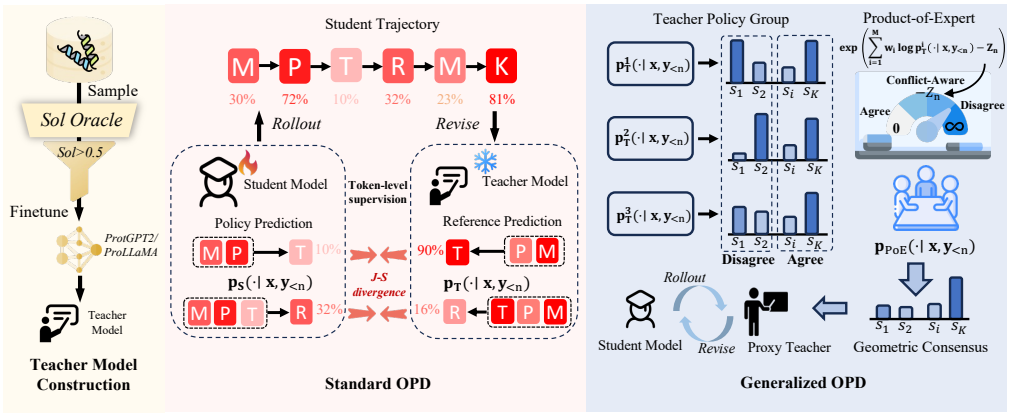

Figure 3: **Overview of our method.** Our framework comprises three stages: teacher model construction, single-objective alignment via standard OPD, and multi-objective alignment via generalized OPD. ProteinOPD flexibly adapts to each scenario, enjoying the effectiveness and efficiency of standard OPD for single-objective alignment and employing our proposed generalized OPD to manage competing objectives in multi-objective settings.

4.1 Single-Objective Alignment

We instantiate the OPD formulation in Section 3.1 on autoregressive decoder-only PLMs, where

$$p_\theta(y \mid x) = \prod_{n=1}^{N} p_\theta(y_n \mid x, y_{<n}). \tag{3}$$

Here, x denotes the conditioning context for protein generation, such as a functional or family description. The single-preference training objective directly follows the token-level OPD objective in Eq. (1). In all experiments, we instantiate the generic divergence D as the generalized Jensen–Shannon divergence:

$$\mathrm{JSD}_\beta(p_S \| p_T) = \beta\,\mathrm{KL}(p_S \,\|\, m) + (1-\beta)\,\mathrm{KL}(p_T \,\|\, m)\,, \tag{4}$$

where $m = \beta p_S + (1-\beta)p_T$, is the interpolated mixture distribution, and we set $\beta = 0.5$ by default. This full-vocabulary logit distillation provides dense, token-level feedback, making preference alignment effective and efficient.

4.2 Multi-Objective Alignment with Generalized OPD

Real-world protein design frequently necessitates the optimization of **multiple competing objectives**, a challenge that standard OPD is ill-equipped to handle. To bridge this gap, we introduce a multi-teacher strategy into the OPD framework. We consider the multi-teacher OPD setting with a student policy p_S and M teacher policies $\{p_T^i\}_{i=1}^M$. Let $\{w_i\}_{i=1}^M$ be non-negative teacher weights satisfying $\sum_{i=1}^M w_i = 1$. Our goal is to construct a unified token-level target that captures the shared preference structure across teachers, and then align the student to this target under the standard OPD framework.

Geometric Consensus Induced by the Multi-teacher Objective. A central question in multi-teacher distillation is whether the student should imitate *coverage* or *consensus*. Arithmetic mixtures emphasize coverage, assigning high probability to tokens preferred by any teacher. Here, instead, we seek a target that retains only the shared support across teachers. For each student-visited state $(x, y_{<n})$, we construct a local consensus distribution by finding the optimal distribution q:

$$\arg\min_q \sum_{i=1}^{M} w_i\,\mathrm{KL}\big(q \,\big\|\, p_T^i(\cdot \mid x, y_{<n})\big)\,. \tag{5}$$

Proposition 1. *The solution to Eq.* (5) *is a normalized product-of-experts in Eq. 6.*

The proof is straightforward and provided in Appendix D.

Normalized PoE target. As mentioned above, the solution to Eq. (5) is the normalized product-of-experts (PoE), which is a closed-form weighted geometric mean of the teacher distributions. This consensus distribution thus naturally upweights tokens that receive joint support across teachers:

$$p_{\mathrm{PoE}}(\cdot \mid x, y_{<n}) = \mathrm{softmax}\left(\sum_{i=1}^{M} w_i \log p_T^i(\cdot \mid x, y_{<n})\right) = \exp\left(\sum_{i=1}^{M} w_i \log p_T^i(\cdot \mid x, y_{<n}) - Z_n\right), \tag{6}$$

where $Z_n = \log \sum_{v \in \mathcal{V}} \exp\Big(\sum_{i=1}^{M} w_i \log p_T^i(v \mid x, y_{<n})\Big)$. Thus, computing the consensus target only requires averaging teacher log-probabilities with weights $\{w_i\}_{i=1}^{M}$, followed by a single `logsumexp` over the vocabulary.

Disagreement from Normalization. The normalization term Z_n also captures how much probability mass is jointly shared across teachers. Applying the weighted Arithmetic Mean-Geometric Mean (AM–GM) inequality, we have the following property.

Proposition 2. *For any prefix* $(x, y_{<n})$*,* $Z_n = \log \sum_{v \in \mathcal{V}} \prod_{i=1}^{M} p_T^i(v \mid x, y_{<n})^{w_i} \leq 0$*, with equality if and only if all teachers with non-zero weight induce the same next-token distribution at prefix* $x, y_{<n}$*.*

Remark 1. *We interpret* $-Z_n$ *as a natural, token-level measure of teacher disagreement (or objective conflict). By definition, when all teachers agree on an amino acid substitution,* $-Z_n$ *approaches 0. Conversely, when teachers severely conflict,* $-Z_n$ *increases. This provides a cost-free indicator to monitor and quantify the internal tug-of-war among different protein properties during training, without requiring any additional computational overhead.*

Training Objective. The **generalized OPD** for multi-objective alignment follows the same protocol in standard OPD, but replaces the single-teacher target with the normalized PoE target in Eq. (6).

$$\mathcal{L}^{\mathrm{multi}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, y \sim p_S(\cdot|x)} \left[\sum_{n=1}^{N} \mathrm{JSD}_\beta\left(p_\theta \,\|\, p_{\mathrm{PoE}}\right)\right]. \tag{7}$$

The normalized PoE defines a unified teacher policy, combining all preference-specific experts.

