LABEL-FREE FINE-TUNING: YOUR PROTEIN LANGUAGE MODEL IS IMPLICITLY A REWARD MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training of protein language models (PLMs) typically requires verifiable labels from simulations or wet-lab experiments, which are costly to acquire in real world. To overcome the supervision bottleneck, this paper investigates the emerging paradigm of label-free fine-tuning (LFT) to improve protein generation quality and fidelity without ground-truth labels. Prior works have explored various proxy rewards for LFT, yet their design suffer from inconsistent performance across sampling schemes and downstream tasks. To this end, we propose pointwise mutual information (PMI) as a principled reward, which measures the difference between the conditional and unconditional log probability of a given PLM. Despite its simplicity, PMI is motivated by Bayes' rule to effectively leverage a well-trained PLM as an implicit reward model, analogous to classifier-free guidance in diffusion models. Extensive experiments also demonstrate that our framework is amenable and transferable to a broad class of PLMs and protein design tasks.

1 INTRODUCTION

Large language models (LLMs) acquire broad capabilities through pretraining on large-scale corpora, while post-training adapts these capabilities to specific instructions and objectives (Ouyang et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024). A similar paradigm is emerging in computational biology, where biomolecular foundation models learn from biological sequences and their associated properties. Exemplified by protein language models (PLMs), these models support the generation of proteins conditioned on functional annotations, sequence fragments, and structural information (Madani et al., 2023; Watson et al., 2023; Hayes et al., 2025). Translating this generative capacity into reliable biological design, however, requires aligning model outputs with the properties specified by the user. Post-training therefore offers an important mechanism for converting knowledge acquired from biological data into controllable design capabilities.

Supervised fine-tuning and preference-based optimization have become widely adopted approaches to language-model adaptation (Ziegler et al., 2019; Wei et al., 2022; Rafailov et al., 2023). Their effectiveness has also motivated applications to biological generation, including aligning PLMs with experimentally measured protein fitness (Stocco et al., 2024; Hayes et al., 2025; Widatalla et al., 2026; Blalock et al., 2026). However, these approaches depend on access to suitable demonstrations, preference annotations, or task-specific feedback. This supervision bottleneck becomes particularly pronounced in scientific domains: evaluating a generated candidate may require specialized simulations or costly experimental measurements, and reliable feedback may be unavailable for new design objectives. Developing adaptation methods that make better use of knowledge already encoded in pretrained models is therefore a promising direction for scalable biological design.

Recent progress in *label-free fine-tuning* (LFT) explores this possibility by constructing learning signals without ground-truth labels for the adaptation examples. In LLMs, intrinsic signals such as majority agreement and semantic entropy have enabled self-improvement on reasoning tasks (Zuo et al., 2025; Zhang et al., 2025), while executable environments can provide external verification for self-generated problems (Zhao et al., 2026). Nevertheless, intrinsic rewards can reinforce incorrect predictions when model confidence is misaligned with correctness (He et al., 2026), and inexpensive external verification is difficult to obtain for many biological properties. Recent work explores label-free fine-tuning of protein generative models by constructing adaptation datasets from self-generated sequences and automatically computed proxy rewards, without requiring ground-truth labels or

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

experimentally annotated preferences for the generated examples (Wang et al., 2025b; Li et al., 2026). These approaches derive training signals from model predictions, including intrinsic uncertainty, semantic consistency, and computationally assessed structural quality. Although encouraging, existing approaches leave open the question of how to construct rewards that transfer across design conditions without task-specific supervision. In particular, the proxy rewards studied by Li et al. (2026) exhibit sensitivity to sampling temperatures and tasks, while selecting among them using ground-truth performance reintroduces supervision into reward selection. These observations motivate a central question: *can a generative model itself provide a transferable measure for how well its outputs satisfy a given condition?*

In this work, we investigate *pointwise mutual information* (PMI) as a principled basis for such a reward, using protein generation as a testbed. Our key insight is that the difference between conditional and unconditional log-likelihoods measures the evidence contributed by a design prompt beyond a sequence's general likelihood. For a prompt x and sequence y, PMI satisfies $r_{\mathrm{PMI}}(x, y) = \log p(y|x) - \log p(y) = \log p(x|y) - \log p(x)$. Thus, for a fixed prompt, a generative model can implicitly score compatibility with that prompt through a likelihood ratio, without training a separate reward model. This perspective connects mutual-information objectives in language generation (Li et al., 2016) with the conditional–unconditional contrast underlying classifier-free guidance (Ho & Salimans, 2022).

Building on this reward, we develop a label-free adaptation pipeline in which a reference PLM generates candidate sequences, scores their compatibility with the conditioning information, and supplies automatically constructed supervision for offline fine-tuning. We adopt binarized reward optimization (BRO) (Li et al., 2026) to reinforce favorable generations and discourage unfavorable ones using the resulting reward assignments. This procedure requires neither experimentally labeled preferences nor an external resources for reward computation.

We evaluate our proposed reward and pipeline with diverse settings covering retrospective prediction of experimentally measured binding affinity and folding stability (Tsuboyama et al., 2023; Shanker et al., 2024), reward diagnostics across sampling temperatures, and post-training for functional and structural conditioning. Across the evaluated tasks, PMI combined with BRO significantly enhances conditional generation fidelity over the base models. For ESM3, it increases structure-to-sequence generation metrics like SS3 accuracy from 0.719 to 0.799 and SASA Spearman correlation from 0.746 to 0.808, while also improving pLDDT and pTM.

In summary, our main contributions are:

- We propose PMI as a general reward principle for label-free adaptation, interpreting the conditional–unconditional likelihood ratio as an implicit measure of prompt compatibility.
- We instantiate this principle in a protein-model fine-tuning pipeline that uses the same reference model for generation and reward computation, enabling efficient self-improvement of PLMs and outperform existing LFT baselines.
- We provide a comprehensive empirical study spanning mutational-effect prediction, reward diagnostics, and controllable protein generation, demonstrating the broad applicability of PMI across model architectures, conditioning modalities, and sampling temperatures.

2 Related Work

2.1 Label-Free Fine-Tuning of Large Language Models

Post-training methods such as reinforcement learning from human feedback and direct preference optimization align large language models with desired behavior using preference supervision (Ouyang et al., 2022; Rafailov et al., 2023). Label-free alternatives instead construct learning signals without ground-truth answers for the adaptation examples. These signals can be grouped by their source (He et al., 2026): intrinsic approaches use the model's own predictions, including majority-vote rewards in TTRL (Zuo et al., 2025) and semantic-entropy minimization in EMPO (Zhang et al., 2025), whereas external approaches obtain feedback from an environment or another computational mechanism. Intrinsic rewards can improve performance by concentrating probability on initially favored answers, but may amplify errors or reduce diversity when confidence is misaligned with correctness (He et al., 2026). External verification can ground learning more directly; for example,

Absolute Zero uses a code executor to validate self-generated tasks and their solutions (Zhao et al., 2026). Such verification is effective in executable domains, but an analogous inexpensive verifier is often unavailable for biological function. This motivates rewards derived from pretrained models that capture task-relevant evidence beyond absolute confidence or agreement. Building on this direction, we study the difference between conditional and unconditional sequence likelihoods as a proxy for prompt compatibility, and use it to construct supervision for protein-model adaptation without task-specific labels.

