Fine-tuning protein Language Models by ranking protein fitness

Minji Lee **Kyungmin Lee** **Jinwoo Shin**
Korea Advanced Institute of Science and Technology (KAIST)
{haewon_lee, kyungmnlee, jinwoos}@kaist.ac.kr

Abstract

The self-supervised protein language models (pLMs) have demonstrated significant potential in predicting the impact of mutations on protein function and fitness, which is crucial for protein design. There are approaches to further condition pLM to language or multiple sequence alignment (MSA) to produce a protein of a specific family or function. However, most of those conditioning is too coarse-grained to express the function, and still exhibit a weak correlation to fitness and struggle to generate fit variants. To address this challenge, we propose a fine-tuning framework for pLM to align it to a specific fitness by ranking the mutants. We show that constructing the ranked pairs is crucial in fine-tuning pLMs, where we provide a simple yet effective method to improve fitness prediction across various datasets. Through experiments on ProteinGym, our method shows substantial improvements in the fitness prediction tasks even using less than 200 labeled data. Furthermore, we demonstrate that our approach excels in fitness optimization tasks.[1]

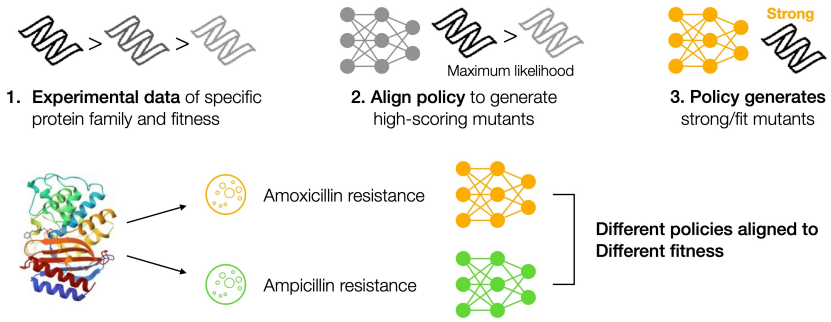


Figure 1: **Overview**. Protein language model aligned to specific fitness enable accurate fitness prediction and generation of fit mutants.

1 Introduction

The self-supervised protein language models (pLMs), which are trained by predicting the amino acids given sequence context, have shown great promise in predicting protein structure [10] and function [1, 7, 4, 17]. In particular, the pLMs excel in capturing the relationship between sequences and functions, showing state-of-the-art performance in fitness prediction [14, 11, 13, 1] without using any labeled data. However, those pLMs are not aligned with specific fitness of interest, and not the zero-shot framework, nor the existing conditional pLM approaches [21, 13] can encompass such conditions. For instance, in the ProteinGym [14] benchmark, we are interested in the resistance of Beta-lactamase TEM (Uniprot: P62593) variants to three different antibiotics of specific concentrations, which

[1] Our code is available in https://github.com/haewonc/align-plm.

NeurIPS 2023 Generative AI and Biology Workshop.

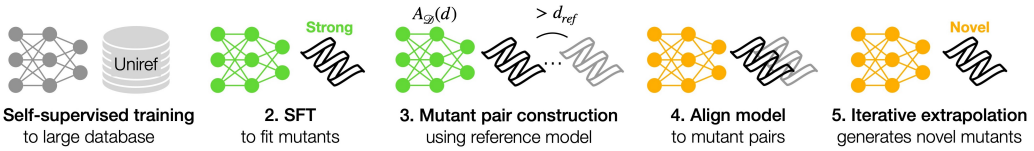


Figure 2: **Proposed framework.** Self-supervised training followed by fine-tuning to fit mutants and alignment to curated pairs allow generation of novel and fit mutants.

cannot be encoded in existing approaches. Moreover, zero-shot fitness prediction methods show a weak correlation with true fitness even in the most accurate cases, which limits their usage in fitness optimization (*e.g.,* see Table 4).

To tackle these limitations, we propose to fine-tune the pLM by ranking the pair of mutants using a maximum likelihood objective. Here, we do not explicitly provide the condition as in language [21] or a prompt [13], but allow the pLM to infer and align to the condition from the ranked data. This ranking approach is particularly useful for protein and therapeutics design, which involve mutagenesis and screening. Since the ranked mutant data is a common product of the protein design process, a method that can learn from such data is valuable. The fitness-aligned pLMs can better predict the fitness and propose useful mutation, which can significantly boost the protein design process.