5 Experiments

In this section, we show that ProteinOPD consistently achieves the best alignment-designability trade-off for multi-objective preference alignment in the main result. Further analysis is organized around three research questions. We examine two representative settings: unconditional protein design with ProtGPT2, which explores uncharted protein design space, and conditional protein design with ProLLaMA, which evaluates whether generated sequences satisfy specified design conditions.

5.1 Experimental Setups

Data Construction We construct data for both unconditional and conditional protein generation from UniRef50 [38]. For the conditional setting, we follow ASPO [20] and restrict the data to the lysozyme-like superfamily, instantiating a representative family-conditioned generation task. Each sequence is annotated with preference scores from property-specific oracles. For each preference objective, we retain high-scoring sequences using predefined thresholds to construct preference-specific adaptation data. Details of the data construction are provided in Appendix B.1.

Evaluation Metrics We evaluate generated proteins from three perspectives: designability, preference alignment, and overall multi-objective performance. For designability, we report perplexity (PPL), Novelty, and, in the conditional setting, ProTrek score [39]. PPL measures the plausibility of generated sequences under a protein language model, while Novelty quantifies their dissimilarity to reference protein databases. We consider two variants: Novelty-U, computed against UniProtKB [40]

and shared across all baselines, and Novelty-T, computed against the training set of each baseline. In conditional generation, ProTrek further evaluates the semantic consistency between generated proteins and the input conditions, following ProDVa [31]. For preference alignment, we use property-specific oracles to score generated proteins. Foldability is measured by ESMFold [41]-derived pLDDT and PAE [42], solubility (Sol.) by Protein-Sol [43], and thermostability (Thermo.) by TemBERTure [36]. Overall performance is measured by hypervolume (HV), summarizing trade-offs across designability and preference-alignment metrics. Details of the definition are provided in Appendix A.

Table 1: Performance comparison of baselines and ProteinOPD in multi-objective preference alignment in the *unconditional* setting. The best and second-best scores are marked in **bold** and underline.

Models	Designability			Preference Alignment Ability						Overall
	PPL (↓)	Novelty-U (↑)	Novelty-T (↑)	pLDDT (↑)	%>70 (↑)	pAE (↓)	%<10 (↑)	Sol (↑)	Thermo (↑)	HV (↑)
`Random (U)`	1765.28	57.13	–	24.02	0.00	24.15	0.00	0.34	0.02	0.05
`Random (E)`	1747.35	59.84	–	26.9	0.39	23.99	0.78	0.44	0.03	0.01
ESM2	1467.04	52.75	60.72	47.52	5.68	25.7	0.55	0.40	0.42	0.15
ESM3	353.71	51.21	72.66	61.30	27.49	19.68	20.52	0.31	0.56	0.21
ProGen2	113.04	**85.76**	58.70	47.44	10.55	24.7	6.25	0.54	0.29	0.44
ProLLaMA	1445.8	55.07	42.95	40.17	0.0	19.4	0.39	0.63	0.49	0.06
ProtGPT2	339.57	59.97	69.17	64.77	42.58	17.69	24.61	0.59	0.48	0.33
Pinal	542.49	46.44	45.68	64.32	52.79	15.92	40.28	0.41	0.28	0.37
ProteinDT	1255.16	50.69	52.25	42.75	11.37	24.77	5.82	0.39	0.36	0.29
ProDVa	79.42	2.21	37.58	58.04	9.38	18.88	7.81	0.36	0.05	0.02
ASPO	804.45	51.57	56.77	47.99	1.95	19.63	3.12	**0.69**	0.68	0.17
MoMPNN	748.73	42.50	65.47	70.19	50.66	16.78	35.05	0.57	0.65	0.46
ProteinOPD	**55.25**	54.96	**75.68**	**74.37**	**61.72**	**13.91**	**45.7**	**0.69**	**0.74**	**0.62**

Baselines General protein design baselines include pretrained PLMs, including ESM2 [41], ESM3 [44], ProGen2 [10], ProLLaMA [9], and ProtGPT2 [7], and text-conditioned models, including Pinal [30], ProteinDT [45], and ProDVa [31]. We also include two random baselines: Random(U), which samples amino acids uniformly, and Random(E), which samples from the empirical amino acid distribution in SwissProt following ProDVa [31]. Preference-alignment baselines cover two paradigms: post-training and test-time steering. For post-training, we use SFT on ProtGPT2 [7] and ProLLaMA [9]. These models also serve as preference-specific teachers in ProteinOPD. For test-time steering, we compare with ASPO [20], which steers PLM generation with property-specific vectors. We further include MoMPNN [14] as a cross-paradigm reference baseline, since few multi-objective alignment baselines are available for sequence-only protein generation. For conditional protein design, all baselines are evaluated under the same lysozyme-like family condition. Details of the baseline settings are provided in Appendix B.3.

5.2 Performance on Multi-Objective Preference Alignment

The main results of the paper are provided by Table 1. We highlight our key findings as follows:

(1) ProteinOPD achieves the best overall performance, with an HV of 0.62. Compared with MoMPNN, the strongest cross-paradigm reference in our evaluation, ProteinOPD improves HV by 34.8%. It also outperforms the best general design baseline, ProGen2, by 40.9%, and the strongest text-conditioned baseline, Pinal, by 67.6%, indicating a stronger global trade-off between designability and preference alignment.

