Contrastive Fitness Learning: Reprogramming Protein Language Models for Low-N Learning of Protein Fitness Landscape

Junming Zhao[1], Chao Zhang[1], Yunan Luo[1,*]

[1] School of Computational Science and Engineering, Georgia Institute of Technology
[*] Corresponding author: yunan@gatech.edu

Abstract

Machine learning (ML) is revolutionizing our ability to model the fitness landscape of protein sequences, which is critical to answering fundamental life science questions and addressing important protein engineering applications, such as quantifying the pathogenicity of disease variants, forecasting viral evolution in a pandemic, and engineering new antibodies. Recently, the protein language model (pLM) has emerged as an effective ML tool in deciphering the intrinsic semantics of protein sequences and become the foundation of state-of-the-art ML solutions for many problems in protein biology. However, significant challenges remain in leveraging pLMs for protein fitness prediction, in part due to the disparity between the scarce number of sequences functionally characterized by high-throughput assays and the massive data samples required for training large pLMs. To bridge this gap, we introduce Contrastive Fitness Learning (ConFit), a pLM-based ML method for learning the protein fitness landscape with limited experimental fitness measurements as training data. We propose a novel contrastive learning strategy to fine-tune the pre-trained pLM, tailoring it to achieve protein-specific fitness prediction while avoiding overfitting, even when using a small number (low-N) of functionally assayed mutant sequences for supervised fine-tuning. Evaluated across over 30 benchmark datasets of protein fitness, ConFit consistently provided accurate fitness predictions and outperformed several competitive baseline methods. Further analysis revealed that ConFit's capability of low-N learning enabled sample-efficient active learning for identifying high-fitness protein variants. Collectively, our work represents a novel strategy to harness the potential of pLMs to elucidate the protein sequence-function relationship. The source code of ConFit is available at https://github.com/luo-group/ConFit.

1 Introduction

Evolution has shaped natural proteins to perform vital functions in life science and address pressing applications ranging from developing targeted therapeutics[1] to engineering herbicide-resistant crops[2] and sustainable biocatalysis[3]. The function capabilities of naturally occurring proteins are further expanded through a biotechnology technique known as protein engineering, which enhances a natural protein's original function (e.g., improving an enzyme's catalytic activity[4]) or repurposes it to a different but related function (e.g., repurposing an antibody to target a new virus[5]), often by introducing "beneficial" mutations into the protein's amino acid sequence. A protein's function is encoded by its amino acid sequence, and the sequence-fitness landscape maps out the relationships between a protein sequence and function. The goal of protein engineering is to search this landscape for high-fitness sequences. Here. 'fitness' broadly refers to protein properties such as ligand-binding affinity, stability, or catalytic activity. Despite the success in practice[6;7], using protein engineering to design or discover new novel proteins with desired functions is still challenging because i) the laboratory experiments are time- and resource-intensive and ii) the sequence search space is astronomically large (scaling exponentially with protein length).

Machine learning (ML) has emerged as an effective approach to infer the sequence-fitness relationship. Used as surrogate models, ML methods, including both supervised and unsupervised models, predict the properties of proteins more rapidly and less expensively than wet-lab experiments, reducing the overall experimental burden in protein engineering. Unsupervised generative ML models have been developed to learn the intrinsic semantics in naturally occurring protein sequences. Although not labeled with measurements for the property of interest, those sequences are subjected to evolutionary pressure and implicitly encode sequence constraints that are prerequisites for protein fitness. To capture those constraints, unsupervised approaches often fit a probability density model (e.g., hidden Markov models[8], Potts models[9–19], latent variable models[20–22], and protein language models[23–29]) on natural sequences – either homologous sequences of the target protein or massive natural protein sequences from databases such as UniProt[30] – to estimate the occurrence probability $p(\boldsymbol{x})$ of a particular sequence $\boldsymbol{x}$ as a proxy of its fitness[9].

Supervised models are particularly useful for explicitly learning the sequence-fitness relationship when paired data of variant sequences and experimental fitness are richly available, often outperforming unsupervised models[31–34]. Yet, their effectiveness is constrained by the scarcity of fitness data. Screening variant fitness can be lab-intensive and require developing tailored assays for a specific function, with typical experiments yielding only 10^0-10^3 labeled variants. However, modern ML models such as neural networks are notoriously data-hungry and often require $>10^5$ training samples. While high-throughput assays can screen fitness at scale, they may compromise fidelity for throughput[35], and not all proteins have high-throughput assays. This gap has spurred the development of ML models that can make accurate fitness predictions even with small-size ('low N') training data. A common strategy is to leverage unsupervised models to capture general sequence patterns across natural proteins, which hold implications for functions, and then adapt those models using low-N fitness data to make supervised, protein-specific fitness prediction[36–38].

Protein language models (pLMs), in particular, are effective unsupervised models used in existing low-N fitness prediction methods[36–38] for learning sequence semantics that are likely to occur in natural proteins. Interestingly, even without fitness-labeled data, pLMs have shown accurate strong performance in predicting the effects of amino acid substitutions in protein sequences, correlating well with experimentally measured fitness data[39–43]. Given their promising predictive capacity, we reason that with even a small number of fitness data, pLMs could be fine-tuned into more accurate supervised models. However, directly fine-tuning a pLM (typically with $\sim 10^9$ parameters) on sparse low-N data ($\sim 10^2$ samples) will likely lead to overfitting, abruptly distorting its learned sequence patterns during pre-training – a phenomenon known as *catastrophic forgetting* in ML[44]. Despite its prevalence in ML-guided protein engineering, only a few studies attempted to tackle it, often compromising prediction accuracy to avoid overfitting[31;36].

Here, we introduce ConFit (Contrastive Fitness Learning), an ML algorithm that effectively reprograms a pre-trained pLM to high-accuracy, sample-efficient fitness prediction model under low-N settings. Our method represents a new paradigm of pLM fine-tuning, which preserves the evolutionary patterns the pLM has learned during pre-training while effectively incorporating small-size fitness data to enable specific and accurate fitness predictions without overfitting. Recognizing that in ML-guided protein engineering (MLPE), ranking variants by their fitness is more crucial than predicting exact fitness values for individual variants, we shift from traditional regression[36;37] to a contrastive learning approach. Our method calibrates the pLM to

predict the *relative fitness order* between pairwise variants rather than their *absolute fitness values*. This not only expands the effective training set quadratically but also circumvents catastrophic forgetting effectively. Benchmarking on over 30 mutagenesis datasets of protein fitness, we showed that ConFit can accurately predict protein fitness even with only N=48 fitness-labeled sequences as training data, outperforming other unsupervised and supervised models. The outstanding low-N prediction capability of our method also enabled a sample-efficient active learning loop for identifying function-enhanced variants. We expect ConFit to be a practically useful tool for guiding and accelerating protein engineering, especially for applications where experimental fitness screening is costly.