2.2 Applications in Protein Language Models

PLM fine-tuning has gained increasing traction for generating proteins with desired properties that are underrepresented in the pretraining data. Nevertheless, many existing methods rely on explicit ground-truth supervision to train reward (Wang et al., 2025a) or predictive models (Xiong et al., 2026), compute steering vectors (Huang et al., 2025), or curate preference pairs (Widatalla et al., 2026). This dependence limits their applicability to biological tasks with scarce experimental labels. Inspired by LFT in LLM literature (Zuo et al., 2025; Zhao et al., 2026; He et al., 2026; Zhang et al., 2025), recent studies (Wang et al., 2025b; Li et al., 2026) have proposed various proxies for label-free fine-tuning of PLM, including intrinsic rewards such as entropy and perplexity to sharpen model confidence, and extrinsic rewards like semantic distance to incentivize generation consistency. However, the reported rewards are sensitive to sampling temperatures and tasks, and the reward selection requires ground truth labels, rendering the procedure not truly label-free. In contrast, our work proposes a genuinely label-free reward that is easy to implement, yet has been demonstrated to be amenable to a broad class of protein generative models and design tasks.

3 Method

3.1 Label-Free Fine-Tuning and Binarized Reward Optimization

Let x denote a design prompt, y a protein sequence, and $\pi_\theta(y|x)$ a conditional protein language model. Conventional adaptation proceeds through pretraining, supervised fine-tuning (SFT), and reward-based optimization (Ethayarajh et al., 2024; Li et al., 2026). During pretraining, the model learns sequence regularities from an unlabeled protein corpus using a generative objective, such as masked token prediction:

$$\mathcal{L}_{\text{pre}} = -\mathbb{E}_{y,m}\left[\sum_{i\in m}\log \pi_\theta(y_i|y_{\setminus m})\right], \tag{1}$$

where m indexes the masked positions. Supervised fine-tuning subsequently incorporates conditioning information, such as functional tags or structural constraints, through

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y),m}\left[\sum_{i\in m}\log \pi_\theta(y_i|y_{\setminus m}, x)\right]. \tag{2}$$

We denote the resulting conditional model by π_{ref}. Models that already incorporate these conditioning modalities during pretraining can serve directly as the reference policy.

Our objective is to further improve controllable generation without collecting ground-truth annotations or experimentally validated preferences for the adaptation data. Here, *label-free* refers specifically to this reward-based adaptation stage; the reference model may have previously undergone supervised training. We assume access to a fixed proxy reward $r(x, y)$, computed without ground-truth labels, that estimates sample fidelity, i.e., how well a generated sequence satisfies its prompt.

For a prompt distribution $p_{\mathcal{X}}$, adaptation is motivated by the KL-regularized reward objective

$$\max_\theta \ \mathcal{J}(\theta) = \mathbb{E}_{x\sim p_{\mathcal{X}}}\Big[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x,y)] - \beta D_{\text{KL}}\big(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big)\Big], \tag{3}$$

where $\beta > 0$ controls the penalty for departing from the reference policy. Over unrestricted conditional distributions, the optimal policy is

$$\pi^*(y|x) = \frac{\pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)}{Z(x)}, \tag{4}$$

$$Z(x) = \sum_y \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)\,. \tag{5}$$

Consequently,

$$\mathcal{J}(\theta) = \beta\mathbb{E}_{x\sim p_{\mathcal{X}}}[\log Z(x)] - \beta\mathbb{E}_{x\sim p_{\mathcal{X}}}\left[D_{\text{KL}}\big(\pi_\theta(\cdot|x)\,\|\,\pi^*(\cdot|x)\big)\right]. \tag{6}$$

Thus, maximizing the regularized reward corresponds to matching an exponentially reweighted reference policy.

In practice, proxy rewards may provide useful coarse distinctions while remaining unreliable in their numerical magnitudes. To reduce dependence on reward calibration, we adopt *binarized reward optimization* (BRO) from (Li et al., 2026), by constructing an offline dataset $\mathcal{D} = \{(x_i, y_i, b_i)\}_{i=1}^N$, where $b_i \in \{0, 1\}$ is a binary assignment derived from the proxy reward. Positive examples are sequences judged more compatible with their prompts, and negative examples are those judged less compatible. These assignments provide automatically generated supervision without requiring ground-truth annotations. Each example is optimized independently, so training does not require explicitly paired preferred and dispreferred responses.

BRO uses the log-likelihood ratio

$$s_\theta(x,y) = \log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \tag{7}$$

as a binary classification logit. Applying the sigmoid yields

$$p_\theta(x,y) = \sigma(s_\theta(x,y)) = \frac{\pi_\theta(y|x)}{\pi_\theta(y|x) + \pi_{\text{ref}}(y|x)}, \tag{8}$$

where $\sigma(t) = (1+\exp(-t))^{-1}$. This quantity measures the learned policy's relative likelihood for a sequence compared with the fixed reference. We fit it to the binary reward assignments by minimizing

$$\begin{aligned}\mathcal{L}_{\text{BRO}} &= \mathbb{E}_{(x,y,b)\sim\mathcal{D}}\left[D_{\text{KL}}\big(\text{Bern}(b)\,\|\,\text{Bern}(p_\theta(x,y))\big)\right] \\ &= -\mathbb{E}_{(x,y,b)\sim\mathcal{D}}\Big[b\log p_\theta(x,y) + (1-b)\log(1-p_\theta(x,y))\Big].\end{aligned} \tag{9}$$

Minimizing this loss encourages higher likelihood relative to the reference for positive examples and lower relative likelihood for negative examples. BRO therefore provides a tractable binary surrogate motivated by the regularized reward formulation. It discards reward magnitudes and avoids computing the sequence-level partition function, while retaining a direct mechanism for reinforcing favorable generations and suppressing unfavorable ones.