In the ProteinGym benchmark, our fitness-aligned pLM shows great improvement in the fitness prediction by using less than 500 mutants without any evolutionary information (see Table 1). We also show that aligning pLM to ranked fitness data enables the generation of fit protein variants with a success rate 6~20$\times$ of zero-shot and 2~8$\times$ of state-of-the-art method in two tasks (see Table 4). Moreover, we show that the training pairs significantly affect the fitness prediction of aligned pLMs (See Table 2), and can be effectively curated by our method employing reference policy.

2 Related works

Predicting the effects of mutations has traditionally focused on the analysis of aligned protein sequences (MSA) to extract the position-specific information [12]. Over time, models have evolved to capture more complex patterns, using energy-based model [8] or variational autoencoder [18, 7]. With the limitations posed by solely training on MSAs and the advances in natural language processing, recent research focuses on transformer models [11, 6] that are trained on a large database of protein sequences across various families. Recently [14] introduced Tranception, a transformer architecture that learns patterns and applies attention to contiguous subsequences. Leveraging the capabilities of the AlphaFold [9], Alpha-Missense [4] predicts the pathogenicity of potential amino acid substitutions. The model is trained to predict structure along with masked language modeling. Simultaneously, there emerged an interest in optimizing fitness, in applications such as drug discovery. Genhance [3] performs generation by making perturbations in a latent space, and ICE [15] masks and infills the amino acids with higher predicted fitness.

2.1 Conditioning pLMs

The approaches for conditioning pLMs can be divided into three: (i) multi-modal training with natural language [21], (ii) control tag approach [13], and (iii) MSA-conditioned model [17, 20]. ProtST [21] performs multimodal representation alignment and mask prediction between pLM and biomedical language model, allowing accurate protein classification and retrieval. It is trained on a dataset consisting of protein names, functions, subcellular locations, and families. Progen2 [13] employs a prompt tuning approach, which provides a token indicating the Pfam ID. However, both approaches fall short in fine-grained conditions due to the limited granularity of the conditions. Conversely, our approach can express any condition by ranking the mutants. Recently, [20] proposed a pLM with a sequence-of-sequence attention method that can effectively learn from MSA.

3 Fine-tuning pLM by ranking fitness

Let us denote protein $\boldsymbol{x}$ by sequence of amino acids, *i.e.*, $\boldsymbol{x} = (x_1, \ldots, x_\ell)$. We consider an autoregressive protein Language Model (pLM) [14], which predicts the next amino acid x_{i+1} with

the prior subsequence of amino acids $(x_1, \ldots, x_i)$. Then the predicted fitness $\widehat{F}$ of a mutant $\boldsymbol{x}^{\text{mut}}$ is defined by the log-likelihood ratio with respect to the wild-type protein sequence $\boldsymbol{x}^{\text{wt}}$ as follows:

$$\widehat{F}\big(\boldsymbol{x}^{\text{mut}}\big) = \log \frac{P(\boldsymbol{x}^{\text{mut}})}{P(\boldsymbol{x}^{\text{wt}})}, \quad \text{where } P(\boldsymbol{x}) = \prod_{i=1}^{\ell} P(x_i \,|\, x_{<i}). \tag{1}$$

3.1 Proposed Method

Fine-tuning objective. We propose to fine-tune pLM by ranking the protein variants using fitness data $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$. To model the ranking between proteins, we use Bradley-Terry (BT) model [2], which estimates the score function from pairwise comparisons. Formally, given a pair of mutants $(\boldsymbol{x}_w, \boldsymbol{x}_l)$, where $\boldsymbol{x}_w$ has higher fitness than $\boldsymbol{x}_l$, BT models the distribution of ranking by

$$p^*(\boldsymbol{x}_1 \succ \boldsymbol{x}_2) = \frac{\exp\left(r(\boldsymbol{x}_1)\right)}{\exp\left(r(\boldsymbol{x}_1)\right) + \exp\left(r(\boldsymbol{x}_2)\right)}, \tag{2}$$

where $r(\boldsymbol{x})$ denotes the score function (*i.e.,* reward models) to rank the fitness of mutant $\boldsymbol{x}$. Instead of training auxiliary reward models, we directly fine-tune pLM by using the implicit reward model following [16]. In specific, the score function r can be expressed by the log-likelihood ratio between the reference policy π_{ref} and optimizing policy π_θ, then the maximum likelihood objective is given by