2 Protein Language Models for Fitness Prediction: Revisited

The advent of ChatGPT and subsequent LLMs[45–48] has not only revolutionized natural language processing but has also had a transformative impact on computational biology through the development of protein language models (pLMs)[23–29]. pLMs adopt LLM strategies, applying self-supervised learning to predict masked amino acids in protein sequences given other residues as context, uncovering patterns indicative of natural protein grammar. Many state-of-the-art ML algorithms for protein biology now rely on pLMs, from the prediction of protein structure[28] to protein properties like solubility[49], stability[24], and binding affinity[43], with particular effectiveness in protein fitness prediction[27;34;43]. The success of pLMs hinges on their extensive model size and the massive training data, allowing for learning general patterns across natural proteins that transfer well to other problems where labeled data are limited.

Formally, we denote the amino acid (AA) sequence of a protein by $\boldsymbol{x} = (x_1, \ldots, x_L) \in \mathcal{X}^L$, where x_i is the i-th AA, L the sequence length, and $\mathcal{X}$ the alphabet of possible AAs (e.g., the 20 standard AAs). Masked pLMs[27;28] aims to learn the conditional probability $p(x_i|x_{-i})$ of an AA appearing at a given position, conditioned on the sequence excluding that position, where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_L)$ (Fig 1a). While our focus is on masked pLMs in this work, we acknowledge alternative models like autoregressive pLMs[29;50], which estimate $p(x_i|x_{<i})$, taking into account all preceding AAs $x_{<i} = (x_1, \ldots, x_{i-1})$.

The natural protein sequences observed today are the culmination of evolutionary selection for fitness and function. Trained on massive natural protein sequence database such as UniProt[30], pLMs implicitly capture the sequence patterns that are important for fitness. Specifically, the conditional probability $p(x_i|x_{-i})$ estimated by pLMs gives a probability distribution over $\mathcal{X}$ for a given site i, indicating the site's 'evolutionary preference' for an AA $x_i \in \mathcal{X}$. A higher $p(x_i|x_{-i})$ intimates an AA x_i that is evolutionarily advantageous, presumably contributing positively to fitness, while a lower value suggests potentially deleterious mutations. Following this rationale, recent studies directly used the PLM-predicted likelihood for zero-shot predictions of AA mutation effects. Even without any supervision from paired sequence-fitness data, pLM's predictions correlate strongly with laboratory-measured protein fitness (e.g., deep mutagenesis scans)[39–41]. PLMs have thus been harnessed for important biomedical applications, including recommendation mutations to enhance antibody binding[43] or predicting the pathogenicity of human missense variants[51].

Given that the pLMs have strong zero-shot fitness prediction performance despite having no prior exposure to fitness data, one may reason that augmenting such models with even a modest number of functionally characterized variants as training data (Fig 1b) could significantly improve their fitness prediction accuracy. The intuitive strategy might be to fine-tune the pLM with available fitness data (Fig 1c), following common practice in ML[52]. However, the unique challenge in protein engineering is that current quantitative assays to measure fitness are not scalable to high throughput, yielding only tens to hundreds of fitness measurements. This paucity of data, juxtaposed against the millions or even billions of parameters in pLMs, collectively presents a challenge where directly fine-tuning pLMs on such sparse data easily leads to catastrophic forgetting, distorting the pLM's pre-trained protein sequence features and causing the model to overfit the sparse fitness data.

To mitigate catastrophic forgetting in low-N fitness prediction, some studies have adopted a compromised approach: freezing the pLM parameters while introducing a new trainable regression layer—typically a ridge or LASSO model—on top of pLM-generated sequence representations[36;37] (Fig 1d). This hybrid model, with significantly fewer trainable parameters, is less prone to overfitting and has been shown to accurately predict fitness even using only $N = 24$ or 48 variants as training data[36;37]. However, these approaches mitigated catastrophic forgetting at the expense of limited prediction accuracy: freezing the pLM's layers

forfeits its capacity to capture complex epistatic interactions [53;54] inherent in protein sequences—a task linear models find challenging. To date, how to fully harness the pre-learned knowledge in pLMs for low-N fitness prediction, without catastrophic forgetting and overfitting, remains an unresolved challenge.

3 Methods

We present ConFit, a pLM-based ML method for low-N protein fitness prediction. Our algorithm is motivated by a key observation of a mismatch exists between the pre-training and fine-tuning learning objectives, which leads to catastrophic forgetting during the latter. We propose a principled way to achieve robust low-N pLM fine-tuning for protein fitness prediction. Different from existing methods that either fully fine-tune the pLM ("full fine-tuning"), suffering from severe overfitting in low-N scenarios, or only re-fit the pLM's top layer while keeping the rest untouched ("top-layer fine-tuning"), resulting in compromised accuracy, our method provides both high accuracy and high sample efficiency for low-N protein engineering [36].

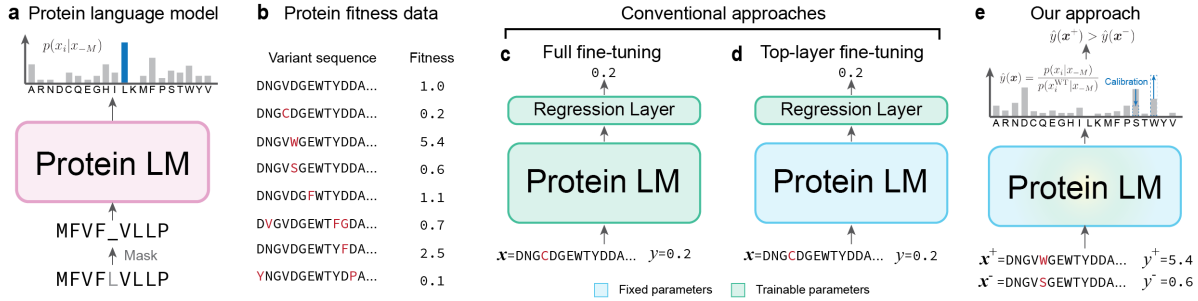

Figure 1: Overview of ConFit. (a) A protein language model (pLM) predicts the probability of amino acid at position given other unmasked positions. (b) Example of protein fitness data in which the variants of a wildtype (WT) sequence are experimentally characterized with fitness values. (c-d) Conventional approaches for fine-tuning pre-trained pLMs for protein fitness prediction, including "full fine-tuning" which updates all parameters in the pLM, and "top-layer fine-tuning" which trains the top-layer on the fitness data while fixing other pre-trained parameters. (e) Our approach, ConFit, calibrates pre-trained pLM for low-N fitness prediction through contrastive learning.

3.1 Motivation and Rationale of ConFit

The motivation behind ConFit lies in the realization that catastrophic forgetting occurs when a pLM is fine-tuned for a task divergent from its initial training. Originally, pLMs maximized the likelihood of natural protein sequences sampled from the training data, i.e., $\max \mathbb{E}_{\boldsymbol{x}} \prod_i p(x_i|x_{-i})$. However, during fine-tuning for fitness prediction, the objective shifts to minimizing the difference between observed and predicted fitness values, measured by mean squared error (MSE): $\min \sum_k \|y^{(k)} - \hat{y}^{(k)}\|^2$, where $y^{(k)} \in \mathbb{R}$ and $\hat{y}^{(k)} \in \mathbb{R}$ are the assayed and predicted fitness for variant k, respectively. This shift not only demands a change of output space, from the probability distribution space over $\mathcal{X}$ to the space of all real values $\mathbb{R}$, but also compels the model to reconfigure its learned parameters, often at the cost of erasing its pre-trained features, especially when the volume of fine-tuning data is limited.