(2) ProteinOPD consistently improves all target preferences while preserving designability. Compared to the base model ProtGPT2, ProteinOPD improves the pLDDT, Sol, and Thermo by 14.8%, 16.9%, and 54.2%, respectively. Meanwhile, it maintains sequence-plausibility and novelty, reducing PPL by 83.7% and improving Novelty-T by 9.4%. Compared with the second-best baseline, ProteinOPD achieves the best pLDDT of 74.37, improving over MoMPNN by 6.0%, matches the best Sol of 0.69, and achieves the best Thermo of 0.74, improving over ASPO by 8.8%.

(3) Existing baselines are limited by their alignment mechanisms. General design models provide strong priors but lack explicit preference optimization. Text-conditioned models offer condition-level guidance, but weakly supervise specific preferences. Pinal remains lower than ProteinOPD by 0.28 in

solubility and 0.46 in thermostability. ASPO uses fixed test-time steering, which is bounded by the pretrained generator and cannot update model parameters. MoMPNN optimizes multiple objectives under an inverse-folding setting. The backbone constraints improve structure-related metrics, but their designability remains weaker than ProteinOPD, with 693.48 higher PPL and 12.46 lower Novelty-T. Overall, ProteinOPD yields a more balanced trade-off across designability and preference alignment.

Performance comparison of the baselines and ProteinOPD in multi-objective preference alignment in the **conditional setting** is provided in Appendix C.1.

5.3 Analysis

Beyond the main result, we perform a three-level analysis of ProteinOPD: (**RQ1**) to examine the trade-off behind its HV gains and whether it expands the Pareto frontier; (**RQ2**) to investigate, in the single-objective case, if OPD transfers preferences while mitigating the designability loss from SFT teachers; and (**RQ3**) to assess its efficiency for training and data needs for teacher construction.

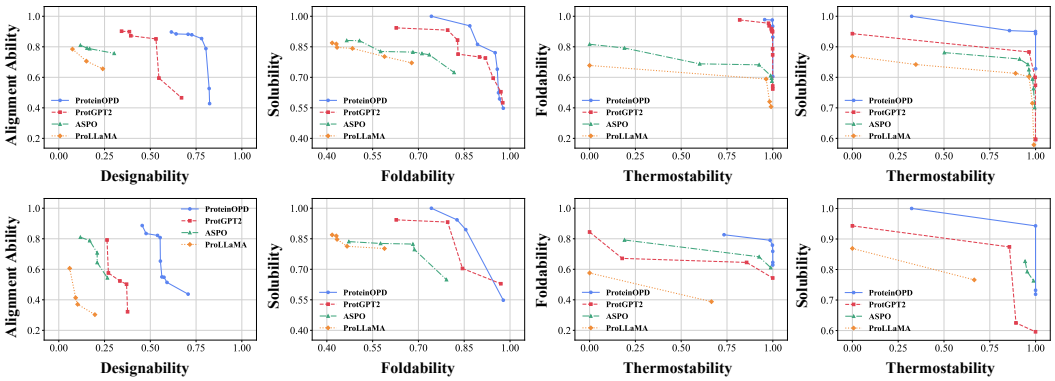

Figure 4: **Pareto-front analysis**. The **top row** reports fronts over all generated sequences, and the **bottom row** reports fronts after filtering sequences with Novelty-U > 0.7. Designability is computed from normalized PPL, and alignment ability is the average of pLDDT, solubility, and thermostability.

Pareto Frontier Expansion in Multi-Objective Preference Alignment (RQ1). As illustrated in Figure 4, across both designability–alignment and pairwise preference comparisons, ProteinOPD consistently shifts the frontier outward, indicating stronger attainable trade-offs among objectives. We further evaluate high-novelty sequences with Novelty-U > 0.7. ProteinOPD retains more favorable fronts, showing its advantage extends to exploratory regions of the sequence space rather than remaining confined to sequences close to UniProtKB. This behavior is essential for unconditional protein design, where the goal is to explore uncharted protein design space.

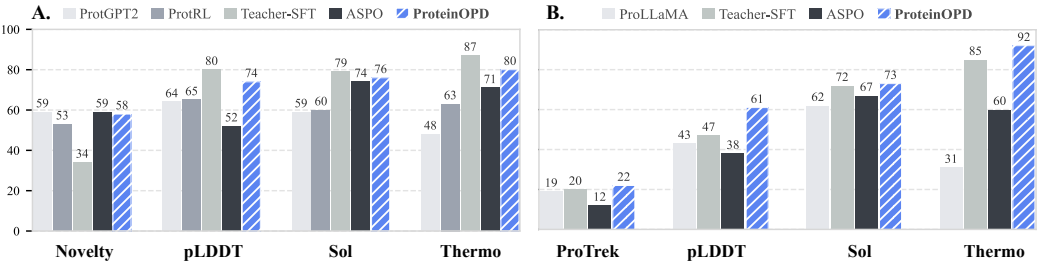

Figure 5: Single-objective alignment results in the unconditional setting (**A**) and the conditional setting (**B**). Teacher-SFT refers to the constructed teacher model adapted to the high-scoring subset.

Performance on Single-Objective Preference Alignment (RQ2). Here, we analyze the results of ProteinOPD in the single-objective setting. Figure 5A shows that OPD can transfer preference signals from SFT teachers while substantially mitigating the designability loss caused by direct fine-tuning. In the unconditional setting, Teacher-SFT improves pLDDT, Sol, and Thermo substantially, but at the

cost of a 42.4% novelty drop. In contrast, ProteinOPD preserves most of these property gains while reducing novelty by only 1.7% relative to ProtGPT2. This gap directly supports our claim that OPD mitigates the forgetting behavior of SFT-based alignment.

In the conditional setting, the advantage of ProteinOPD is even more pronounced because alignment must be achieved without breaking condition consistency, as shown in Figure 5B. Compared with Teacher-SFT, ProteinOPD improves ProTrek Score by 10.0%, indicating better consistency with the original conditional generation capability. It also further improves pLDDT, solubility, and thermostability by 29.8%, 1.4%, and 8.2%, respectively. The behavior is consistent with the nature of OPD: the student is corrected on its own visited states, rather than being forced to cover the full distribution of the high-scoring SFT data. Compared with ProtRL and ASPO, ProteinOPD provides more stable single-objective alignment while better preserving either novelty in unconditional generation or condition fidelity in conditional generation.