3.2 Pointwise Mutual Information

The effectiveness of LFT and BRO depends on the quality and generality of the proxy reward. We thereby propose pointwise mutual information (PMI) as a simple yet principled candidate:

$$r_{\text{PMI}}(x,y) \equiv \log\frac{p(x,y)}{p(x)p(y)} = \log p(y|x) - \log p(y). \tag{10}$$

For a given PLM π_θ, Eq. (10) translates to $r_{\text{PMI}}(x,y;\theta) = \log\pi_\theta(y|x) - \log\pi_\theta(y)$, which measures the discrepancy between the conditional and unconditional log probability of an outcome pair (x, y). Plugging r_{PMI} into the optimization objective $\mathcal{J}(\theta)$ in Eq. (3) yields

$$\begin{aligned}\mathcal{J}_{\text{PMI}}(\theta) &= \mathbb{E}_{x\sim p_{\mathcal{X}}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log\frac{\pi_\theta(y|x)}{\pi_\theta(y)}\right] - \beta D_{\text{KL}}\big(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big)\right] \\ &= I_{\pi_\theta}(X,Y) - \beta\mathbb{E}_{x\sim p_{\mathcal{X}}}\Big[D_{\text{KL}}\big(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big)\Big].\end{aligned} \tag{11}$$

where the first term is precisely the mutual information between X and Y induced by π_θ, rendering the label-free fine-tuning as a constrained optimization of mutual information $I(X, Y)$. Moreover, $r_{\text{PMI}}(x, y) = \log p(y|x) - \log p(y) \propto \log p(x|y)$ by Bayes' rule, which in our realization can be interpreted as an implicit classifier or reward model $\pi^i_\theta(x|y)$ by inverting the generative model $\pi_\theta(y|x)$ (Grandvalet & Bengio, 2004) [1]. Therefore, according to Eq. (4), the optimal policy of the mutual information maximization becomes

$$\pi^*(y|x) \propto \pi_{\text{ref}}(y|x)\left(\frac{\pi_\theta(y|x)}{\pi_\theta(y)}\right)^{\frac{1}{\beta}} \propto \pi_{\text{ref}}(y|x)\pi^i_\theta(x|y)^{\frac{1}{\beta}} \tag{12}$$

which effectively enhances the probability of data for which the classifier $\pi^i_\theta(x|y)$ assigns high likelihood to the correct label.

The formulation is reminiscent of classifier-free guidance (Ho & Salimans, 2022) in diffusion models, where the gradient of a classifier model $p_\phi(c|z)$ (c and z are label and latent respectively) is replaced by that of an implicit classifier $p^i(c|z) \propto p(z|c)/p(z)$ induced by a conditional model $p(z|c)$ and an unconditional model $p(z)$ to achieve label-free guidance. Despite the mathematical analogy, unlike diffusion models which estimates gradients in the intermediate stochastic process, we employ PMI as the optimization objective by computing the log likelihood of outcome sequence given input prompt, which is easy to implement and directly applicable to fine-tuning algorithms such as PPO (Schulman et al., 2017; Ouyang et al., 2022), DPO (Rafailov et al., 2023) and BRO (Li et al., 2026).

4 Experiments

Since our primary goal is to design a general label-free reward with broad applicability in protein engineering, the experiments are organized to progressively evaluate PMI in the following diverse scenarios. Section 4.1 shows that PMI enables zero-shot mutational effect prediction on protein binding affinity and thermodynamic folding stability benchmarks. Section 4.2 illustrates PMI's generality as a proxy for conditional generation fidelity across protein design tasks and sampling temperatures. Finally, we integrate PMI with BRO in Section 4.3 to demonstrate its effectiveness in steering frontier PLMs for improved sample fidelity without task-specific labels. Following (Ho & Salimans, 2022), we use a single PLM to parameterize both conditional and unconditional models, where for the unconditional probability $p(y) = \sum_x p(y|x)p(x)$ which is generally intractable for high-dimensional prompts, we simply input null tokens $\varnothing$ for generation, i.e. $p(y) = p(y|x = \varnothing)$. The conditions in our experiments cover amino acid, functional and single-chain/complex structural information for comprehensive evaluation.

4.1 Predicting Protein Mutational Landscapes

Deep mutational scanning (Fowler et al., 2010; Fowler & Fields, 2014) (DMS) is a high-throughput laboratory method to explore potential mutations with higher fitness. Recently, Shanker et al. (2024); Gordon et al. (2025) showed that the log likelihood of a structure-conditioned protein language model exhibits good agreement with antibody binding affinity in DMS, and can therefore guide evolution without task-specific training data or experimental labels. Motivated by the same label-free principle, we present a retrospective study of PMI for zero-shot prediction of protein mutational landscapes. To assess its generality, we consider two distinct settings: (1) antibody binding affinity prediction using the structure-conditioned PLM ESM-IF1, following (Shanker et al., 2024), and (2) thermodynamic folding stability prediction using the multimodal PLM ESM3 (Hayes et al., 2025), evaluated on measurements from (Tsuboyama et al., 2023). We compare PMI with conditional log-likelihood, as used by Shanker et al. (2024), and unconditional log-likelihood.

Figure 1(a) presents the spearman correlation between the metrics and binding affinity scores for three wild-type antibodies CR9114, CR6261 and g6.31, where ESM-IF1 is conditioned on the antigen-antibody complex structures. Figure 1(b) measures the same spearman ρ between the metrics and folding stability across 371 and 153 unique parent proteins for single mutants and double

[1] Since x is a prompt or condition, $p(x|y)$ can be understood as predicting a certain property (e.g., function, structure, description, etc.) of the generated protein sequence y.

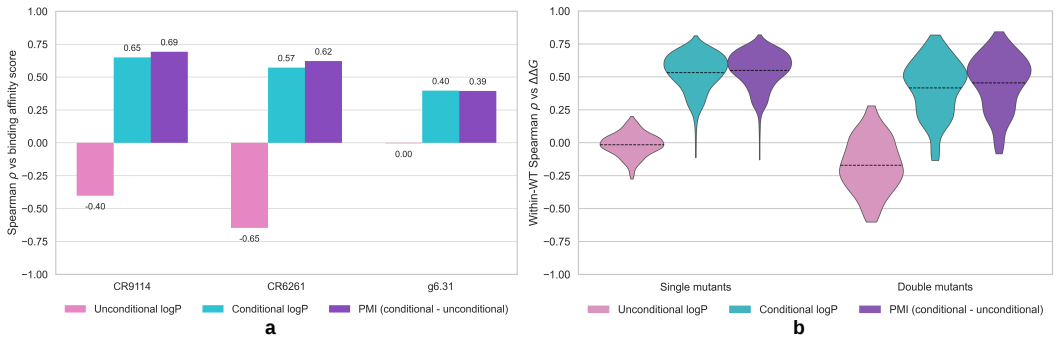


Figure 1: **Comparative analysis on deep mutational scanning (DMS) benchmarks**. PMI achieves consistently favorable correlations with experimentally measured (a) antibody binding score (scFv equilibrium dissociation constants (K_D)) across three antibodies; (b) thermodynamic folding stability ($\Delta\Delta G$) of single and double mutants of 371 parent proteins, the dashed lines denote the distribution mean.

mutants respectively, where ESM3 is conditioned on the single-chain structures. Both binding affinity and folding stability scores are measured experimentally via DMS, including 78,171 and 548,119 measurements in total. Remarkably, PMI exhibits stronger correlations with the measured properties than either conditional or unconditional logP on both datasets, despite antibody binding affinity and single-chain folding stability being distinct biophysical properties. For most cases, conditional logP correlates positively with the fitness measures whereas unconditional logP correlates negatively, which aligns with the intuition that high-fitness proteins are rare in unconditional pretraining data. This also explains how PMI works: it decreases the unconditional likelihood of the sample while increasing the conditional likelihood by maximizing their difference.