Based on this observation, our key idea to address catastrophic forgetting is to align the fine-tuning objective with the pre-training objective. We hypothesize that the pre-trained pLM, already adept at assessing the evolutionary plausibility of sequence mutations, only needs minor adjustments in its output probability distributions to characterize a protein-specific fitness landscape. Therefore, rather than retraining the pLM on a different task (MSE minimization), we maintain the likelihood maximization paradigm and only subtly adjust the pLM's parameters such that the output probability distribution $p(x_i|x_{-i})$ more closely correlate with the given low-N fitness data, thereby enhancing the pLM's prediction accuracy without losing its pre-trained evolutionary patterns.

To gain more intuitions about our method, let us consider a special example of single-mutation variants (Fig 1e). For the i-th residue of a wildtype sequence $\boldsymbol{x}$, a pLM predicts a 20-dimensional probability distribution $p(x_i|x_{-i})$. Previous studies [41;43;51;55] found that when the wildtype AA x_i mutated to another AA $x_i' \in \mathcal{X}$, the predicted probability $p(x_i'|x_{-i})$ (or its normalized version to the wildtype probability) is

a reasonable proxy of the mutated sequence's fitness $y_{x_i'}$. Now, if we are given the fitness scores of some mutations at site i as labeled data, we could refine the original distribution $p(x_i|x_{-i})$ to become more consistent with the fitness values. For two substitutions x_i' and x_i'' on the same position where $y_{x_i'} > y_{x_i''}$, we want the calibrated distribution p' to preserve the same fitness ordering in the probability space, i.e., $p'(x_i'|x_{-i}) > p'(x_i''|x_{-i})$. This calibration steers the pre-trained pLM to be more consistent with the fitness data while still respecting its pre-learned evolutionary patterns.

3.2 Contrastive Fitness Learning

We now introduce our novel fine-tuning algorithm, dubbed contrastive fine-tuning, to train a pLM-based model for the low-N learning of the protein fitness landscape. Formally, in our protein fitness prediction problem, we are given i) a pre-trained pLM f_{θ_0} parameterized by θ_0, which predicts the probability $p_0(x_i|x_{-i})$, and ii) a small (low-N) set of functionally characterized variants $\{\boldsymbol{x}^{(k)}, y^{(k)}\}_{k=1}^{N}$ of a particular protein, where $\boldsymbol{x}_k$'s are the variant sequences and y_k's are the experimentally measured fitness values. Our goal is to calibrate the pre-trained pLM f_{θ_0}, through fine-tuning on the low-N fitness data, into an enhanced model f_θ for low-N fitness prediction. In this work, we selected ESM-1v[40] as the pre-trained pLM f_{θ_0} for its effective zero-shot prediction performance in fitness prediction[40;43], but other pLMs[26–28] can also be used. The neural network architecture of f_θ mirrored that of f_{θ_0}, with the parameters θ initialized using the pre-trained weights θ_0.

From pLM to fitness prediction. Distinct from conventional pLM fine-tuning approaches (e.g., full or top-layer fine-tuning, Figs 1c-d), ConFit maintains the pre-training objective (i.e., predicting probability $p_\theta(x_i|x_{-i})$) in fine-tuning, rather than fine-tuning the pLM to directly predict the fitness values $\hat{y}$ using a regression objective. As discussed in Sec. 3.1, this approach circumvents the drastic parameter changes to accommodate the new learning task and output space.

In lieu of directly estimating fitness values, ConFit extracts fitness proxy from the conditional probability $p_\theta(x_i|x_{-i})$ predicted by pLM f_θ. This is informed by prior research[40;43;51], which demonstrates the effectiveness of using such probabilities in predicting AA mutation effects. We quantify the impact of a given mutation by comparing its pLM portability with the reference probability of the wild type. Specifically, let $\boldsymbol{x}^{\mathrm{MT}}$ and $\boldsymbol{x}^{\mathrm{WT}}$ denote the mutant and wild-type sequences, respectively, and M the mutated sites. The effect of a substitution $x_i^{\mathrm{WT}} \to x_i^{\mathrm{MT}}$ is quantified by the log probability ratio at this site: $\log p_\theta(x_i^{\mathrm{MT}}|x_{-M}) - \log p_\theta(x_i^{\mathrm{WT}}|x_{-M})$. Consequently, we aggregate the mutational effects to approximate the evolutionary plausibility of the mutant sequence:

$$\hat{y}_\theta(\boldsymbol{x}^{\mathrm{MT}}) = \sum_{i \in M} \log p_\theta(x_i^{\mathrm{MT}}|x_{-M}) - \log p_\theta(x_i^{\mathrm{WT}}|x_{-M}), \tag{1}$$

Previous work[20;37;41;43] found that this score is an informative proxy of the fitness of mutant $\boldsymbol{x}^{\mathrm{MT}}$.

Contrastive fine-tuning. To fine-tune f_θ for fitness prediction, instead of re-learning a model to minimize the MSE between predicted and true fitness values, we propose to *calibrate* the pLM probability distribution $p_\theta(x_i|x_{-M})$, adhering to the original pre-training objective of likelihood maximization. Our calibration strategy is inspired by contrastive learning, an ML strategy that compares multiple outputs of a model and encourages them to be ordered according to some criteria. In our case, we aim for a calibrated model where the likelihood-derived evolutionary scores align with the experimental fitness values: given two samples $(\boldsymbol{x}^+, y^+)$ and $(\boldsymbol{x}^-, y^-)$ with $y^+ > y^-$, we want to have $\hat{y}_\theta(\boldsymbol{x}^+) > \hat{y}_\theta(\boldsymbol{x}^-)$. This objective can be described by a loss function defined based on a classic probability model known as Bradley–Terry (BT) model[56], which we term the BT loss:

$$\mathcal{L}_{\mathrm{cal}} = \sum_{y^{(i)} > y^{(j)}} \log \left[1 + \exp\left(-[\hat{y}_\theta(\boldsymbol{x}^{(i)}) - \hat{y}_\theta(\boldsymbol{x}^{(j)})]\right)\right] \tag{2}$$

This loss function encourages the model to rank predicted fitness scores in the correct order relative to the ground truth. Although ConFit is not directly trained to predict the precise *magnitude* of fitness values, its prediction of fitness *rankings* is particularly valuable in the context of protein engineering, where ML models are often employed to efficiently rank a set of candidate variants *in silico*, thereby prioritizing the most promising variants for further experimental screening. Moreover, employing pairs from the N samples to compute the BT loss increases the effective training size from $O(N)$ to $O(N^2)$, further alleviating the data scarcity. Several other (scaled) ranking losses [57] can be explored in future work in place of $\mathcal{L}_{\mathrm{cal}}$. A similar

BT loss was also used in recent studies for protein sequence design[58] and fitness epistasis learning[59].