$$\mathcal{L}_{\text{ranking}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(\boldsymbol{x}_w, \boldsymbol{x}_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(\boldsymbol{x}_w)}{\pi_{\text{ref}}(\boldsymbol{x}_w)} - \beta \log \frac{\pi_\theta(\boldsymbol{x}_l)}{\pi_{\text{ref}}(\boldsymbol{x}_l)} \right) \right], \tag{3}$$

where $\beta > 0$ is a parameter controlling the deviation from the reference policy. Intuitively, the ranking loss increases the likelihood of fit mutants $\boldsymbol{x}_w$ and decreases the likelihood of less fit mutants $\boldsymbol{x}_l$. Remark that one can use different supervised training objective such as regression, to use fitness data in fine-tuning pLMs. However, we empirically observe that using regression in fine-tuning pLMs even show worse performance than zero-shot models (see Table 1). This is due to the existence of label bias in the fitness dataset [4].

Pair construction In fine-tuning pLMs with ranking loss, the choice of ranked data pair is crucial for learning effective pLM (*e.g.,* see Table 2). Given the sorted set of mutants $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ with known fitness order $\widehat{F}(\boldsymbol{x}^{(1)}) > \widehat{F}(\boldsymbol{x}^{(2)}) \cdots > \widehat{F}(\boldsymbol{x}^{(N)})$, the adjacent pairs $\mathcal{P}_{\text{adjacent}} = \{\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(i+1)}\}_{i=1}^{N-1}$ is straightforward in representing the ranks of the mutants. However, this approach degrades the performance as the model struggles to discriminate the mutants of similar fitness. Especially, the fitness values might be noisy as they are excerpted from human-experiments [7, 4], or different protein sequences might have same phenotype. Thus, we propose to sample the training pair by selecting pair that has distance larger than a threshold. Since the optimal threshold might different from different tasks, we introduce a generic method which uses the reference model π_{ref} to curate the pairs, where we demonstrate in Algorithm 1. We employ the ranking i of the mutant $\boldsymbol{x}^{(i)}$, and aim to find the optimal distance $d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = |i - j|$ threshold to sample training pairs. The key idea of curation is that the accuracy of reference policy π_{ref}, *i.e.,* we define

$$A_{\mathcal{D}}(d) = \frac{1}{N-d} \sum_{i=1}^{N-d} 1, \quad \text{if } P(\boldsymbol{x}^{(i)}) > P(\boldsymbol{x}^{(i+d)}), \quad \text{else } 0, \tag{4}$$

which can be used as a proxy for the quality and the difficulty of the pairs. The hard protein pair $(\boldsymbol{x}_w, \boldsymbol{x}_l)$, *i.e.,* that has low value of $A_{\mathcal{D}}(d)$ might be informative, yet they might contain the label noise or $\boldsymbol{x}_w$ and $\boldsymbol{x}_l$ have same phenotype. On the other hand, easy examples from d with high $A_{\mathcal{D}}(d)$ value are less informative, and training from those examples degrades the performance (See Section 4.1). Since our method solely rely on ranking of fitness, it can be applied to fitness/functionality of proteins that are hard to quantify and/or calibrate.

3.2 Proposed framework

We introduce a general framework for few-shot fine-tuning of pLM to fitness data, as shown in Figure 2.

1. **Self-supervised learning.** We use Tranception [14], a state-of-the-art pLM in fitness prediction. Our approach is general and can be applied to other pLMs trained in both autoregressive and masked language modeling objective [10] (See Appendix A.2).

Table 1: **Fitness prediction results.** We report Spearman's rank correlation (mean ρ) of various zero-shot, conditionally trained, and fine-tuned fitness prediction methods. †Parentheses indicate the type of condition, and the result are calculated in all 87 assays.

	Method	Mean ρ (↑)
Zero-shot	Baseline [14]	0.407
	Baseline + MSA	0.455
	ESM-1v [11]	0.382
Conditional†	ProtST (Language) [21]	0.412
	Progen2 XL (Control tag) [13]	0.402
	MSA-Transformer (MSA) [17]	0.423
	PoET (MSA) [20]	0.484
MSA training	EVE [7]	0.453
	Alpha-Missense [4]	0.527
Few-shot regression		0.280
Few-shot ranking	SFT	0.430
	SFT + Align	**0.566**

Table 2: Spearman's rank correlation by the distance of pairs used in Algin phase on 20 selected DMS assays (See Appendix A.3).