4.2 Reward Diagnostics in Protein Generative Design

Despite offering critical insight on the effectiveness of PMI, DMS presents only a narrowed scope for the generative design of functional proteins. In contrast, frontier PLMs like ProGen and ESM3 enable controllable de novo generation and design (Ferruz & Höcker, 2022) by following prompts that specify protein families, functions, or other high-level design objectives. We are therefore interested in whether PMI still provides meaningful signals for protein generative tasks with greater design space. To this end, we investigate two de novo Func2Seq design tasks: (1) Prompting variants of the autoregressive ProGen2 with a combination of family tag and partial amino acid sequence, following (Li et al., 2026), and (2) Prompting the multimodal masked language model (MLM) ESM3 with residue-level functional tokens, following (Hayes et al., 2025). We ensure the prompts are unseen for both models during pretraining and SFT, see details in. Since both tasks aim to generate novel proteins with the specified functional or family properties, we employ InterProScan (Paysan-Lafosse et al., 2023) to create a binary label as the ground truth in assessing whether the generated sequence recovers the desired keyword. In addition to conditional/unconditional log-likelihood, we also compare PMI with L1, a measure of semantic distance within a group of generated sequences with the same condition, introduced in (Li et al., 2026) as a competitive baseline[2].

Figure 2 illustrates the datasetwise (first row) and promptwise (second row) AUROC evaluated across three PLMs and five sampling temperatures. For autoregressive ProGen2 variants, both conditional and unconditional logP transition from negative AUROC at low temperature to positive AUROC at high temperature, making them sensitive to the sampling scheme and therefore cannot be used as a universal proxy reward. PMI as their difference, however, exhibits a consistently positive correlation with the ground truth across all temperatures and models. Moreover, PMI outperforms L1 in most settings without access to external information. These results highlight PMI's potential as a simple yet effective proxy for generation fidelity, which is generally applicable across different PLM model architectures, generative design tasks and sampling schemes.

[2]Note that unlike intrinsic metrics such as PMI or conditional/unconditional logP, L1 requires an external PLM like ESM C (ESM Team, 2024) to generate sequence embeddings for computing semantic distance.

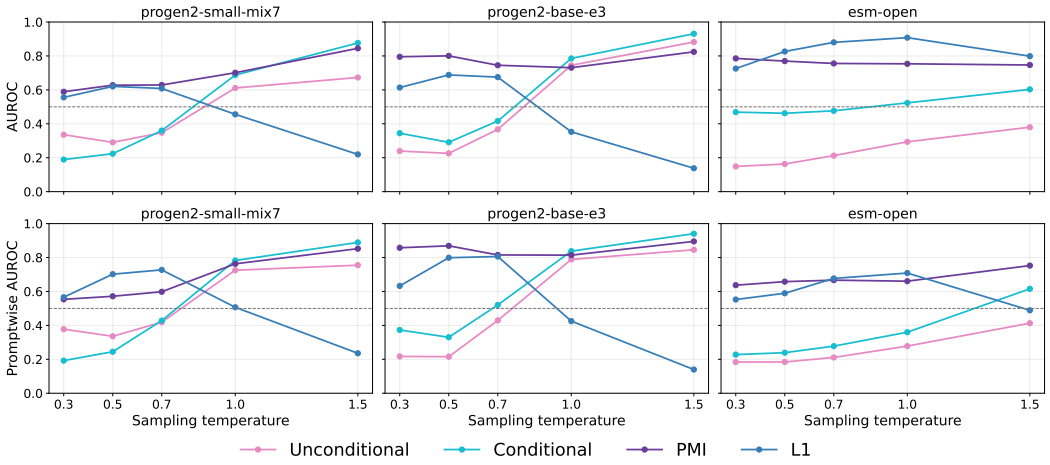


Figure 2: **Reward analysis on Func2Seq design tasks across three protein language models.** (1) `progen2-small-mix7`: a 151M-parameter autoregressive ProGen2 model fine-tuned by Hrbáň & Hoksza (2024) using supervised fine-tuning (SFT) on sequences from seven Pfam families. (2) `progen2-medium-mix7`: a 764M-parameter ProGen2 model fine-tuned by us under the same setup to assess the effect of model scale. (3) `esm-open`: the publicly released 1.4B-parameter ESM3 model, included to assess applicability to multimodal masked language models. The top and bottom rows report dataset-level AUROC and mean prompt-level AUROC, respectively, across sampling temperatures.

4.3 Fine-Tuning Evaluation

By demonstrating PMI as a broadly applicable proxy for PLM controllability (i.e., conditional generation fidelity), we now apply it as a label-free reward for fine-tuning PLMs on protein generative design tasks. Our fine-tuning setup includes (1) Function-to-sequence (Func2Seq) generation, which uses the same prompts and base models as reward diagnostics in Section 4.2, and (2) Structure-to-sequence (Struct2Seq) generation, which prompts ESM3 with secondary structure (SS8) and solvent accessible surface area (SASA) tokens as conditions. Following (Hayes et al., 2025), we define three consistency metrics for each task: (i) Keyword recovery (Func2Seq), the fraction of prompt key words recovered by InterProScan; (ii) SS3 accuracy (SS8), the fraction of residues where three-class secondary structure between the prompt and generations match; (iii) Spearman ρ (SASA), the correlation between the SASA prompt and the corresponding region of the generation. We also assess conventional metrics for protein generation including diversity, novelty (against UniRef90 (Suzek et al., 2015)), pLDDT and pTM. Following (Li et al., 2026), for each PLM, we sample an on-policy training set, compute PMI reward and perform fine-tuning via the offline BRO algorithm introduced in Section 3.1, all with the same base model. For detailed description of our data generation, fine-tuning and evaluation schemes, see Section A.

For comparison, we consider two representative baselines: (1) Best-of-n sampling, as a form of test-time compute scaling (Didi et al., 2026): given an increasing compute budget, we grow the number of generated samples N and select top N/n samples ranked by an evaluation metric (e.g., PMI score). It is also referred to as post hoc filtering in some studies (Xiong et al., 2026); (2) L1 reward optimized with BRO, which is reported as the best-performing strategy in (Li et al., 2026). Note that computing semantic L1 distance requires an external model to generate sequence embeddings, which leverages more information than PMI does. For reference, we also include a supervised baseline obtained by fine-tuning the corresponding base model with BRO using ground-truth labels[3].

[3]Note that this is not a strict performance upper-bound, but only a reference. For binary ground-truth labels on Func2Seq tasks, the positive and negative samples are randomly chosen and therefore the data distribution might differ substantially from datasets filtered by L1 or PMI scores. For Struct2Seq tasks with continuous ground-truth labels, due to limited computational resources, we directly compute the ground truth labels on the filtered PMI training set, which can be significantly imbalanced (positive samples take up only 27.5% for the SS8 task). This explains the performance collapse in Tab. 2.