Regularization. To better avoid catastrophic forgetting, we further introduce another Kullback–Leibler (KL) divergence-based regularization loss to prevent the calibrated distribution p_θ deviating too far away from the pre-trained distribution p_{θ_0}:

$$\mathcal{L}_{\text{reg}} = \sum_i p_\theta(x_i^{\text{MT}}|x_{-i}^{\text{MT}}) \log \frac{p_\theta(x_i^{\text{MT}}|x_{-i})}{p_{\theta_0}(x_i^{\text{MT}}|x_{-i}^{\text{MT}})} \tag{3}$$

Our final loss function is a linear combination of the calibration loss and the KL regularizer: $\mathcal{L} = \mathcal{L}_{\text{cal}} + \lambda\mathcal{L}_{\text{reg}}$, where λ is a coefficient that balances the two losses.

Remarks: Our fine-tuning algorithm is partially inspired by a key technique known as Reinforcement Learning with Human Feedback (RLHF)[60] that enables ChatGPT to generate high-quality texts. In ChatGPT, human feedback about generation quality (e.g., text A is better than text B) is used to fine-tune GPT to align with human preferences. Similarly, in ConFit, we use experimental fitness data (e.g., mutation A leads to higher fitness than mutation B) to fine-tune a pLM into a low-N fitness prediction model.

3.3 Efficient Fine-Tuning with Low-Rank Reparameterization

The pre-trained pLM used in ConFit (ESM-1v) has 650 million parameters, and updating all parameters during our contrastive fine-tuning can be computationally expensive. We thus employed Low-Rank Adaptation (LoRA)[61], a parameter-efficient fine-tuning (PEFT) method to reduce the number of updated parameters. LoRA factorizes the weight updates to each neural network layer in ESM1v, denoted as a matrix $\Delta W \in \mathbb{R}^{d\times d'}$, into two low-rank matrices $\Delta W = UV$ where $U \in \mathbb{R}^{d\times r}$, $V \in \mathbb{R}^{r\times d'}$, and the rank $r \ll \min(d, d')$, thereby reducing the number of trainable parameters from $O(d \times d')$ to $O((d + d') \times r)$. In our experiment, we applied LoRA to ESM-1v with rank $r = 8$, reducing the effective number of parameters to 1.35 million, a 99.79% reduction compared to the 650 million parameters in the full ESM-1v model (Supplementary Information).

In addition to its advantages of computational efficiency, the reduction in trainable parameters holds significant implications for ConFit's adaptation to make protein-specific fitness predictions, especially in low-data scenarios. Recent studies in ML suggested that updates to language model weights often exhibit a low 'intrinsic rank'[61] and PEFT methods enable superior performance in few-shot learning tasks compared to conventional full-parameter fine-tuning methods[61–65]. While the vast capacity of the 650 million parameters enables ESM-1v to capture complex patterns in protein fitness landscapes, it could be over-parameterized when dealing with limited fitness data. Fine-tuning the entire model in such cases risks catastrophic forgetting. We thus leveraged LoRA to exploit the low-rank structure of pLM parameters and achieve efficient adaptation to new proteins using low-N data.

3.4 Enhancing Fitness Prediction with MSA Context Retrieval

While the pre-trained pLM used in ConFit captured global evolutionary patterns across natural protein sequences, when predicting fitness for a protein of interest, we further retrived its evolutionary-related sequences (homology) to reinforce the local evolutionary contexts specific to this protein.

Specifically, starting with the wildtype sequence of the protein, we searched in the UniProt protein sequence database for its homology and built a multiple sequence alignment (MSA). Next, we fit DeepSequence[20], a variational autoencoder (VAE)-based density model, on the MSA to estimate sequence likelihood $p_{\text{MSA}}(\boldsymbol{x})$. Different from the pLM sequence likelihood estimated from all natural sequences, the MSA-based sequence likelihood p_{MSA} here was only fit on the target protein's MSA sequences, thus capturing the local evolutionary contexts specific to this protein. Similarly, DeepSequence used the log-odd ratio to predict the fitness of a variant: $\hat{y}_{\text{MSA}}(\boldsymbol{x}^{\text{MT}}) = \log p_{\text{MSA}}(\boldsymbol{x}^{\text{MT}}) - \log p_{\text{MSA}}(\boldsymbol{x}^{\text{WT}})$, which have been shown as competitive unsupervised fitness predictor[20;22]. A recent study leveraging both pLM and retrieval of MSA sequences achieved state-of-the-art performance on zero-shot protein fitness prediction[42].

In ConFit, we refined the pLM-based fitness prediction $\hat{y}_\theta(\boldsymbol{x})$ by fusing it with the MSA-based prediction $\hat{y}_{\text{MSA}}(\boldsymbol{x})$: $\hat{y}(\boldsymbol{x}) = \alpha\hat{y}_\theta(\boldsymbol{x}) + (1 - \alpha)\hat{y}_{\text{MSA}}(\boldsymbol{x})$, where α is a weighting factor. We empirically determined the value as $\alpha = 0.8$ by inner-loop cross-validation. Note that ConFit integrates multi-scale evolutionary

contexts to predict protein fitness, including the global scale (natural protein sequences), intermediate scale (MSA sequences), and the most specific scale (protein-specific fitness data).

3.5 Implementation Details

ESM Ensemble. The ESM-1v study[40] released five sets of pre-trained model weights, all based on the same model architecture but initialized with different random seeds. Following their approach, we average the outputs of the five models to derive the fitness prediction $\hat{y}_\theta$ (Eq. 1) (Supplementary Information).

Training details and hyperparameter tuning. We trained ConFit on 4 NVIDIA A40 GPUs using the Adam optimizer and a cosine annealing scheduler, which reduced the learning rate from an initial rate to a minimal value. To mitigate the risk of overfitting, we early-stopped the training if the validation performance was not improved. We tuned the hyperparameters using an inner-loop validation data split within the training set, and the choices of hyperparameters were listed in (Supplementary Information).

4 Results

We conducted multiple evaluation experiments to demonstrate ConFit's low-N prediction ability for protein fitness and its utility in navigating protein fitness landscapes for protein engineering applications.

4.1 Experimental Settings

Datasets: To benchmark protein fitness prediction, we downloaded 34 fitness datasets across 27 proteins generated from deep mutational scanning (DMS) or random mutagenesis studies, following the choice of a prior work[38] (Supplementary Information). These datasets were originally curated by the DeepSequence study[20] and supplemented by datasets from other mutagenesis studies[66;67]. The fitness-labeled variant sequences range from single or double mutants to more extensively mutated sequences[67].

Data split: For each fitness dataset, we withheld 20% of randomly sampled data as test set, with the remaining 80% data used as the training library. To simulate varying degrees of low-N scenarios, we created the training set by randomly sampling N =48, 96, 168, or 240 samples for the training library. These particular sizes were chosen to reflect the common dimensions of laboratory well plates used in experimental assays. Each training set size was tested across 10 independent trials to ensure reproducibility.