Distance	Mean ρ (↑)
1 ($\mathcal{P}_{\text{adjacent}}$)	0.460
64	0.562
128	0.593
256	0.588
Ours (d_{ref})	**0.597**
Baseline	0.464
SFT	0.489
Alpha-Missense [4]	0.552

2. **Supervised fine-tuning (SFT) to the highest fitness mutants.** Given that zero-shot model might not capture the specificity required for predicting the certain protein fitness, we fine-tune model to the highest fitness mutants using maximum likelihood objective, obtaining a reference model π_{ref}.
3. **Mutant pair construction.** We find d_{ref} and curate mutant pairs $\mathcal{P}$ following Algorithm 1.
4. **Align model to curated pairs.** We align the reference model π_{ref} to curated mutant pairs $\mathcal{P}$ using maximum likelihood objective as described in Eq. 3, resulting an aligned model π_θ.
5. **Iterative extrapolation (IE) using aligned model.** We leverage the aligned pLM to repeatedly predict the fitness and subsequently mutate protein sequences, as detailed in Algorithm 2.

4 Experimental results

4.1 Fitness prediction

We evaluate the fitness prediction performance of aligned pLM on the ProteinGym substitution benchmark, which consists of 85 Deep Mutational Scanning (DMS) assays of various fitness. We use Spearman's rank correlation coefficient (ρ) between predicted fitness $\widehat{F}$ and the experimental fitness F as a metric, following [14]. See Appendix A.3 for implementation details.

As shown in Table 1, our ranking-based fine-tuning significantly increases the fitness prediction performance by only using a maximum of 500 data points. Considering that nearly 50% of ProteinGym consists of > 4000 data, it is only 12.5% of the data. Our pipeline surpasses all types of conditionally trained models with large margin, by being able to encompass fine-grained conditions. We also outperform Alpha-Missense [4], which exploits a rich structural prior. Table 2 shows the significance of mutant pairs when aligning pLM to fitness data. pLM tuning using the distance discovered by the reference model shows the best performance, implying that the optimal distance (i) differs by the DMS assay, and (ii) can be effectively found using our proposed method. As shown in Table 3, the decrease in performance is slight even when using a maximum of 200 labeled data (5% of the data). See Figs 4-5 for a comparison of our method to baselines in DMS-level.

Comparison with Alpha-Missense. pLM also holds a significant advantage over Alpha-Missense by being able to handle insertions, deletions, and multiple substitutions. As shown in Fig 3, our method excels in DMS assays including multiple amino acid substitutions. Since Alpha-Missense only supports single amino acid substitutions and calculate the effect of multiple amino acid substitutions as summation of single amino acid substitutions, performance is slightly lower in DMS assays including multiple substitutions than only including single substitutions.

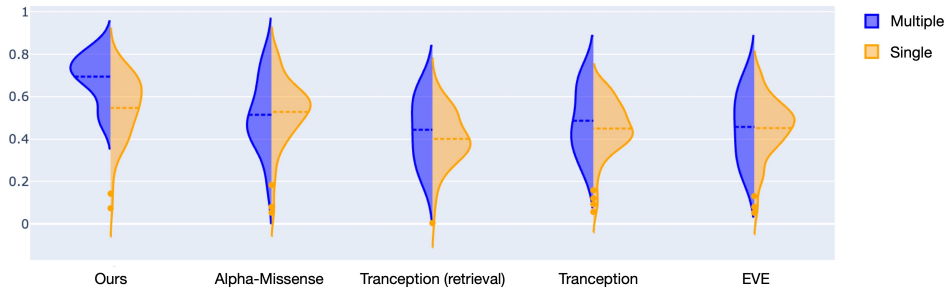


Figure 3: **Performance by multiple amino acids substitution.** Performance of assays that include multiple amino acid substitutions (blue) and include only single amino acid substitution (yellow).

Table 3: **Ablation on the number of shots.** on 20 selected DMS assays.

Shots	Mean ρ (↑)
200	0.573
500	0.597

Table 4: **Fitness optimization results.** We report success rates of three extrapolation regions in AAV and ACE tasks.