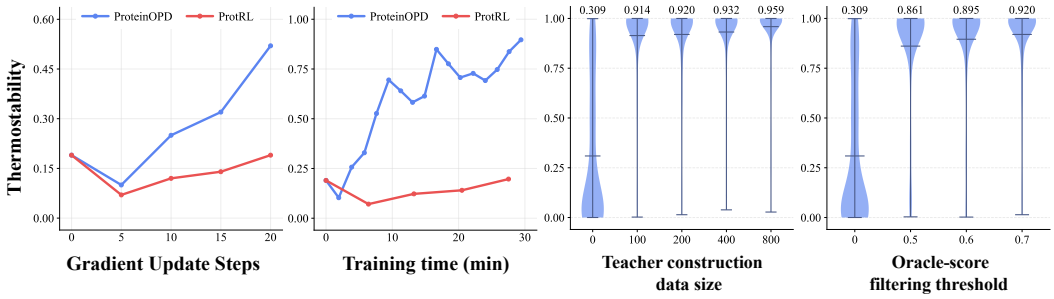

Figure 6: **Efficiency analysis.** Online cost is evaluated by comparing ProteinOPD and ProtRL over gradient steps and wall-clock time. Offline cost is evaluated by varying the teacher-construction data size and oracle-score filtering threshold. The last two panels report the best evaluation performance within 100 training steps, and top numbers denote mean predicted thermostability.

Efficiency in Computation and Data (RQ3). Beyond alignment quality, a practical protein design method must be computationally viable, especially given expensive oracle queries. We therefore assess whether ProteinOPD maintains the efficiency advantages in data and training computation. As shown in Figure 6, ProteinOPD proves efficient in both online and offline alignment. For online cost, ProteinOPD improves thermostability much faster than ProtRL. Within 20 gradient steps, ProteinOPD increases the thermostability score from about 0.19 to 0.52, whereas ProtRL remains around 0.19. Under the same wall-clock comparison, ProteinOPD reaches about 0.70 within 10 minutes and further improves to 0.89 by 30 minutes, while ProtRL stays below 0.20. ProteinOPD reaches a thermostability score of 0.25 in 3.86 minutes, whereas ProtRL requires 31.2 minutes to reach a comparable score. This corresponds to an 8$\times$ training speed-up. This indicates that token-level OPD provides a denser and more efficient learning signal than reward-based online optimization. For offline cost, we find that using 100 filtered sequences already yields a mean thermostability of 0.914, close to that of a larger quantity. As for data quality, when raising the filtering threshold from 50 to 70, the mean score increases from 0.861 to 0.920. These results show that ProteinOPD enables effective teacher construction with limited offline data, with additional gains from oracle-based filtering.

Case Visualization. We present generated examples of ProteinOPD and ASPO, ensuring that the thermostability of both exceeds 0.95, and show their scores of pLDDT and solubility. Moreover, we ensure novelty against UniRef by requiring a max seq. identity below 5%. Thus, ProteinOPD enables effective multi-attribute alignment and the creation of novel sequences outside current databases in practical use.

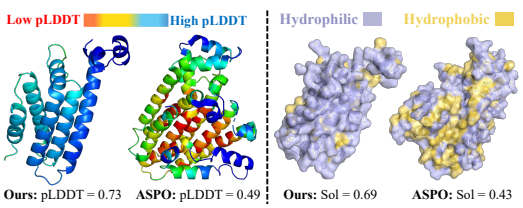

Figure 7: Residue-level visualization.

6 Conclusion

We present ProteinOPD, a multi-objective preference alignment framework for protein language models that aligns protein generation with desired preferences while preserving pretrained designability. ProteinOPD constructs preference-specific teachers through lightweight oracle-guided adaptation and transfers their preference guidance to a shared student via token-level on-policy distillation on student-generated trajectories. For multi-objective preference alignment, ProteinOPD extends OPD to the multi-teacher setting with a normalized product-of-experts consensus target over preference-specific teachers. Experiments across unconditional and conditional protein generation show that ProteinOPD achieves superior designability–preference trade-offs compared with existing baselines, while delivering an $8\times$ training speedup over RL-based alignment. These results establish OPD as an effective and efficient mechanism for preference-aligning protein language models.

References

[1] Ahmad S Khalil and James J Collins. Synthetic biology: applications come of age. *Nature Reviews Genetics*, 11(5):367–379, 2010.

[2] Xu Yan, Xu Liu, Cuihuan Zhao, and Guo-Qiang Chen. Applications of synthetic biology in medical and pharmaceutical fields. *Signal transduction and targeted therapy*, 8(1):199, 2023.

[3] Andres Cubillos-Ruiz, Tingxi Guo, Anna Sokolovska, Paul F Miller, James J Collins, Timothy K Lu, and Jose M Lora. Engineering living therapeutics with synthetic biology. *Nature Reviews Drug Discovery*, 20(12):941–960, 2021.

[4] Sasha B Ebrahimi and Devleena Samanta. Engineering protein-based therapeutics through structural and chemical design. *Nature communications*, 14(1):2411, 2023.

[5] Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature biotechnology*, 42(2):275–283, 2024.

[6] Lei Lu, Xuxu Gou, Sophia K Tan, Samuel I Mann, Hyunjun Yang, Xiaofang Zhong, Dimitrios Gazgalis, Jesús Valdiviezo, Hyunil Jo, Yibing Wu, et al. De novo design of drug-binding proteins with predictable binding energy and specificity. *Science*, 384(6691):106–112, 2024.

[7] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[8] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[9] Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *IEEE Transactions on Artificial Intelligence*, 2025.

[10] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

[11] Jiawei Luo, Xianliang Liu, Jiahao Li, Qingcai Chen, and Junjie Chen. Flexible and controllable protein design by prefix-tuning large-scale protein language models. *BioRxiv*, pages 2023–12, 2023.

[12] Geraldene Munsamy, Ramiro Illanes-Vicioso, Silvia Funcillo, Ioanna T Nakou, Sebastian Lindner, Gavin Ayres, Lesley S Sheehan, Steven Moss, Ulrich Eckhard, Philipp Lorenz, et al. Conditional language models enable the efficient design of proficient enzymes. *bioRxiv*, pages 2024–05, 2024.