Baseline methods: ConFit was compared to 13 leading sequence-based protein fitness prediction methods, including supervised and unsupervised models (Supplementary Information). For supervised methods, we included 'augmented models' (Augmented VAE and Augmented EVmutation) proposed by Hsu et al.[37] which combine amino acid sequences and evolutionary density scores as input features, and are recognized for their state-of-the-art supervised fitness prediction performances[37;38]. The comparison also included eUniRep, a model specifically tailored for low-N applications[36]. We further included 10 top-ranked unsupervised models from the ProteinGym benchmark for zero-shot fitness prediction[42], covering a wide range of modeling techniques, including language models (ESM-1b[27], ESM-1v[40], UniRep[25], WaveNet[50], MSA Transformer[68]), Potts models (e.g., EVmutation[9]), latent variable models (e.g., EVE[22], DeepSequence[20]), and hybrid models (e.g., Tranception[42], TranceptEVE[69]. Given our focus on sequence-based predictions, recent fitness models requiring structural data inputs[32;38] were not included in our comparison.

4.2 Accurate Prediction of Protein Fitness

In our first experiment, we evaluated the accuracy of the fitness prediction of ConFit. We restricted the training set size to include only $N = 240$ variants randomly sampled from the training library. In contrast to prior studies[34] where methods were trained on copious amounts of training samples (e.g., 80% of data), here we limited the training data to a mere $N = 240$ variants randomly sampled from the training library (14% data on average per datasets), which simulated a more realistic data-scarce scenario in protein engineering.

We first compared ConFit to leading unsupervised methods, including ESM1v, EVE, EVmutation, and TranceptEVE. These methods fit generative sequence density models on general natural sequences or specific homologous sequences to estimate the likelihood of a protein sequence and predict the mutation effects by

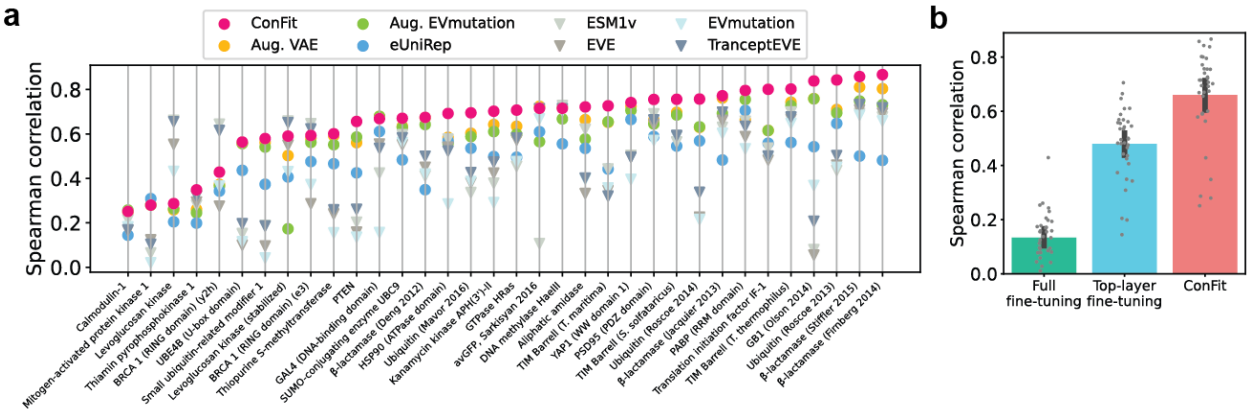

Figure 2: Comparisions between ConFit and other methods for protein fitness prediction. (**a**) Benchmark results on using 34 mutagenesis datasets. Supervised methods were shown as dots and unsupervised methods as triangles. Supervised models were trained on randomly sampled $N = 240$ samples. Each dot represents the average Spearman correlation over 10 repetitions. (**b**) Comparisons between ConFit and conventional fine-tuning strategies. Bar plots represent the mean±SD spearman correlations over the 34 fitness datasets.

comparing the log-odds ratio between the mutated and wild-type sequences. We found that, despite the small training set size, ConFit effectively harnessed information from the $N = 240$ samples and improved the fitness prediction accuracy without overfitting (Fig 2a). On average, ConFit achieved a mean Spearman correlation of 0.66 across all datasets, outperforming the unsupervised methods with clear margins on 29/34 datasets. In particular, compared to ESM1v – the non-finetuned counterpart of its base model – ConFit substantially improved the prediction accuracy (median difference of Spearman correlation $\Delta\rho = 0.23$; Fig 2b), suggesting the effectiveness of our contrastive fine-tuning algorithm. ConFit also significantly outperformed TranceptEVE (median $\Delta\rho = 0.19$; Fig S1), the best predictor as evaluated by the ProteinGym fitness prediction benchmark [42], and six additional unsupervised methods (Fig. S2).

We then compared ConFit to three supervised methods, including Augmented VAE, Augmented EVmutation, and eUniRep [36]. Existing studies demonstrated these methods are effective low-N fitness predictors[36;37]. We observed that ConFit was still in a leading position when compared to these supervised baselines, achieving the maximal Spearman scores on 29/34 datasets (Fig 2a). Even on the remaining 5/34 datasets where ConFit was not the top performer, its Spearman scores were very comparable to the top methods, Augmented VAE or Augmented EVmutation (mean $\Delta\rho = -0.01$; Fig S1). Moreover, ConFit achieved a 38% increase in Spearman scores compared to eUniRep, a pLM using the top-layer fine-tuning strategy, and a substantial 400% improvement compared to the full fine-tuning strategy (Fig. 2b). This gap affirmed our discussions in Sec. 2: full pLM fine-tuning tends to overfit, and while top-layer fine-tuning mitigates this risk, it can sacrifice predictive precision as it may not fully capture the complex features in protein fitness landscapes. In contrast, ConFit better characterized the fitness landscape with our contrastive fine-tuning strategy, which takes full advantage of pLMs by fine-tuning the entire model while preventing overfitting.

Overall, the improvements over state-of-the-art methods in this evaluation demonstrated the effectiveness of ConFit in adapting pre-trained pLMs for protein fitness prediction.

4.3 Low-N Learning of Fitness Landscape

Having evaluated the fitness prediction performance of ConFit across 34 DMS datasets in comparison to existing methods, we proceed to specifically assess the low-N learning capability of ConFit. We systematically varied the size of the training set by randomly sampling $N = 48, 96, 168, 240$ variants from the non-test data.