Method	AAV success rate			ACE success rate		
	> 0	> 1	> 2	$<$ -5	$<$ -6	$<$ -7
ICE [15]	0.223	0.036	0.002	0.361	0.098	0.019
pLM (Zero-shot)	0.261	0.027	0.008	0.051	0.025	0.018
pLM (SFT)	0.489	0.174	0.075	0.122	0.048	0.013
pLM (SFT + Align)	**0.541**	**0.293**	**0.161**	**0.363**	**0.240**	**0.112**

4.2 Fitness optimization

Protein fitness optimization aims to generate novel variants that have high fitness. We follow the notion of training and extrapolation region as introduced in [15]. Since we aim to design proteins with attributes higher than ever seen, the training dataset is restricted to have attributes below a certain threshold, namely *training region*. *Extrapolation region* corresponds to the range of attribute values we want to design. Following the prior works, we report the *success rate*, *i.e.*, the fraction of variants in the extrapolation region. We compare the performance of our method to the state-of-the-art IE method, ICE [15]. Note that we excluded black-box optimization baselines since they assume interaction with oracle during optimization.

AAV fitness. We generate variants of the adeno-associated virus (AAV) capsid protein that have a **higher** fitness value. We evaluate the fitness by the CNN oracle as proposed in FLIP benchmark [5]. As shown in Table 4, pLM aligned to experimental data is able to generate fit variants with large margin from ICE. It is notable that 16.1% of our proposed variants have fitness > 2, which is very successful compared to 0.2% of ICE.

ACE stability. We generate variants of the human angiotensin-converting enzyme 2 (ACE2) with higher stability, *i.e.*, **lower** energy. We measure the *ddG*, a change in free energy from the wild-type ACE2 protein via FoldX suite [19]. As shown in Table 4, pLM aligned to experimental data generates very stable mutants with a large margin from ICE. It is also notable that 11.2% of our proposed variants have ddG $<$-7, which is very successful compared to 1.9% of ICE.

5 Conclusion

In this work, we propose to fine-tune the protein language model by ranking the protein fitness. To this end, we curate the mutant pair using the accuracy of the reference policy as a proxy, then propose to fine-tune pLMs by the maximum likelihood objective of the Bradley-Terry model. Through experimental validation on the ProteinGym benchmark, we show our method outperforms previous methods by using only 200 labeled data. Furthermore, we show our approach can optimize the fitness/functionality of proteins by simply mutating the amino acids with high predicted fitness. We will further enhance the efficiency by studying parameter efficient fine-tuning. We believe our method can be generalized to pLMs trained to masked language modeling objectives as well.

References

[1] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[2] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[3] A. Chan, A. Madani, B. Krause, and N. Naik. Deep extrapolation for attribute-enhanced generation. *Advances in Neural Information Processing Systems*, 34:14084–14096, 2021.

[4] J. Cheng, G. Novati, J. Pan, C. Bycroft, A. Žemgulytė, T. Applebaum, A. Pritzel, L. H. Wong, M. Zielinski, T. Sargeant, R. G. Schneider, A. W. Senior, J. Jumper, D. Hassabis, P. Kohli, and Žiga Avsec. Accurate proteome-wide missense variant effect prediction with alphamissense. *Science*, 0(0):eadg7492.

[5] C. Dallago, J. Mou, K. E. Johnston, B. J. Wittmann, N. Bhattacharya, S. Goldman, A. Madani, and K. K. Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11, 2021.

[6] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[7] J. Frazer, P. Notin, M. Dias, A. Gomez, J. K. Min, K. Brock, Y. Gal, and D. S. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

[8] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. Schärfe, M. Springer, C. Sander, and D. S. Marks. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

[9] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[10] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

[11] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

[12] P. C. Ng and S. Henikoff. Predicting deleterious amino acid substitutions. *Genome research*, 11(5):863–874, 2001.

[13] E. Nijkamp, J. Ruffolo, E. N. Weinstein, N. Naik, and A. Madani. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

[14] P. Notin, M. Dias, J. Frazer, J. M. Hurtado, A. N. Gomez, D. Marks, and Y. Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[15] V. Padmakumar, R. Y. Pang, H. He, and A. P. Parikh. Extrapolative controlled sequence generation via iterative refinement. *arXiv preprint arXiv:2303.04562*, 2023.

[16] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[17] R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, and A. Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[18] A. J. Riesselman, J. B. Ingraham, and D. S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[19] J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005.

[20] T. F. Truong Jr and T. Bepler. Poet: A generative model of protein families as sequences-of-sequences. *arXiv preprint arXiv:2306.06156*, 2023.

[21] M. Xu, X. Yuan, S. Miret, and J. Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. *arXiv preprint arXiv:2301.12040*, 2023.