Table 1: **Performance on Func2Seq generative tasks.** We report performance metrics averaged over five sampling temperatures $T \in \{0.3, 0.5, 0.7, 1.0, 1.5\}$. For each base model, the best method except Supervised is **bolded** and the second best is underlined. The diversity results of Best-of-n baselines are omitted due to unfair comparison by sequence clustering. More details about the evaluation protocol can be found in Section A.

Method	Keyword Recovery ↑	Diversity ↑	Novelty ↓	pLDDT ↑	pTM ↑
Progen2-small-mix7 (151M)					
Base	0.281	**0.673**	0.672	0.533	0.371
L1 Best-of-64	0.403	-	0.653	0.563	0.417
L1	**0.433**	0.577	0.716	**0.594**	0.446
PMI Best-of-64	0.347	-	0.652	0.580	0.442
PMI	0.430	0.619	**0.646**	0.582	**0.466**
Supervised	0.488	0.683	0.601	0.572	0.438
Progen2-medium-mix7 (764M)					
Base	0.385	**0.623**	0.778	0.522	0.374
L1 Best-of-64	0.524	-	0.757	0.543	0.416
L1	0.613	0.508	0.779	0.607	**0.491**
PMI Best-of-64	**0.658**	-	**0.605**	0.570	0.455
PMI	0.585	0.538	0.762	**0.621**	0.477
Supervised	0.603	0.508	0.749	0.621	0.454
ESM3 (1.4B)					
Base	0.197	0.548	0.493	0.590	0.421
L1 Best-of-64	0.224	-	0.488	0.585	0.414
L1	0.255	0.578	0.494	0.619	0.484
PMI Best-of-64	0.195	-	**0.475**	0.657	0.473
PMI	**0.277**	**0.614**	0.478	**0.679**	**0.548**
Supervised	0.260	0.691	0.478	0.636	0.514

Tab. 1 reports performance on Func2Seq tasks for two autoregressive ProGen2 variants and the multimodal masked language model ESM3. Fine-tuning with PMI substantially improves all metrics except diversity over the corresponding base models across model sizes and architectures. The slight decrease in diversity for ProGen2 variants is expected due to the trade-off between sample quality (controllability) and diversity (Ho & Salimans, 2022). Compared to the alternative L1 reward, which by construction discourages sample variety, PMI achieves competitive keyword recovery and consistently better diversity, novelty and pLDDT/pTM. We speculate that compared to self-consistency metrics which can be easily misspecified (e.g., the generated sequence can all be low-quality yet still have a good agreement with each other), PMI instills more regularities to PLMs by reinforcing the inductive biases learned during pretraining and SFT stage and therefore enables a more favorable controllability-diversity tradeoff. Additionally, the best-of-n sampling methods are also effective with both reward functions, but not comparable to their fine-tuned counterparts. Tabs. 2 and 3 further extend the evaluation to Struct2Seq tasks, using ESM3's multimodal capabilities to generate protein sequences that satisfy structural design constraints. Here, fine-tuning with PMI shows a more pronounced advantage over the other baselines in residue-level accuracy or Spearman correlation, sequence novelty, and pLDDT/pTM. Overall, these findings highlight the advantage and generality of our proposed label-free fine-tuning paradigm: integrating PMI with fine-tuning algorithms like BRO reliably improves PLM generation quality without requiring any task-specific supervision.

5 Conclusion

We have introduced PMI as a model-derived reward for label-free fine-tuning of protein language models. By contrasting conditional and null-conditioned sequence scores, the approach uses the reference model itself to assess prompt compatibility and construct supervision for BRO, without task-specific labels or a separate reward model. Evaluations spanning mutational-effect prediction, reward diagnostics, and functional and structural generation support the usefulness of this reward

Table 2: **Performance on Struct2Seq generative tasks conditioning on the secondary structure (SS8) track of ESM3.** Metrics are averaged across five sampling temperatures. Excluding Supervised, the best results are in **bold** and the second best are underlined. Novelty values other than PMI are synthetic placeholders; their rankings are illustrative.

Method	Accuracy ↑	Diversity ↑	Novelty ↓	pLDDT ↑	pTM ↑
Base	0.719	**0.946**	0.337	0.405	0.237
L1 Best-of-64	0.728	-	0.340	0.404	0.235
L1	0.735	0.916	0.338	0.436	0.234
PMI Best-of-64	0.750	-	0.330	0.422	0.245
PMI	**0.799**	0.768	**0.327**	**0.490**	**0.264**
Supervised	0.594	0.487	0.342	0.582	0.330

Table 3: **Performance on Struct2Seq generative tasks conditioning on the solvent accessible surface area (SASA) track of ESM3.** Metrics are averaged across five sampling temperatures. Excluding Supervised, the best results are in **bold** and the second best are underlined. Novelty values other than Base and PMI are synthetic placeholders; their rankings are illustrative.

Method	Spearman ↑	Diversity ↑	Novelty ↓	pLDDT ↑	pTM ↑
Base	0.746	**0.628**	**0.398**	0.598	0.533
L1 Best-of-64	0.752	-	0.411	0.598	0.534
L1	0.754	0.611	0.402	0.604	0.537
PMI Best-of-64	0.775	-	0.405	0.610	0.546
PMI	**0.808**	0.515	0.404	**0.617**	**0.546**
Supervised	0.807	0.493	0.404	0.602	0.532

across autoregressive and multimodal architectures. PMI+BRO improves controllability over the base models and offers competitive performance against alternative rewards and best-of-n selection, subject to a quality-diversity trade-off. These findings support conditional–unconditional contrasts as a practical foundation for self-supervised adaptation of protein generators. Future work may extend the pipeline to diffusion and flow-matching models, where the intractable log-likelihood currently impedes direct application; and test whether the observed improvements translate into experimentally validated protein properties.

Broader Impact. This work investigates label-free adaptation of protein language models to improve conditional protein generation. By reducing the need for task-specific labels during adaptation, this approach could make computational protein design more accessible and support research in biotechnology and therapeutic discovery. However, model-derived rewards can inherit biases and errors from pretrained models, and improved computational scores do not necessarily establish biological function, safety, or therapeutic efficacy. More capable protein generation also presents potential dual-use risks. Downstream applications therefore require experimental validation and application-specific assessment of biosafety and biosecurity.

Ethics Statement

Our study consists of computational protein generation and retrospective evaluation using existing protein datasets and pretrained models. It does not report new human-subject studies, animal experiments, or experimental testing of generated proteins. The proposed PMI reward measures model-derived compatibility with a conditioning prompt; it does not assess toxicity, immunogenicity, or other safety properties. Consequently, our results should not be interpreted as establishing the safety of generated sequences. The label-free claim applies specifically to reward construction and adaptation, while the reference models and evaluation benchmarks may incorporate supervised information.

Reproducibility Statement

Section 3 defines the PMI reward and the adopted binarized reward optimization objective, and Sections 4.1–4.3 describe the benchmarks, reward diagnostics, and fine-tuning evaluations. Section A documents implementation details including dataset generation, reward computation, optimization and evaluation settings. A copy of documented source code has been provided as Supplementary Material in the submission.