We first observed that, as expected, ConFit's prediction accuracy increased with increasing labeled training data (Fig. 3a). Notably, using only $N = 48$ training samples, ConFit predicted protein fitness at a Spearman correlation of 0.56, only 15% lower than its performance at $N = 240$, suggesting ConFit can effectively adapt a general pLM to accurately characterize a protein-specific fitness landscape even using an extreme low-N fitness dataset. The results also suggested that the MSA context retrieval consistently

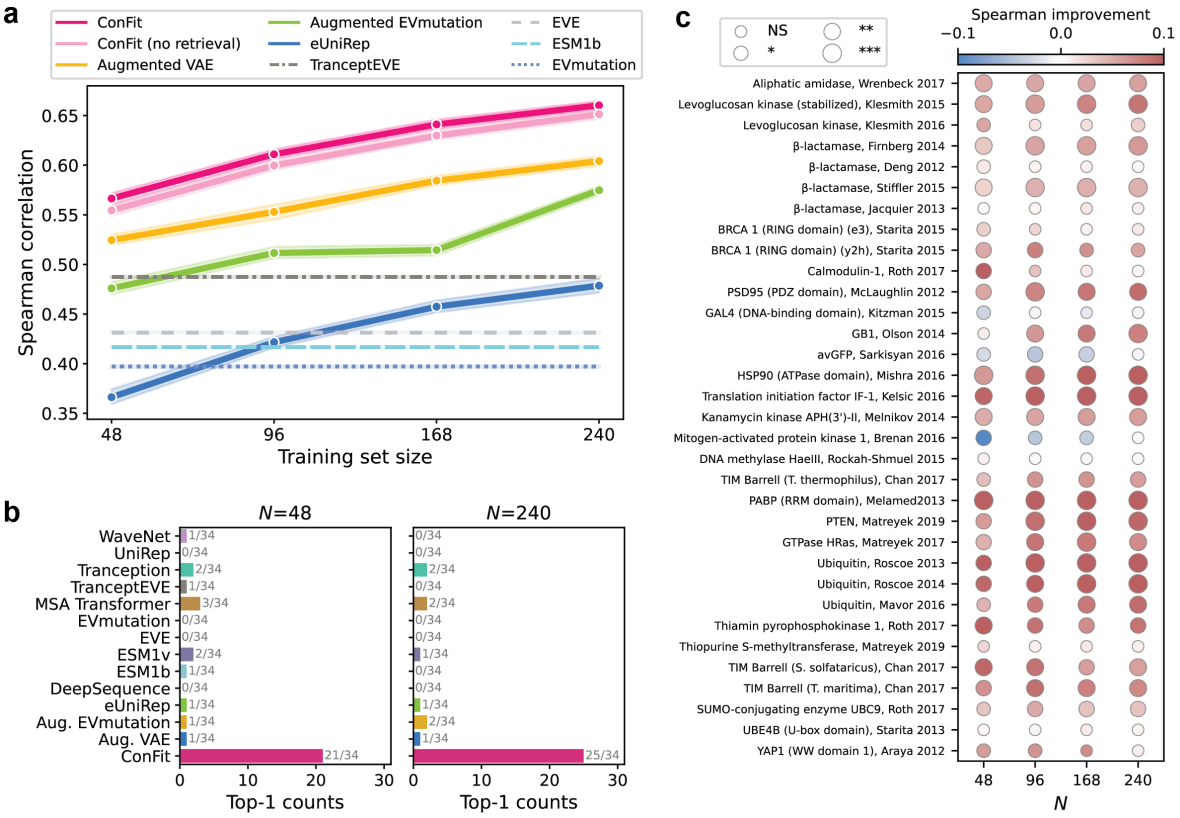

Figure 3: Evaluations on low-N learning of protein fitness. (**a**) Prediction performance of methods trained on N =48, 96, 168, or 240 samples. Solid lines and error bands indicate the mean±SD of Spearman correlations achieved by supervised methods. Performances of unsupervised methods are shown in dashed lines. (**b**) The frequency of each method ranked as the best across 34 datasets for training set sizes 48 and 240. (**c**) Detailed comparison between ConFit and Augmented VAE. The heatmap shows the improvements in Spearman correlation achieved by ConFit for each dataset and each training set size N. The dot size indicates the statistical significance. NS: not significant; *: $P \leq 0.05$; **: $P \leq 0.01$; ***: $P \leq 0.001$.

enhanced ConFit's performance across all training data sizes (Fig. 3a).

Compared to baseline methods, ConFit consistently outperformed three supervised methods and ten unsupervised methods across various training data sizes (Figs. 3a and S3). The performance improvements achieved by ConFit in low-N fitness learning were also substantial: for training data sizes $N = 48, 96, 168, 240$, ConFit achieved the top-1 Spearman correlation on 21, 24, 26, and 25 out of the 34 DMS datasets, respectively (Figs. 3b and S4), while second best baseline ranked at the top for only 3/34 datasets. We further analyzed the prediction performances of ConFit and Augmented VAE, the state-of-the-art supervised method for low-N fitness prediction[37]. ConFit outperformed Augmented VAE across all training sizes N on 28/34 DMS datasets and outperformed for at least one N value for 31/34 DMS datasets, and the majority of the improvements were statistically significant (Fig 3c).

Together, these results demonstrate the superior low-N fitness learning ability of ConFit. ConFit's performance gains stem from our novel contrastive fine-tuning algorithm that prevents overfitting without compromising prediction accuracy. Our approach significantly advances the current paradigm of low-N protein fitness prediction. The prevailing strategy employed by prior methods to achieve low-N fitness learning is to use informative pLM embeddings or evolutionary scores derived from unsupervised models as input, while keeping the predictive model in the simplest form (e.g., linear regression)[36;37]. The rationale behind this strategy is that models with fewer parameters are less prone to overfitting on low-N data. Although not overfitting, this strategy compromises prediction accuracy due to the limited effectiveness of the predictive models. ConFit, however, completely changed this paradigm through the contrastive fine-

tuning of pLMs – the pLM in ConFit is trainable, rather than being fixed as in previous methods[36;37], and carefully tailored through the contrastive fine-tuning algorithm to better modeling the complex sequence-fitness relationships, boosting the prediction accuracy without overfitting.

4.4 Extrapolation from Single to Higher-order Mutants

A critical metric for evaluating protein fitness prediction methods is their generalizability from lower-order mutants to higher-order mutants, which is vital for guiding protein engineering to discover novel higher-order mutants based on the fitness data of lower-order mutants. Among the 34 fitness datasets, we selected three—pertaining to avGFP, GB1, and PABP proteins—that included high-order mutants and conducted an evaluation where the model trained on single-mutation data is used to predict fitness for high-order mutants (Fig 4a). To further challenge the evaluated methods, we simulated the low-N scenarios with limited training data size $N = 48, 96, 168, 240$. We found that ConFit exhibited an outstanding extrapolation performance, with substantially improved or competitive performance as compared to other supervised baselines (Fig 4b). With increasing N, ConFit also increased in prediction accuracy and the performance gap over other methods.

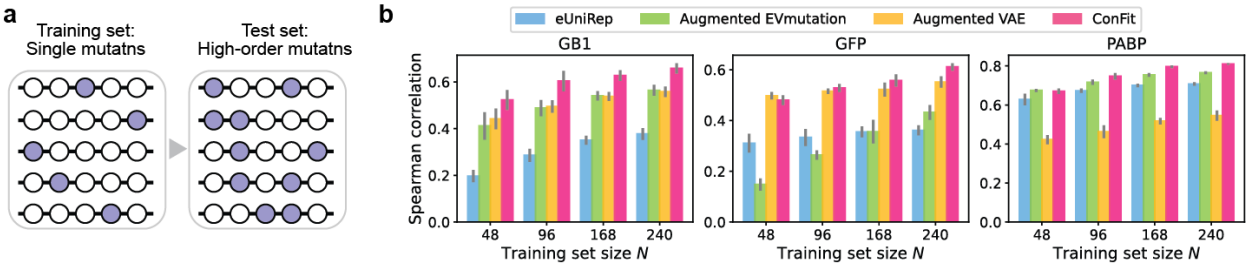

Figure 4: Extrapolation performance of fitness prediction on multi-mutation variants. (**a**) Each method was trained using the fitness data of single mutants data and then tested on higher-order mutants. (**b**) Comparisons between ConFit and three supervised baseline methods on three proteins (GB1, GFP, and PABP). Randomly sampled N =48, 96, 168, or 240 single mutants were used as training samples.