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

[14] Xiaoyang Hou, Junqi Liu, Chence Shi, Xin Liu, Zhi Yang, and Jian Tang. Property-driven protein inverse folding with multi-objective preference alignment. *arXiv preprint arXiv:2603.06748*, 2026.

[15] 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.

[16] Xiangyu Liu, Yi Liu, Silei Chen, and Wei Hu. Controllable protein sequence generation with llm preference optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 505–513, 2025.

[17] Filippo Stocco, Maria Artigues-Lleixa, Andrea Hunklinger, Talal Widatalla, Marc Guell, and Noelia Ferruz. Guiding generative protein language models with reinforcement learning. *arXiv preprint arXiv:2412.12979*, 2024.

[18] Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, pages 2024–05, 2024.

[19] Zihao Jiang, Miao Peng, Zhenyan Shan, Wenjie Xu, Ben Liu, Gong Chen, Ziqi Gao, and Min Peng. Tkg-thinker: Towards dynamic reasoning over temporal knowledge graphs via agentic reinforcement learning. *arXiv preprint arXiv:2602.05818*, 2026.

[20] Long-Kai Huang, Rongyi Zhu, Bing He, and Jianhua Yao. Steering protein language models. *arXiv preprint arXiv:2509.07983*, 2025.

[21] Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, 2025.

[22] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[23] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

[24] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[25] Ziqi Gao, Chenyi Zi, Zijing Liu, Ziqiao Meng, Yu Li, and Jia Li. Learning the interaction prior for protein-protein interaction prediction: A model-agnostic approach. *arXiv preprint arXiv:2605.09964*, 2026.

[26] Miaosen Zhang, Yishan Liu, Shuxia Lin, Xu Yang, Qi Dai, Chong Luo, Weihao Jiang, Peng Hou, Anxiang Zeng, Xin Geng, et al. Towards on-policy sft: Distribution discriminant theory and its applications in llm training. *arXiv preprint arXiv:2602.12222*, 2026.

[27] Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

[28] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

[29] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

[30] Fengyuan Dai, Shiyang You, Yudian Zhu, Yuan Gao, Lihao Fu, Xibin Zhou, Jin Su, Chentong Wang, Yuliang Fan, Xiaoxiao Ma, et al. Toward de novo protein design from natural language. *BioRxiv*, pages 2024–08, 2024.

[31] Nuowei Liu, Jiahao Kuang, Yanting Liu, Tao Ji, Changzhi Sun, Man Lan, and Yuanbin Wu. Protein design with dynamic protein vocabulary. *arXiv preprint arXiv:2505.18966*, 2025.

[32] Chaohao Yuan, Songyou Li, Geyan Ye, Yikun Zhang, Long-Kai Huang, Wenbing Huang, Wei Liu, Jianhua Yao, and Yu Rong. Annotation-guided protein design with multi-level domain alignment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1855–1866, 2025.

[33] Filippo Stocco, Michele Garibbo, and Noelia Ferruz. Steering generative models for protein design: Aligning and conditioning strategies. *Current Opinion in Structural Biology*, 98:103250, 2026.

[34] Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. Hierarchical graph learning for protein–protein interaction. *Nature communications*, 14(1):1093, 2023.

[35] Ziqi Gao, Zijing Liu, Yu Li, and Jia Li. Towards stable representations for protein interface prediction. *Advances in Neural Information Processing Systems*, 37:73079–73097, 2024.

[36] Chiara Rodella, Symela Lazaridi, and Thomas Lemmin. Temberture: advancing protein thermostability prediction with deep learning and attention mechanisms. *Bioinformatics Advances*, 4(1):vbae103, 2024.

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[38] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and the UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[39] Jin Su, Xibin Zhou, Xuting Zhang, and Fajie Yuan. Protrek: Navigating the protein universe through tri-modal contrastive learning. *bioRxiv*, pages 2024–05, 2024.

[40] Uniprot: the universal protein knowledgebase in 2023. *Nucleic acids research*, 51(D1):D523–D531, 2023.

[41] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022: 500902, 2022.

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

[43] Max Hebditch, M Alejandro Carballo-Amador, Spyros Charonis, Robin Curtis, and Jim Warwicker. Protein–sol: a web tool for predicting protein solubility from sequence. *Bioinformatics*, 33(19):3098–3100, 2017.

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

[45] Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. A text-guided protein design framework. *Nature Machine Intelligence*, 7(4):580–591, 2025.

[46] Jiahao Kuang, Nuowei Liu, Jie Wang, Changzhi Sun, Tao Ji, and Yuanbin Wu. Pdfbench: A benchmark for de novo protein design from function. *arXiv preprint arXiv:2505.20346*, 2025.

A Details of Metrics

This section provides detailed definitions of the evaluation metrics used in our experiments.

Perplexity. Perplexity (PPL) measures how likely a generated protein sequence is under a reference protein language model. Given a generated sequence $x = (x_1, x_2, \ldots, x_L)$, we compute its perplexity as

$$\mathrm{PPL}(x) = \exp\left(-\frac{1}{L}\sum_{t=1}^{L}\log p_{\mathrm{ref}}(x_t \mid x_{<t})\right), \tag{8}$$

where p_{ref} denotes the next-token distribution predicted by the reference PLM. In our experiments, we use ProtGPT2 as the evaluator. A lower PPL indicates that the generated sequence is more consistent with the distribution learned by the reference PLM, and is therefore used as a proxy for sequence plausibility.

Novelty. Novelty measures how different the generated proteins are from known or training sequences. Let $\mathcal{R}$ denote the reference set, such as the training set or a database of natural proteins, and let $\mathrm{sim}(P, R)$ denote the sequence similarity between a generated sequence P and a reference sequence $R \in \mathcal{R}$. The novelty of P is defined as its dissimilarity to the nearest reference sequence:

$$\mathrm{Novelty}(P) = 1 - \max_{R\in\mathcal{R}} \mathrm{sim}(P, R), \tag{9}$$

where $\mathrm{sim}(P, R)$ denotes the normalized sequence similarity between P and R, computed using MMseqs2. The final novelty score is obtained by averaging over all generated proteins:

$$\mathrm{Novelty}(\mathcal{G}) = \frac{1}{|\mathcal{G}|}\sum_{P\in\mathcal{G}} \mathrm{Novelty}(P). \tag{10}$$

A higher novelty score indicates that the generated proteins are less similar to known sequences and are less likely to be memorized from the training data.