AI Use Statement

We used pretrained protein language models to generate synthetic adaptation sequences and compute model-derived rewards, as described in Sections 3 and 4. We also used an AI assistant for polishing writing, literature search, feedback on mathematical formulations and experimental methodology, and assistance with plotting and dataset inspection. The authors take responsibility for the final content of the paper, including all AI-assisted material.

References

Nathaniel Blalock, Srinath Seshadri, Kensuke Nakamura, Agrim Babbar, Sarah A Fahlberg, Ameya Kulkarni, and Philip A Romero. Functional alignment of protein language models via reinforcement learning. *Nature Communications*, 2026.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations*, volume 2026, pp. 61056–61109, 2026.

ESM Team. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning, 2024. URL https://evolutionaryscale.ai/blog/esm-cambrian.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, pp. 12634–12651. PMLR, 2024.

Noelia Ferruz and Birte Höcker. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6):521–532, 2022.

Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature methods*, 11(8):801–807, 2014.

Douglas M Fowler, Carlos L Araya, Sarel J Fleishman, Elizabeth H Kellogg, Jason J Stephany, David Baker, and Stanley Fields. High-resolution mapping of protein sequence-function relationships. *Nature methods*, 7(9):741–746, 2010.

Cade Gordon, Amy Lu, and Pieter Abbeel. Protein language model fitness is a matter of preference. In *International Conference on Learning Representations*, volume 2025, pp. 39759–39775, 2025.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

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.

Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, Xiusi Chen, Youbang Sun, Xingtai Lv, Xuekai Zhu, Li Sheng, Ran Li, Huan ang Gao, Yuchen Zhang, Lifan Yuan, Bowen Zhou, et al. How far can unsupervised RLVR scale LLM training? In *The Thirteenth International Conference on Learning Representations*, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Hugo Hrbáň and David Hoksza. Protein family sequence generation through progen2 fine-tuning. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 7058–7060. IEEE, 2024.

Long-Kai Huang, Rongyi Zhu, Bing He, and Jianhua Yao. Steering protein language models. In *International Conference on Machine Learning*, pp. 26247–26260. PMLR, 2025.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 110–119, 2016.

Lanqing Li, Shentong Mo, Yang Yu, and Pheng-Ann Heng. Be your own teacher: Steering protein language models via unsupervised reward optimization. *arXiv preprint arXiv:2606.18961*, 2026.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, et al. Interpro in 2022. *Nucleic acids research*, 51(D1):D418–D427, 2023.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Varun R Shanker, Theodora UJ Bruun, Brian L Hie, and Peter S Kim. Unsupervised evolution of protein and antibody complexes with a structure-informed language model. *Science*, 385(6704): 46–53, 2024.

Filippo Stocco, Maria Artigues-Lleixa, Andrea Hunklinger, Talal Widatalla, Marc Guell, and Noelia Ferruz. Guiding generative protein language models with reinforcement learning, 2024. URL https://arxiv.org/abs/2412.12979.

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.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444, 2023.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen Wang, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. In *International Conference on Learning Representations*, volume 2025, pp. 47871–47899, 2025a.

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*, 2025b.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Talal Widatalla, Ashir A Borah, Samuel H King, Claudia L Driscoll, Rafael Rafailov, and Brian L Hie. Aligning protein-generative models to experimental fitness with proteindpo. *Nature Methods*, pp. 1–9, 2026.

Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M Oltrogge, David F Savage, and Jennifer Listgarten. Property guidance for protein sequence generative models with proteinguide. *Nature Biotechnology*, pp. 1–13, 2026.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. In *Thirty-ninth Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=k8Mim6RI5O. Spotlight Presentation.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2026.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *Thirty-ninth Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VuVhgEiu20. Poster Presentation.

A Implementation Details

We describe the data, reward computation, optimization, and evaluation protocols for the experiments in Sections 4.1–4.3. The ProGen2 experiments follow the shared experimental settings of Li et al. (2026), while the ESM3 experiments extend the evaluation to function-, secondary-structure-, and solvent-accessibility- conditioned fine-tuning. Unless stated otherwise, all methods for a given base model use the same evaluation prompts and decoding configuration. The label-free setting concerns construction of the PMI and L1 adaptation rewards: experimental measurements and task-specific evaluation labels are not used to assign their training labels.

A.1 Models and conditioning interfaces

We evaluate two autoregressive ProGen2 models with 151M and 764M parameters. Their identifiers are `progen2-small-mix7` and `progen2-medium-mix7`, respectively. The small model is the seven-family checkpoint from Hrbáň and Hoksza (2024); the medium model is supervised fine-tuned on the same family dataset. Both use a family-specific control token followed by an amino-acid prefix. We also use the 1.4B-parameter ESM3-open model, `esm3-sm-open-v1`, whose function, secondary-structure, and SASA tracks provide the corresponding conditioning information. ESM-IF1 is used for the antibody mutational-effect experiments. The L1 baseline uses ESMC-600M as a separate sequence representation model.

A.2 Prompt construction and data partitions

ProGen2 Func2Seq. The seven target families are PF00002, PF00042, PF00125, PF00127, PF00257, PF00262, and PF03668. To construct compositional out-of-distribution prompts, we concatenate each family token with the first ten amino acids of a protein from the next family in the cyclic order shown in Table 4. Thus, the target-family control tag and the sequence-prefix family differ. This tests generalization to their combination; it does not by itself establish that the component sequences were absent from pretraining.

Table 4: Shared ProGen2 prompt construction. Each prompt contains a target-family token followed by a ten-residue prefix from the indicated source family.

Target family	Family description	Prefix-source family
PF00002	G protein-coupled receptors	PF00042
PF00042	Globins	PF00125
PF00125	Core histones	PF00127
PF00127	Copper-binding proteins	PF00257
PF00257	Dehydrins	PF00262
PF00262	Calreticulins	PF03668
PF03668	P-loop ATPase	PF00002

The adaptation prompt pool, Pfam1400, contains 1,400 prompts, with 200 prompts per target family. The held-out evaluation pool, Pfam700, contains 700 prompts, with 100 prompts per target family. We use an 80:20 training–validation split for adaptation, following the shared protocol; Pfam700 is a separate evaluation pool.

ESM3 Func2Seq. The function prompts are derived from InterPro annotations of *Dracunculus medinensis* proteins in the DRAME subset of AlphaFoldDB, following the dataset construction in Li et al. (2026). Each prompt specifies the desired functional annotations and their residue spans.

ESM3 SS8 and SASA conditioning. We evaluate secondary-structure and solvent-accessibility conditioning as separate tasks. For each source protein, the corresponding per-residue prompt is provided through the ESM3 secondary-structure or SASA track. This differs from the coordinate-conditioned inverse-folding diagnostic in the previous paper. The prompt tracks used for scoring are the tracks saved during generation, including any masked positions. In the structural generation implementation, PyDSSP assigns three-state secondary structure, which is mapped to H/E/C tokens in the ESM3 secondary-structure interface. Thus, SS8 denotes the conditioning interface; these prompts

contain three-state assignments. SASA prompts use continuous per-residue values computed with the ESM `ProteinChain.sasa()` routine. Each sequence has the length specified by its source prompt.