The strong extrapolation performance of ConFit is particularly useful for comprehensively characterizing the protein fitness landscapes. Current DMS studies often only sample point-mutation variants or variants within a local sequence space, mainly due to the extensive experimental efforts associated with the generation and analysis of higher-order variants. ConFit thus offers an effective solution to extrapolate from low-order mutants to identify higher-order functional mutants in protein engineering.

4.5 Navigating Fitness Landscape with Active Learning

Given the outstanding low-N learning capability and extrapolation performance of ConFit, we next demonstrated its utility in protein engineering using an active learning experiment, in which the goal is to strategically select training variants such that better fitness prediction performance can be achieved with less data. Active learning simulates the "prediction–acquisition–screening–retraining" loop in ML-guided protein engineering applications: scientists iterate between ML prediction and experimental screening for multiple rounds; in each round, the ML model proposes promising variants for screening, and the screened fitness data are then utilized to retrain the ML model, which in turn informs the subsequent cycle of predictions.

We evaluate ConFit's efficiency for landscape exploration on three proteins (GB1, PTEN, and UBE4B) with various definitions of fitness (binding, stability, and Ubiquitin ligase activity, respectively). We again set aside 20% of the data in each dataset as test data, reserving the remaining 80% as a training library from which we can draw samples. We initiated the active learning loop by training ConFit on $N = 48$ randomly sampled variants. In each subsequent round, ConFit predicted the fitness for all test variants and prioritized 48 variants from them, which were then added to the training set with their ground-truth fitness values for retraining ConFit. The prioritization was based on a scoring function called upper confidence bound (UCB), defined as $s(\boldsymbol{x}) = \hat{y}(\boldsymbol{x}) + \beta\sigma(\hat{y}(\boldsymbol{x}))$, where $\hat{y}(\boldsymbol{x})$ is ConFit's fitness prediction for sequence $\boldsymbol{x}$, and $\sigma(\hat{y}(\boldsymbol{x}))$ quantifies the prediction uncertainty using the standard deviation of outputs from the five ESM-1v models within ConFit. The coefficient β balances the tradeoff between high predicted fitness (exploitation) and high

uncertainty (exploration). We set $\beta = 1$ in our experiments. For comparisons, we included a greedy scoring function that ranks variants based on the predicted fitness and a random scoring function that randomly samples variants from the training library.

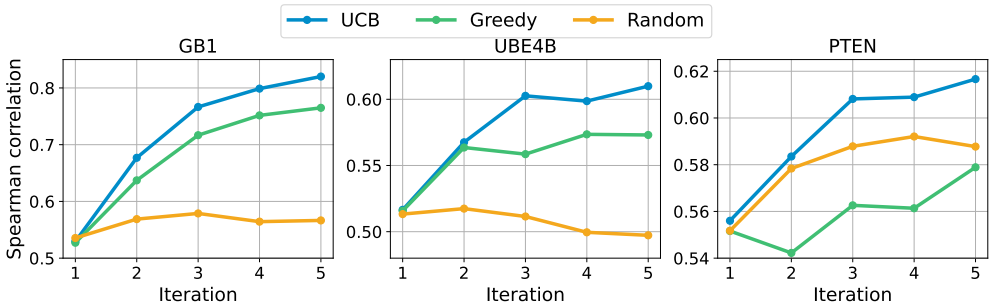

Figure 5: ConFit efficiently characterizes fitness landscapes with active learning. *In silico* active learning experiments were simulated on three protein fitness landscapes. In each iteration, variants prioritized by ConFit and a certain data acquisition function (Upper Confidence Bound, Greedy, or Random) were labeled with ground-truth fitness and used to augment the training set for model re-training.

Test results showed an increasing trend in ConFit's performance through five iterations of active learning on all three proteins (Fig 5). Interestingly, the random scoring function did not yield an increasing trend in performance, especially on GB1 and UBE4B. This indicated that the quality of the training set matters more than its quantity for improving the model's prediction accuracy. Since the protein sequence space contains a large fraction of non-functional sequences, the random strategy easily sampled non-functional sequences, which are less informative for model training. Additionally, the greedy sampling strategy was consistently outperformed by UCB, suggesting that only exploiting high-fitness regions in the sequence space may restrict the model's generalizability to unexplored regions. In contrast, using UCB, ConFit prioritized not only variants likely to deliver high fitness but also those that can best address the model's uncertainty for improving prediction accuracy.

The results also demonstrated the advantages of iterative data acquisition and model refinement. For example, on the UBE4B protein, ConFit was progressively refined through active learning using an cumulative number of 240 variants (48 per round $\times$ 5 rounds) and culminated in a Spearman correlation of 0.609 (Fig 5, UBE4B). However, when trained on the same amount of samples in a single training session, ConFit only achieved a Spearman correlation of 0.497 (Fig 5, UBE4B). This suggested that the accurate low-N fitness predictions of ConFit enable a feedback mechanism between ML models and experiments, which leads to more efficient navigation of protein fitness landscapes given the same experiment budget.

5 Discussion

We presented ConFit, an ML method for low-N protein fitness prediction through contrastive fine-tuning of pLMs. Many state-of-the-art ML solutions for protein science problems now rely on pLMs since pLMs capture the implicit evolutionary, structural, and biophysical constraints in protein sequences, thanks to their large model sizes and training data sizes. However, it is challenging to adapt the general pre-trained pLMs to problem-specific predictive models for protein biology tasks because large pLMs typically tend to overfit on small-size task-specific training data. Observing that ML fitness prediction models are often used in protein engineering to rank variants for prioritizing promising ones for screening, we proposed a contrastive fine-tuning algorithm to reprogram a pre-trained pLM for predicting the relative order of variant fitness. This strategy makes the pLM's pre-training and fine-tuning objectives consistent, avoiding the catastrophic forgetting issues frequently observed in naive fine-tuning approaches. Extensive evaluations suggested that ConFit outperformed over ten existing ML methods for protein fitness prediction and excelled especially in small-data scenarios. We expect ConFit to serve as an efficient ML algorithm for low-N protein engineering. Moreover, our contrastive fine-tuning algorithm represents a new paradigm for fine-tuning pLM better than conventional approaches (full or top-layer fine-tuning) and can be extended to other problems in protein informatics.

References

[1] Aaron Chevalier, Daniel-Adriano Silva, Gabriel J Rocklin, Derrick R Hicks, Renan Vergara, Patience Murapa, Steffen M Bernard, Lu Zhang, Kwok-Ho Lam, Guorui Yao, et al. Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674):74–79, 2017.

[2] Loredano Pollegioni, Ernst Schonbrunn, and Daniel Siehl. Molecular basis of glyphosate resistance–different approaches through protein engineering. *The FEBS journal*, 278(16):2753–2766, 2011.