ProTrek Score. For conditional protein generation, we use the ProTrek Score to evaluate whether the generated protein is semantically aligned with the input condition. Given a textual condition c and a generated protein sequence P, ProTrek encodes them into a shared representation space using its text encoder and protein sequence encoder, respectively. The ProTrek Score is computed as the cosine similarity between the two embeddings:

$$\mathrm{ProTrek}(c, P) = \cos\left(\tau_c(c), \tau_p(P)\right), \tag{11}$$

where $\tau_c(c)$ denotes the embedding of the textual condition and $\tau_p(P)$ denotes the embedding of the generated protein sequence. A higher ProTrek Score indicates stronger semantic consistency between the generated sequence and the specified condition.

pLDDT. The predicted Local Distance Difference Test (pLDDT) is used to evaluate the foldability of generated proteins. For a generated sequence P of length L, a structure prediction model produces a residue-level confidence score $\mathbf{s} = (s_1, s_2, \ldots, s_L)$, where each $s_i \in [0, 100]$ measures the confidence of the predicted local structure around residue i. We report the average pLDDT over all residues:

$$\mathrm{pLDDT}(P) = \frac{1}{L}\sum_{i=1}^{L} s_i. \tag{12}$$

Higher pLDDT values indicate more confident structure predictions and are commonly used as a proxy for protein foldability.

PAE. Predicted Aligned Error (PAE) measures the expected positional error between residue pairs in a predicted structure. For a generated protein of length L, the structure prediction model outputs a matrix $\mathbf{E} \in \mathbb{R}^{L\times L}$, where E_{ij} represents the predicted error of residue i when the predicted and actual structures are aligned on residue j. We compute the average PAE as

$$\mathrm{PAE}(P) = \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L} E_{ij}. \tag{13}$$

Lower PAE values indicate more reliable global structure prediction, especially with respect to residue-pair geometry, domain packing, and long-range structural arrangement.

Solubility We download the Protein-Sol oracle as the solubility predictor from `https://protein-sol.manchester.ac.uk/software`. The predictor outputs a score ranging from 0 to 1 for each sequence.

Thermostability We download the TemBERTure model from `https://github.com/ibmm-unibe-ch/TemBERTure` and use the **temBERTure_CLS** mode. The model outputs a Thermophilic score ranging from 0 to 1 for each sequence.

Hypervolume. Hypervolume (HV) measures the dominated volume of the Pareto set in the normalized objective space. For each generated protein P, let $m_k(P)$ denote its raw value on the k-th metric. We first apply a metric-specific transformation

$$\psi_k(m) = \begin{cases} \log(1+m), & \text{if } m \text{ is PPL,} \\ m, & \text{otherwise,} \end{cases} \tag{14}$$

where the $\log(1+\cdot)$ transformation compresses the large numerical range of PPL. We then orient all metrics such that larger values are better:

$$a_k(P) = s_k \psi_k(m_k(P)), \quad s_k = \begin{cases} +1, & \text{for metrics to maximize,} \\ -1, & \text{for metrics to minimize.} \end{cases} \tag{15}$$

Here, metrics to maximize include Novelty, pLDDT, solubility, thermostability, and ProTrek Score, while metrics to minimize is PPL.

The oriented scores are min-max normalized over all generated sequences from all compared methods under the same experimental setting:

$$z_k(P) = \frac{a_k(P) - a_k^{\min}}{a_k^{\max} - a_k^{\min}}, \quad a_k^{\min} = \min_{P \in \mathcal{U}} a_k(P), \quad a_k^{\max} = \max_{P \in \mathcal{U}} a_k(P), \tag{16}$$

where $\mathcal{U}$ denotes the union of generated sequences from all methods. The normalized objective vector is therefore $\mathbf{z}(P) = (z_1(P), \ldots, z_K(P)) \in [0,1]^K$.

Given the generated set $\mathcal{G}$ of a method, we extract the non-dominated subset $\mathrm{ND}(\mathcal{G})$ in the normalized objective space. The HV is defined as

$$\mathrm{HV}(\mathcal{G}) = \lambda_K \left(\bigcup_{P \in \mathrm{ND}(\mathcal{G})} [\mathbf{0}, \mathbf{z}(P)] \right), \tag{17}$$

where $\lambda_K(\cdot)$ denotes the K-dimensional Lebesgue measure and $\mathbf{0}$ is used as the reference point. A larger HV indicates that the generated sequences achieve a better trade-off across designability and preference-alignment objectives.

B Details of Experimental Setups

B.1 Data Construction Details

We construct preference-specific datasets using the same oracle models described in Section 5.1. For each sampled sequence, foldability is evaluated using ESMFold-derived pLDDT and pAE, thermostability using TemBERTure, and solubility using Protein-Sol.

We apply separate filtering thresholds in the unconditional and conditional settings. For unconditional generation, sequences are sampled from UniRef50 and filtered independently for pLDDT, Sol and Thermo using 80, 0.7, and 0.5, respectively. For conditional generation, sequences are sampled from the conditional protein subset and independently filtered for pLDDT, Sol, and Thermo using thresholds of 80, 0.7, and 0.7, respectively. For all experiments, we use 200 sequences to finetune the pre-trained model unless otherwise specified.

B.2 Training Details

All training runs are conducted on 2 NVIDIA A800-SXM4-80GB GPUs using AdamW with bfloat16 precision and a warm-up ratio of 0.1. The maximum completion length is set to 512 tokens, with

Table 2: Training configuration for teacher construction and ProteinOPD.