A Appendix

A.1 Algorithms

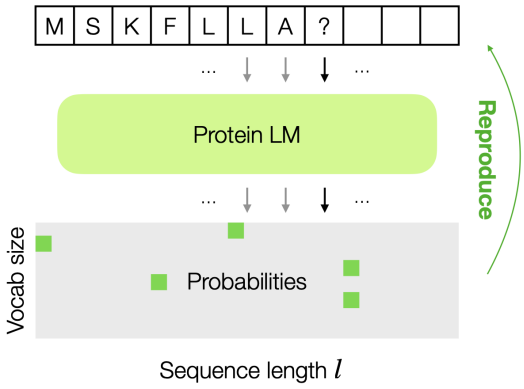
In this section, we detail the algorithms used to construct pair and the iterative extrapolation.

Algorithm 1 Pair construction

Require: Dataset of mutants $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ with known fitness order $\widehat{F}(x^{(1)}) > \cdots > \widehat{F}(x^{(N)})$
Require: $A_{\textbf{threshold}}$ hyperparameter to decide the quality and/or the difficulty of pairs
$\mathcal{P} = \{\}$ set of curated pairs
$d_{\text{ref}} = \min d$ such that $A_{\mathcal{D}}(d) > A_{\textbf{threshold}}$
for $i \in [1, N - 2 \cdot d_{\text{ref}}]$ **do**
    for $j \in [i + d_{\text{ref}}, i + 2 \cdot d_{\text{ref}}]$ **do**
        $\mathcal{P} \leftarrow \mathcal{P} \cup (x^{(i)}, x^{(j)})$
    end for
end for

Algorithm 2 Iterative extrapolation using LM

Require: $\mathcal{I}$, set of initial sequences
Require: G, number of iterations
Require: N_1, children per each parent
Require: N_2, sequences to reproduce
$\mathcal{R} = \mathcal{I}$, set of selected parents
$\mathcal{C} = \emptyset$, set of proposed sequences
for $g \in [0, G)$ **do**
    for $r \in \mathcal{R}$ **do**
        $\mathcal{C}_r \leftarrow$ top N_1 probable mutants of r
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_r$
    end for
    $\mathcal{R} \leftarrow$ top N_2 fit sequences of $\mathcal{C}$
end for



A.2 Application to pLM trained to masked language modeling objective

In this section, we summarize the adpation of our method to pLMs trained to masked language modeling objective (masked pLM), such as ESM2 [10] and language model head of Alpha-Missense [4]. Masked pLM is trained in a self-supervised fashion to predict the masked token `<mask>` in the protein sequence based on the context of other amino acids. Let us denote the mutant $\boldsymbol{x}^{\text{mut}}$ and the wild-type protein sequence $\boldsymbol{x}^{\text{wt}}$. The sequence $\boldsymbol{x}^{\text{mask}}$ is defined as:

$$\boldsymbol{x}^{\text{mask}} = \begin{cases} \texttt{<mask>} & x_i^{\text{wt}} \neq x_i^{\text{mut}} \\ x_i^{\text{wt}} & x_i^{\text{wt}} = x_i^{\text{mut}} \end{cases}.$$

Then the predicted fitness $\widehat{F}$ of a mutant $\boldsymbol{x}^{\text{mut}}$ is defined by the summation of log-likelihood ratio with respect to the wild-type protein sequence $\boldsymbol{x}^{\text{wt}}$ as follows:

$$\widehat{F}(\boldsymbol{x}^{\text{mut}}) = \sum_{0 \leq i < l, x_i^{\text{wt}} \neq x_i^{\text{mut}}} \log \frac{P(x_i^{\text{mut}} | \boldsymbol{x}^{\text{mask}})}{P(x_i^{\text{wt}} | \boldsymbol{x}^{\text{mask}})}$$

Similarly, given a pair of mutants $(\boldsymbol{x}_w, \boldsymbol{x}_l)$, where $\boldsymbol{x}_w$ has higher fitness than $\boldsymbol{x}_l$, and $\boldsymbol{x}^{\text{mask}}$ is defined as:

$$\boldsymbol{x}^{\text{mask}} = \begin{cases} \texttt{<mask>} & x_{w,i} \neq x_{l,i} \\ x_{w,i} & x_{w,i} = x_{l,i} \end{cases},$$

the maximum likelihood objective for BT model is given as follows:

$$\mathcal{L}_{\text{masked}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(\boldsymbol{x}_w, \boldsymbol{x}_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(\boldsymbol{x}_w|\boldsymbol{x}^{\text{mask}}))}{\pi_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{x}^{\text{mask}}))} - \beta \log \frac{\pi_\theta(\boldsymbol{x}_l|\boldsymbol{x}^{\text{mask}}))}{\pi_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{x}^{\text{mask}})} \right) \right]. \quad (5)$$