[3] Roger A Sheldon and Pedro C Pereira. Biocatalysis engineering: the big picture. *Chemical Society Reviews*, 46(10):2678–2691, 2017.

[4] SB Jennifer Kan, Xiongyi Huang, Yosephine Gumulya, Kai Chen, and Frances H Arnold. Genetically programmed chiral organoborane synthesis. *Nature*, 552(7683):132–136, 2017.

[5] H Kaspar Binz, Patrick Amstutz, and Andreas Plückthun. Engineering novel binding proteins from nonimmunoglobulin domains. *Nature biotechnology*, 23(10):1257–1268, 2005.

[6] Yajie Wang, Pu Xue, Mingfeng Cao, Tianhao Yu, Stephan T Lane, and Huimin Zhao. Directed evolution: methodologies and applications. *Chemical reviews*, 121(20):12384–12444, 2021.

[7] Frances H Arnold. Innovation by evolution: bringing new chemistry to life (nobel lecture). *Angewandte Chemie International Edition*, 58(41):14420–14426, 2019.

[8] Hashem A Shihab, Julian Gough, David N Cooper, Peter D Stenson, Gary LA Barker, Keith J Edwards, Ian NM Day, and Tom R Gaunt. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden markov models. *Human mutation*, 34(1):57–65, 2013.

[9] Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta PI Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

[10] Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 2011.

[11] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013.

[12] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[13] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[14] Sergey Ovchinnikov, Lisa Kinch, Hahnbeom Park, Yuxing Liao, Jimin Pei, David E Kim, Hetunandan Kamisetty, Nick V Grishin, and David Baker. Large-scale determination of previously unsolved protein structures using evolutionary information. *elife*, 4:e09248, 2015.

[15] Stefan Seemayer, Markus Gruber, and Johannes Söding. Ccmpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 2014.

[16] William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

[17] Jaclyn K Mann, John P Barton, Andrew L Ferguson, Saleha Omarjee, Bruce D Walker, Arup Chakraborty, and Thumbi Ndung'u. The fitness landscape of hiv-1 gag: advanced modeling approaches and validation of model predictions by in vitro testing. *PLoS computational biology*, 10(8):e1003776, 2014.

[18] Ryan R Cheng, Faruck Morcos, Herbert Levine, and José N Onuchic. Toward rationally redesigning bacterial two-component signaling systems using coevolutionary information. *Proceedings of the National Academy of Sciences*, 111(5):E563–E571, 2014.

[19] Matteo Figliuzzi, Hervé Jacquier, Alexander Schug, Oliver Tenaillon, and Martin Weigt. Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase tem-1. *Molecular biology and evolution*, 33(1):268–280, 2016.

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

[21] Xinqiang Ding, Zhengting Zou, and Charles L Brooks III. Deciphering protein evolution and fitness landscapes with latent space models. *Nature communications*, 10(1):5644, 2019.

[22] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

[23] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

[24] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[25] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[26] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin 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.

[27] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[28] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[29] 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*, pages 1–8, 2023.

[30] Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.

[31] Claire N Bedbrook, Kevin K Yang, J Elliott Robinson, Elisha D Mackey, Viviana Gradinaru, and Frances H Arnold. Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics. *Nature methods*, 16(11):1176–1184, 2019.

[32] Sam Gelman, Sarah A Fahlberg, Pete Heinzelman, Philip A Romero, and Anthony Gitter. Neural networks to learn protein sequence–function relationships from deep mutational scanning data. *Proceedings of the National Academy of Sciences*, 118(48):e2104878118, 2021.

[33] Drew H Bryant, Ali Bashir, Sam Sinai, Nina K Jain, Pierce J Ogden, Patrick F Riley, George M Church, Lucy J Colwell, and Eric D Kelsic. Deep diversification of an aav capsid protein by machine learning. *Nature Biotechnology*, 39(6):691–696, 2021.

[34] Yunan Luo, Guangde Jiang, Tianhao Yu, Yang Liu, Lam Vo, Hantian Ding, Yufeng Su, Wesley Wei Qian, Huimin Zhao, and Jian Peng. Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications*, 12(1):5743, 2021.

[35] Jack W Scannell and Jim Bosley. When quality beats quantity: decision theory, drug discovery, and the reproducibility crisis. *PloS one*, 11(2):e0147215, 2016.

[36] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

[37] Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature biotechnology*, 40(7):1114–1122, 2022.

[38] Yuchi Qiu and Guo-Wei Wei. Persistent spectral theory-guided protein engineering. *Nature Computational Science*, 3(2):149–163, 2023.

[39] Brian Hie, Ellen D Zhong, Bonnie Berger, and Bryan Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021.

[40] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex 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.

[41] Brian L Hie, Kevin K Yang, and Peter S Kim. Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins. *Cell Systems*, 13(4):274–285, 2022.

[42] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[43] Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 2023.

[44] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[45] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023.

[47] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann,

A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.

[48] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311, 2022.

[49] Vineet Thumuluri, Hannah-Marie Martiny, Jose J Almagro Armenteros, Jesper Salomon, Henrik Nielsen, and Alexander Rosenberg Johansen. Netsolp: predicting protein solubility in escherichia coli using language models. *Bioinformatics*, 38(4):941–946, 2022.

[50] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403, 2021.

[51] Nadav Brandes, Grant Goldman, Charlotte H Wang, Chun Jimmie Ye, and Vasilis Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, pages 1–11, 2023.

[52] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[53] Daniel M Weinreich, Yinghong Lan, C Scott Wylie, and Robert B Heckendorn. Should evolutionary geneticists worry about higher-order epistasis? *Current opinion in genetics & development*, 23(6):700–707, 2013.

[54] David M McCandlish, Etienne Rajon, Premal Shah, Yang Ding, and Joshua B Plotkin. The role of epistasis in protein evolution. *Nature*, 497(7451):E1–E2, 2013.

[55] Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems*, 35:38873–38884, 2022.

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

[57] Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*, 2022.

[58] Alvin Chan, Ali Madani, Ben Krause, and Nikhil Naik. Deep extrapolation for attribute-enhanced generation. *Advances in Neural Information Processing Systems*, 34:14084–14096, 2021.

[59] David H Brookes, Jakub Otwinowski, and Sam Sinai. Contrastive losses as generalized models of global epistasis. *arXiv preprint arXiv:2305.03136*, 2023.

[60] 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.

[61] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[62] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[63] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[64] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[65] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023.

[66] C Anders Olson, Nicholas C Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current biology*, 24(22):2643–2651, 2014.

[67] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

[68] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[69] Pascal Notin, Lood Van Niekerk, Aaron W Kollasch, Daniel Ritter, Yarin Gal, and Debora S Marks. Trancepteve: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, pages 2022–12, 2022.

[70] Nathan J Rollins, Kelly P Brock, Frank J Poelwijk, Michael A Stiffler, Nicholas P Gauthier, Chris Sander, and Debora S Marks. Inferring protein 3d structure from deep mutation scans. *Nature genetics*, 51(7):1170–1176, 2019.

[71] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.