Parameter	ProtGPT2 Teacher	ProLLaMA Teacher	ProteinOPD (ProtGPT2)	ProteinOPD (ProLLaMA)
Base Model	ProtGPT2	ProLLaMA	ProtGPT2	ProLLaMA
Training Stage	Teacher SFT	Teacher SFT	OPD	OPD
Fine-tuning Method	Prefix Tuning	LoRA Tuning	LoRA Tuning	LoRA Tuning
Learning Rate	1×10^{-3}	2×10^{-5}	2×10^{-5}	2×10^{-5}
Training Steps	50	20	200	100
Prefix Length	100	–	–	–
LoRA Rank (r)	–	8	64	8
LoRA Alpha (α)	–	16	128	16
LoRA Target Modules	–	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj	c_attn, c_proj, c_fc	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

top-k sampling of 200 and top-p sampling of 0.95. For ProteinOPD, the student rollout temperature is set to 1.0, while teacher distributions are computed with a temperature of 0.7. The JSD loss uses $\beta = 0.5$ for all OPD experiments. For multi-teacher OPD, we use the same distillation weights in both unconditional and conditional settings. Specifically, the Foldability, Thermostability, and Solubility teachers are assigned weights of 0.3, 0.4, and 0.3, respectively. More implementation details are provided in Table 2.

B.3 Baseline Details

The multi-objective preference alignment baselines are limited. For ASPO, we adapt its test-time steering mechanism to the multi-objective setting by linearly combining property-specific steering vectors with predefined weights. Specifically, the steering vector is constructed as a weighted sum of the thermostability, solubility, and foldability vectors, with weights 1.0, 0.5, and 0.5, respectively. This combined vector is then used to guide sequence generation toward multiple target preferences.

MoMPNN is included as a cross-paradigm reference baseline rather than a task-identical sequence-generation baseline. Unlike ProteinOPD, which generates protein sequences without requiring structural inputs, MoMPNN operates in an inverse-folding setting and requires backbone structures as design conditions. Since few multi-objective preference-alignment baselines are available for sequence-only protein generation, we include MoMPNN to provide an additional reference point for multi-objective optimization. We randomly sample 256 protein backbones and use MoMPNN to design one sequence for each backbone. The resulting sequences are then evaluated with the same oracle-based preference metrics and designability metrics as all other methods. Therefore, the comparison should be interpreted as a downstream performance reference across design paradigms rather than a strictly task-identical comparison.

Since ESM2 and ESM3 are auto-encoding protein language models rather than autoregressive generators, they cannot directly generate protein sequences from left to right. Following ASPO [20], we use an iterative masked-token infilling protocol to obtain generated sequences from these models. Specifically, we first sample reference protein sequences from UniRef50 as reference sequences. In each iteration, we randomly select and mask 10% of the tokens without replacement, then generate them using the model.

ProLLaMA is applicable to both unconditional and conditional generation. Therefore, we include ProLLaMA as a baseline in both settings. In the unconditional setting, it is used to generate protein sequences without additional family constraints. In the conditional setting, it is prompted with the lysozyme-like superfamily description.

For text-guided baselines, we use fixed, property-explicit prompts to ensure that the target preference is directly provided to the model. For single-objective generation, the prompt template is "Design novel protein sequences with high [PROPERTY]." where [PROPERTY] is instantiated as foldability, solubility, or thermostability. For multi-objective generation, we use "Design novel protein sequences with high foldability, high solubility, and high thermostability." In the conditional setting, we additionally include the family constraint: "Design novel protein sequences from the lysozyme-like superfamily with high [PROPERTY]." or its multi-objective counterpart.

To ensure a fair comparison, all baselines are evaluated using the same generation and evaluation protocol unless otherwise specified. We generate 256 protein sequences for each method with

temperature 0.7, and a maximum generation length of 512 tokens. We evaluate generated sequences using designability and preference-alignment metrics. Specifically, pLDDT, pAE, ProTrek Score, Novelty, and PPL are computed based on PDFBench [46]. For each metric, we report the mean value over all 256 generated sequences.

C Additional Experimental Results

C.1 Performance on Multi-objective Preference Alignment in the conditional setting

Table 3: Performance comparison of baselines and ProteinOPD in multi-objective preference alignment in the *conditional* setting. The best and second-best scores are marked in **bold** and underline.

Models	Designability		Foldability		Solubility	Thermostability	HV
	PPL (↓)	ProTrek % (↑)	pLDDT (↑)	pAE (↓)	Sol (↑)	Thermo (↑)	HV (↑)
ProLLaMA	1473.68	<u>19.18</u>	43.03	20.78	0.62	0.31	0.09
ASPO	1264.59	8.46	43.27	23.07	**0.76**	<u>0.62</u>	<u>0.44</u>
ProDVa	<u>953.33</u>	18.95	<u>56.08</u>	<u>20.41</u>	0.30	0.13	0.05
ProteinOPD	**766.6**	**22.05**	**61.36**	**15.56**	<u>0.64</u>	**0.8**	**0.61**

In the conditional multi-objective setting, ProteinOPD achieves the strongest overall trade-off among competing objectives. Compared with the base ProLLaMA model, ProteinOPD reduces PPL by 707.08 points, from 1473.68 to 766.60, and improves the ProTrek score by 2.87 points, from 19.18 to 22.05. This suggests that ProteinOPD improves sequence plausibility while maintaining, and even strengthening, alignment with the conditional input. For foldability, ProteinOPD improves pLDDT by 18.33 points and reduces pAE by 5.22 points compared with ProLLaMA, indicating more plausible predicted structures. Although ASPO achieves a higher solubility score than ProteinOPD, ProteinOPD improves thermostability by 0.18 over ASPO and achieves a 0.17 higher hypervolume score. These results indicate that ProteinOPD does not merely optimize a single property, but provides a better overall balance across designability, foldability, solubility, and thermostability.

C.2 Independent PPL Evaluation

Since ProtGPT2 is used as the primary evaluator for PPL in our main experiments, we further compute PPL using ProGen2 as an independent autoregressive protein language model. This evaluation provides an additional check on whether the improved designability is specific to the ProtGPT2 evaluator. ProteinOPD obtains a ProGen2-based PPL of 5.83, lower than that of the original ProtGPT2 generator (7.25). This result suggests that the generated sequences remain more plausible under an evaluator that is not used in either teacher construction or OPD training.