A.3 Candidate generation

We use five sampling temperatures, $\mathcal{T} = \{0.3, 0.5, 0.7, 1.0, 1.5\}$, and generate 64 candidates per prompt and temperature in the shared sampling protocol. A candidate group is a fixed base model, prompt, and temperature. ProGen2 uses autoregressive top-k decoding with $k = 15$. Consequently, the complete ProGen2 candidate pools contain $1{,}400 \times 5 \times 64 = 448{,}000$ adaptation candidates and $700 \times 5 \times 64 = 224{,}000$ evaluation candidates per base model, before any filtering or subsampling.

For the ESM3 one-pass SS8/SASA configuration, all amino-acid positions are initially masked. A conditional forward pass produces position-wise logits, and all amino acids are sampled simultaneously. This samples positions independently conditional on the provided tracks and length; it does not condition later positions on previously sampled amino acids. The function-generation implementation instead uses iterative unmasking. Its default schedule has $\max(1, \lfloor L/8 \rfloor)$ steps for a sequence of length L, with top-$p = 1$ and the ESM3 temperature-annealing option enabled. Generation and subsequent full-mask reward scoring are separate operations.

A.4 Reward computation and binary assignments

Autoregressive log scores and PMI. For an autoregressive model, the conditional log score is

$$\ell_{\mathrm{AR}}(y \mid x) = \sum_{i \in \mathcal{I}(y)} \log \pi_{\mathrm{ref}}(y_i \mid y_{<i}, x), \tag{13}$$

where $\mathcal{I}(y)$ denotes the scored output positions. In the default ProGen2 scoring configuration, the supplied ten-residue prefix is removed from the sequence to obtain the completion. The conditional score includes the family token and residue prefix as context; the null score uses the sequence start token alone as context for the same completion. Prompt tokens do not contribute to the completion score. The scorer supports summed and mean token log probabilities, with summed scores as its default. The PMI reward is the difference between conditional and null-conditioned scores, computed with the same frozen reference model:

$$r_{\mathrm{PMI}}(x, y) = \ell(y \mid x) - \ell(y \mid \varnothing). \tag{14}$$

Full-mask ESM3 scoring. For full-mask evaluation, we use

$$\ell_{\mathrm{FM}}(y \mid x) = \sum_{i=1}^{L} \log \pi_{\mathrm{ref},i}(y_i \mid \mathrm{MASK}^L, x). \tag{15}$$

The conditional pass supplies the function/residue-annotation, secondary- structure, or SASA track for the relevant task. The null-conditioned pass removes that conditioning using the corresponding model interface. BOS, EOS, and padding positions are excluded. We record the summed contrast and its per-residue counterpart, r_{PMI}/L. Both forward passes keep all output-sequence positions masked, so their probability distributions can be reused across candidates and sampling temperatures sharing an identical prompt and length. Scoring uses model probabilities without generation-temperature scaling or sampling-time top-p truncation. The scoring pipeline writes both summed and mean scores. Within a fixed prompt, all ESM3 candidates have the same length, so these two reductions yield identical PMI rankings.

Equation equation 15 is a full-mask sequence score, not the likelihood of an iterative decoding trajectory. Likewise, $\ell(y \mid \varnothing)$ is an operational null-conditioned baseline rather than an explicitly evaluated marginal over the empirical prompt distribution.

ESMC L1 reward. Let $e(y)$ be the mean of the final ESMC-600M residue embeddings, excluding special tokens and padding. For a group of n candidates, the L1 reward is

$$r_{\mathrm{L1}}(x, y_i) = -\frac{1}{n-1} \sum_{j \neq i} \|e(y_i) - e(y_j)\|_1. \tag{16}$$

Thus, smaller mean distance gives a higher reward. Repeated sequences remain separate samples even when their embeddings are computed only once. Distances are computed within each prompt–temperature group. Including the zero self-distance and dividing by n instead changes the scale by $(n-1)/n$ but leaves the within-group ranking unchanged.

Binary reward assignments. Within each prompt–temperature group, we retain the four candidates with the highest rewards and the four with the lowest, assigning labels $b = 1$ and $b = 0$, respectively, following the shared ProGen2 protocol. This produces $1{,}400 \times 5 \times 8 = 56{,}000$ labeled adaptation examples before the training–validation split. Applying the same selection to Pfam700 yields 28,000 labeled diagnostic examples; these are distinct from evaluation on the full candidate pool. Binary labels determine the BRO objective; the numerical magnitudes of the original rewards are discarded after selection.

A.5 BRO OPTIMIZATION AND COMPARISON METHODS

We initialize the trainable policy from the reference checkpoint and keep the reference policy fixed. Each selected example is optimized independently with the BRO binary cross-entropy objective in Section 3.1; no preferred/dispreferred pairs are required. Candidate generation and reward labeling are performed offline before optimization.

Table 5: ProGen2 BRO optimization settings.

Setting	ProGen2-small (151M)	ProGen2-medium (764M)
Optimizer	AdamW	AdamW
Learning rate	5×10^{-7}	5×10^{-7}
Epochs	1	1
Batch size	32	8
Training–validation ratio	80:20	80:20
Training hardware	One NVIDIA A40	One NVIDIA A40

ESM3 optimization. The ESM3 trainer implements BRO as two-class cross-entropy with logits $[\beta\ell_{\text{ref}}, \beta\ell_\theta]$, which is equivalent to binary cross-entropy with logit $\beta(\ell_\theta - \ell_{\text{ref}})$. BRO is selected with `-algo_type BRO`. The reference scores are computed with the frozen reference model, and both models use the same scoring mode and reduction. The default BRO reduction is the mean log probability over scored residues, with $\beta = 1$. Thus, the implemented ESM3 logit uses a length-normalized score difference.

Table 6 gives the task-specific configurations in the supplied training launchers. The shared trainer defaults are AdamW with learning rate 5×10^{-7}, one epoch, zero weight decay, gradient-norm clipping at 1, an 80:20 training–validation split, and seed 42. The learning rate increases linearly during the first 150 optimizer steps and then follows cosine decay. Non-sequence output heads are frozen because the objective consumes only sequence logits. The function configuration places LoRA adapters in the attention QKV and output projections, with rank 8, scale parameter 16, and dropout 0.05; the remaining base-model parameters are frozen.

Table 6: ESM3 configurations supplied with the implementation. Batch size counts prompt groups; flattened candidates are processed in forward microbatches.