A.3 Implementation details for fitness prediction

Baseline. We use a large checkpoint of Tranception [14] as a baseline pLM. pLM can be used with or without multiple sequence alignment (MSA) retrieval, which provides model an evolutionary information. We indicate the usage of MSA information with '+ MSA'.

Data processing. We truncate the sequence length to a maximum of 768. We match the center of the mutation region with the center of truncation when truncating the sequence. When the mutation region is larger than 768, we did not truncate the sequence and use a small checkpoint of Tranception (2 datasets BRCA1_HUMAN_Findlay_2018, and MSH2_HUMAN_Jia_2020). We excluded two datasets (POLG_HCVJF_Qi_2014, SCN5A_HUMAN_Glazer_2019) since the annotation of the mutation region did not match the data, which resulted in a total of 85 datasets. Note that EVE and ESM-1v are evaluated in full DMS assay, whereas other methods are evaluated in test subset of DMS assay (which is the size of $\min(|\mathcal{D}| - 1000, 0.6|\mathcal{D}|)$. For two datasets (HIS7_YEAST_Pokusaeva_2019, SPG1_STRSG_Olson_2014) we sample 20,000 data for the test subset considering the large data size and limited resources.

Ablation benchmark. We randomly choose 20 subsets to report richer ablation results. The criteria is $2000 < |\mathcal{D}| < 10000$ and the size of mutation region ≤ 768. See Supplementary material for a full list of subsets used in ablation.

SFT and Align. We use top 20% of the training data (*i.e.*, maximum 100 data points for the results in Table 1) in SFT phase. $A_{\text{threshold}}$ is set to 0.65. For faster training, we (i) use 5000 pairs randomly chosen from curated pairs when using a Large checkpoint of Tranception, or (ii) use a Small checkpoint of Tranception to efficiently train a model to full curated pairs. See Supplementary material for a list of assays that we used Small checkpoint of Tranception.

Regression objective. We compare our ranking objective with the regression objective as well. We replace the final token classification layer of Tranception with the multi-layer perceptron (MLP) and train to the few-shot fitness data with L2 loss.

A.4 Implementation details for fitness optimization

We used smallest checkpoint of Tranception for the fast inference. We set $|\mathcal{I}| = 100, G = 10, N_1 = 10, N_2 = 100$ for both tasks. We propose 1,000 variants following Algorithm 2 and evaluate the success rate. Note that baseline methods generate 10,000 variants.

A.5 Fitness prediction performance for optimization tasks

We report the reward model performance, *i.e.* fitness prediction performance of pLMs used in optimization tasks in the Table 5.

Table 5: Spearman's rank correlation in two tasks

Task	AAV fitness	ACE2 stability
pLM (Zero-shot)	0.0451	0.1153
pLM (SFT)	0.2471	0.3735
pLM (SFT+ Align)	0.2716	0.4459

A.6 Detailed performance on the ProteinGym benchmark

We report DMS level results for our model, Alpha-Missense, Tranception, EVE, and ESM-1v in ProteinGym benchmark in Figs 4-5.

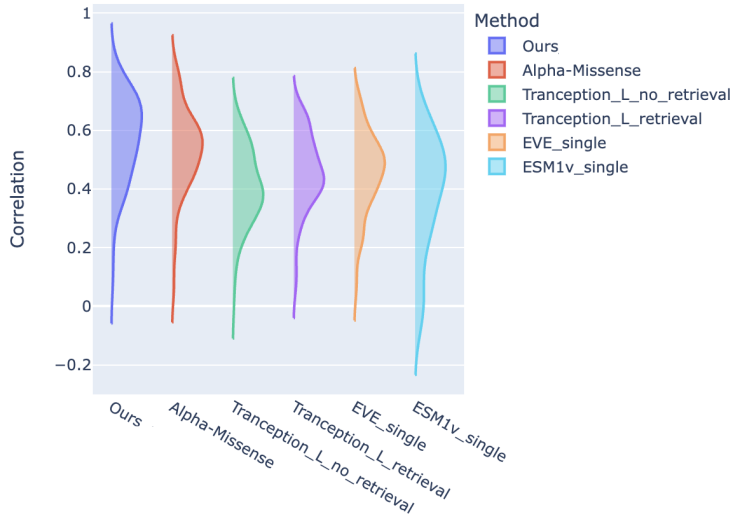


Figure 4: **Distribution of performance** on the ProteinGym benchmark.