D Proofs for Generalized OPD

This section provides the proofs for the normalized product-of-experts target and the normalization-based disagreement measure used in generalized OPD. Let $\mathcal{V}$ denote the token vocabulary, and let $\Delta(\mathcal{V})$ denote the probability simplex over $\mathcal{V}$. For each teacher $i \in \{1, \dots, M\}$, $p_T^i(\cdot \mid x, y_{<n})$ denotes its next-token distribution at this prefix. The teacher weights satisfy $w_i \geq 0$ and $\sum_{i=1}^M w_i = 1$. Since PLMs parameterize next-token distributions through softmax outputs, we assume $p_T^i(v \mid x, y_{<n}) > 0$ for all $v \in \mathcal{V}$.

Proof of Proposition 1. At the prefix $(x, y_{<n})$, generalized OPD constructs the local consensus target by solving

$$q_n^\star = \arg\min_{q_n \in \Delta(\mathcal{V})} \sum_{i=1}^M w_i \mathrm{KL}\left(q_n(\cdot) \,\middle\|\, p_T^i(\cdot \mid x, y_{<n})\right). \tag{18}$$

Expanding the objective gives

$$\sum_{i=1}^{M} w_i \mathrm{KL}\left(q_n \,\|\, p_T^i(\cdot \mid x, y_{<n})\right) = \sum_{v \in \mathcal{V}} q_n(v) \left[\log q_n(v) - \sum_{i=1}^{M} w_i \log p_T^i(v \mid x, y_{<n})\right]. \tag{19}$$

Define the normalized product-of-experts distribution

$$p_{\mathrm{PoE}}(v \mid x, y_{<n}) = \exp\left(\sum_{i=1}^{M} w_i \log p_T^i(v \mid x, y_{<n}) - Z_n\right), \tag{20}$$

where

$$Z_n = \log \sum_{u \in \mathcal{V}} \exp\left(\sum_{i=1}^{M} w_i \log p_T^i(u \mid x, y_{<n})\right). \tag{21}$$

By substituting this definition into the expanded objective, we obtain

$$\sum_{i=1}^{M} w_i \mathrm{KL}\left(q_n \,\|\, p_T^i(\cdot \mid x, y_{<n})\right) = \sum_{v \in \mathcal{V}} q_n(v) \log \frac{q_n(v)}{p_{\mathrm{PoE}}(v \mid x, y_{<n})} - Z_n \tag{22}$$

$$= \mathrm{KL}\left(q_n \,\|\, p_{\mathrm{PoE}}(\cdot \mid x, y_{<n})\right) - Z_n. \tag{23}$$

Since Z_n is independent of q_n and the KL divergence is non-negative, the minimum is achieved if and only if

$$q_n^\star(\cdot) = p_{\mathrm{PoE}}(\cdot \mid x, y_{<n}). \tag{24}$$

Therefore, the solution to the weighted multi-teacher objective is exactly the normalized product-of-experts target in Eq. (6). This proves Proposition 1.

Proof of Proposition 2. For the same prefix $(x, y_{<n})$, the normalization term in Eq. (6) is

$$Z_n = \log \sum_{v \in \mathcal{V}} \prod_{i=1}^{M} p_T^i(v \mid x, y_{<n})^{w_i}. \tag{25}$$

For each token v, the weighted AM–GM inequality gives

$$\prod_{i=1}^{M} p_T^i(v \mid x, y_{<n})^{w_i} \le \sum_{i=1}^{M} w_i p_T^i(v \mid x, y_{<n}). \tag{26}$$

Summing both sides over the vocabulary yields

$$\sum_{v \in \mathcal{V}} \prod_{i=1}^{M} p_T^i(v \mid x, y_{<n})^{w_i} \le \sum_{v \in \mathcal{V}} \sum_{i=1}^{M} w_i p_T^i(v \mid x, y_{<n}) \tag{27}$$

$$= \sum_{i=1}^{M} w_i \sum_{v \in \mathcal{V}} p_T^i(v \mid x, y_{<n}) = 1. \tag{28}$$

Taking the logarithm gives

$$Z_n \le 0. \tag{29}$$

The equality condition of the weighted AM–GM inequality holds if and only if, for every token v and every teacher with $w_i > 0$, the values $p_T^i(v \mid x, y_{<n})$ are identical across teachers. Thus, $Z_n = 0$ if and only if all non-zero-weight teachers induce the same next-token distribution at the prefix $(x, y_{<n})$; otherwise, $Z_n < 0$. Equivalently, $-Z_n \ge 0$ is zero under complete agreement and increases as the shared probability mass among teachers decreases. This proves Proposition 2.

E Limitations

In this work, we evaluate generated proteins using widely adopted computational Oracles for foldability, thermostability, solubility, and conditional sequence–function alignment. While these Oracles provide efficient and informative proxies for large-scale protein generation, they may not fully capture the biological validity, functional efficacy, or experimental feasibility of the designed proteins. Therefore, future work will focus on wet-lab experiments to empirically validate the structural and biochemical properties of the generated sequences.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims are supported by the proposed framework in the method and the experimental results on representative datasets.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Include limitations in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Include proof in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We describe the datasets, preprocessing protocol, evaluation splits, baselines, metrics, and implementation details needed to reproduce the main experiment results.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The datasets used in this work are publicly available, and we provide detailed descriptions of data preprocessing, experimental settings, and evaluation protocols. The code is not publicly released at submission time, but we plan to release it upon publication.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the datasets, train-test splits, evaluation metrics, baselines, model settings, and key hyperparameters used in the experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We report the average results over independent runs, but do not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the details of training in the appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics and uses publicly available biological datasets and curated knowledge resources without involving private or sensitive human-subject data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the potential positive impact of the efficient protein design framework in the introduction motivation part.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work does not release high-risk models, scraped datasets, or other assets with substantial misuse risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the sources of the datasets in the experimental setups part.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce a new public dataset.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve human-subject studies and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLM is used only for polishing the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.