Setting	Func2Seq	SS8 / SASA
Parameter updates	LoRA adapters	Backbone and sequence head
Per-device batch size	1	1
Gradient accumulation steps	8	1
Forward microbatch size	1	1
Maximum encoded length	512	514
Training score mode	Iterative unmasking	Full mask

The function training configuration scores tokens in eight randomly ordered teacher-forced reveal chunks. The structural configurations explicitly select full-mask scoring. These training score modes

are distinct from the full-mask PMI evaluator described above: the reward ranks the offline candidates, whereas the policy and reference scores enter the optimization loss.

Best-of-n selection. We score candidates from the base model with PMI or the L1 reward and retain the highest-ranked outputs independently within each prompt–temperature group. For a pool of M candidates, best-of-n retains M/n candidates when M is divisible by n. In particular, selecting one candidate from a pool of 64 implements best-of-64.

Supervised baseline. The supervised baseline uses task-specific evaluation labels in place of proxy reward assignments. For Func2Seq, positives and negatives are sampled using the binary recovery labels. For SS8/SASA, the current study assigns supervised labels within the PMI-filtered candidate subset, so its data distribution need not match the proxy-reward datasets. As noted in the main text, it is a supervised reference rather than a strict upper bound. In particular, positives constitute 27.5% of the SS8 supervised training subset, so this supervised comparison is affected by class imbalance.

A.6 Mutational-effect prediction

For antibody binding, we evaluate ESM-IF1 on CR9114, CR6261, and g6.31, conditioning on antigen–antibody complex structures and following the experimental data setting of Shanker et al. (2024). We compare conditional log scores, null-conditioned log scores, and their PMI contrast using Spearman correlation with the measured affinity outcomes.

For folding stability, we use the measurements of Tsuboyama et al. (2023) with structure-conditioned ESM3. The main-text evaluation contains 78,171 single-mutant measurements from 371 parent proteins and 548,119 double-mutant measurements from 153 parent proteins. Correlations are evaluated separately for the two mutation classes.

A.7 Evaluation metrics

Functional consistency. We annotate generated sequences with InterProScan. For a single Pfam target, the label is one if the requested family is recovered and zero otherwise. For an ESM3 prompt with a set $K(x)$ of requested keywords, the recovery score is

$$\text{Recovery}(x, y) = \frac{|K(x) \cap A(y)|}{|K(x)|}, \tag{17}$$

where $A(y)$ is the set of recovered annotations.

Structural consistency and confidence. Generated sequences are folded with ESMFold-v1. For secondary-structure conditioning, SS3 accuracy is the fraction of evaluated residues whose three-state assignments agree with the prompt. For SASA conditioning, we report Spearman correlation between prompted accessibility values and accessibility computed from the corresponding residues of the predicted structure. Secondary-structure assignments are obtained with PyDSSP. The three-state mapping merges H/G/I into helix, E/B into strand, and the remaining coil/turn/bend assignments into coil. Assignments use residues with complete backbone coordinates. SASA is computed with the same `ProteinChain.sasa()` routine used to construct prompts. The evaluators match source and generated residues by sequence position by default, and can restrict comparison to the prompted span. For SASA, only matched finite values contribute; fewer than two matched values or a constant profile yields an undefined correlation. Undefined values are excluded from numerical summaries, while missing predictions and evaluation failures are recorded separately.

We report residue-averaged pLDDT and sequence-level pTM from the same predictions. Both are displayed on a 0–1 scale; pLDDT values originally expressed on a 0–100 scale are divided by 100. Sequence-level confidence scores are averaged over evaluated sequences.

Diversity. Sequence diversity is the number of sequence clusters divided by the number of evaluated samples. Repeated sequences remain in the denominator. The implementation clusters generated sequences with MMseqs2 `easy-linclust`; its default configuration is `-min-seq-id 0.1 -c 0.7 -cov-mode 1`. It produces a global cluster fraction for each temperature-specific FASTA file and optionally clusters each prompt independently. For the latter, it reports an equally weighted mean

of prompt-level cluster fractions. The default sample budget is 64 sequences per prompt; oversized groups are subsampled without replacement with seed 0 before either clustering procedure. Best-of-n diversity is omitted from the main comparisons because its smaller retained output budget changes the clustering statistic.

Sequence novelty. We assess sequence novelty using the mean maximum sequence identity among generated samples with qualifying MMseqs2 hits against a fixed release of UniRef90 representatives. A qualifying hit has E-value at most 10^{-3} and covers at least 80% of the query sequence. Let $H(y)$ denote the qualifying hits returned for sample y, and let $\mathcal{S}_{\text{hit}}$ contain the searched samples with at least one such hit. We report

$$\text{MeanMaxIdentity} = \frac{1}{|\mathcal{S}_{\text{hit}}|} \sum_{y \in \mathcal{S}_{\text{hit}}} \max_{h \in H(y)} \text{Identity}(y, h), \tag{18}$$

where identity is expressed as a fraction in $[0, 1]$. Lower values indicate greater sequence novelty among samples with detected reference matches. The maximum is taken over returned qualifying alignments, so it is an approximate reference-similarity measure under the specified search settings. Samples without qualifying hits are excluded from this mean, rather than assigned identity zero. The metric is undefined if no samples have a hit. The metric is reported as mean maximum identity (↓). The evaluator exports identities as percentages; these are divided by 100 when presented on a 0–1 scale.

The search uses the following configuration, with paths omitted:

```
mmseqs easy-search queries.fasta uniref90 hits.tsv tmp \
  --search-type 1 --alignment-mode 3 --seq-id-mode 0 \
  -s 7.5 -e 0.001 -c 0.8 --cov-mode 2 \
  --max-seqs 1000 --threads 16 --split-memory-limit 16G \
  --format-output query,target,pident,qcov,tcov,evalue,bits
```

Identical query sequences are searched once and their results are mapped back to all original samples. Under the default preprocessing policy, sequences containing nonstandard residues are excluded and counted separately; they are not treated as no-hit samples. If evaluation is restricted to the first N records per prompt–temperature group within each FASTA file, selection is performed in file order before residue filtering and deduplication. This subsampling operates independently for each input file and is distinct from reward-ranked best-of-n selection. A combined query database allows sequences from all input FASTA files to share one reference search, after which results are restored to their original files and prompt–temperature groups.

Reward diagnostics and aggregation. Reward quality is measured by dataset-level AUROC and mean prompt-level AUROC at each temperature, as in Figure 2. Higher PMI, conditional and unconditional log scores, and negative L1 distance are oriented toward positive labels.

The main tables average results across the five temperatures.

A.8 Software, compute, and reproducibility

The implementation separates candidate generation, score computation, offline BRO optimization, and evaluation. Models and optimization are implemented in Python and PyTorch. ESM interfaces provide ESM3 conditioning and ESMC embeddings; InterProScan provides functional annotations; ESMFold-v1, PyDSSP, and the ESM protein-chain utilities provide structural evaluation; and MMseqs2 provides clustering and reference-sequence search.

Scores retain sample and prompt identifiers and sampling temperatures. Duplicate-sequence embeddings and full-mask distributions for identical prompts and lengths are cached while preserving sample multiplicity. Novelty outputs retain the sample manifest, per-sample matches, grouped summaries, and search configuration.