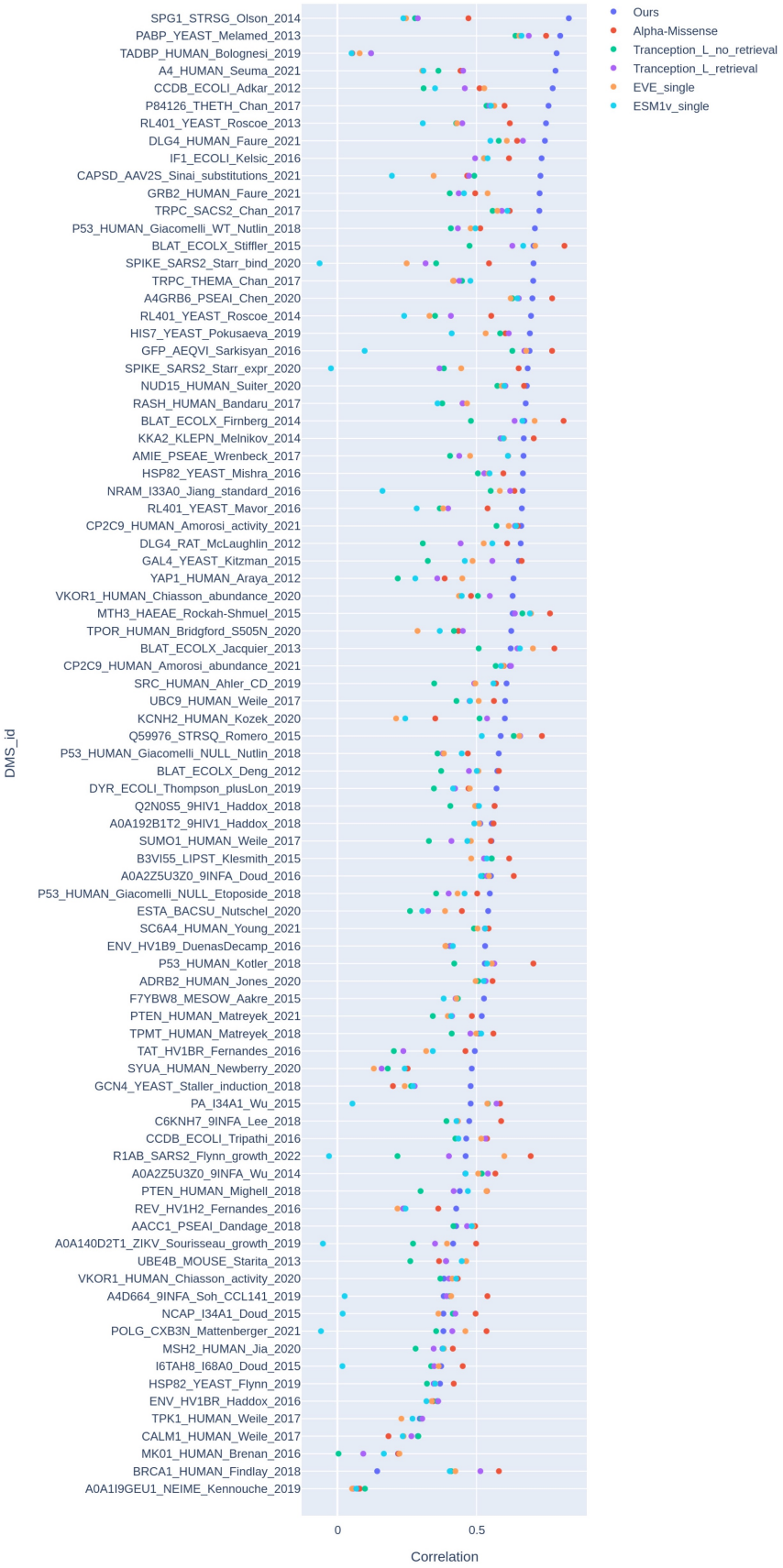


Figure 5: **Full results.** DMS-level performance on the ProteinGym benchmark.