Parameter-Efficient Protein Fitness Prediction: Combining Contrastive Sequence Learning and Structure-Aware Fusion

Sai Khadloya
International Institute of Information Technology, Hyderabad
sai.khadloya@students.iiit.ac.in

Arihant Jain
International Institute of Information Technology, Hyderabad
arihant.j@research.iiit.ac.in

Abstract

Protein fitness prediction, crucial for understanding protein function and guiding protein engineering, faces significant challenges in data-scarce (low-N) settings due to the high cost of experimental assays. Pre-trained protein language models (pLMs) offer powerful sequence representations, but naive fine-tuning on limited fitness data leads to catastrophic forgetting and overfitting. This paper explores two distinct yet complementary parameter-efficient approaches to tackle low-N protein fitness prediction on the ProteinGym benchmark.

First, we present **ConFit++**, an extension of the Contrastive Fitness Learning framework. ConFit++ reprograms a large pLM (ESM2) using a triplet-based contrastive loss that enforces relative fitness ordering in the latent space. It incorporates an attention-based module to explicitly model epistatic interactions between mutations and utilizes Low-Rank Adaptation (LoRA) and memory optimizations for efficient fine-tuning.

Second, we investigate a **Sequence and Structure Fusion** approach. This method leverages LoRA-fine-tuned ESM2 sequence embeddings and combines them with structural embeddings derived from AlphaFold-predicted Cα distance maps encoded by a shallow Convolutional Neural Network (CNN). A small Multi-Layer Perceptron (MLP) then regresses fitness from the concatenated sequence and structure features.

Both methods employ LoRA on ESM2 to drastically reduce trainable parameters and GPU memory requirements, enabling effective learning from limited data while preserving valuable pre-trained knowledge. We evaluate both approaches on subsets of the ProteinGym benchmark, demonstrating their superior performance in low-N regimes compared to various baselines. ConFit++ excels in accurate low-N sequence-based fitness prediction and epistasis modeling, while the Sequence and Structure Fusion approach highlights the complementary value of structural information in improving predictive accuracy.

1 Introduction

Predicting the functional consequences of amino acid substitutions is a fundamental problem in molecular biology and a core task in protein engineering. The relationship between a protein's sequence and its fitness (e.g., stability, activity, binding affinity) defines its fitness landscape, a

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

complex, high-dimensional space often characterized by non-linear interactions between mutations (epistasis) (1; 2). Deep mutational scanning (DMS) assays enable high-throughput measurement of fitness for thousands to millions of variants, providing invaluable data for mapping these landscapes (3; 4). However, running DMS experiments is costly and time-consuming, resulting in limited labeled data (low-N) for any specific protein or function.

Pre-trained protein language models (pLMs) like ESM2 (6), ESM-1v (7), and ProtT5 (8) have emerged as powerful tools for capturing evolutionary and biophysical principles from vast unlabeled protein sequence databases. Trained on objectives such as masked language modeling (MLM), these models learn contextualized representations of amino acids that correlate well with experimental fitness measurements, exhibiting notable zero-shot prediction capabilities (9). This suggests that pLMs encode a strong prior over functional sequence space, which could be leveraged for supervised fitness prediction.

However, directly fine-tuning these large pLMs (with parameters ranging from hundreds of millions to billions) on small, task-specific fitness datasets (low-N, e.g., $N \sim 10^2 - 10^3$) is challenging due to *catastrophic forgetting* (10). The model's pre-trained knowledge, essential for generalization, is rapidly overwritten by overfitting to the limited supervised signal. This necessitates parameter-efficient fine-tuning strategies.

Furthermore, accurately modeling protein fitness often requires capturing both sequence context and structural properties, as mutations exert their effects through changes in protein structure and dynamics. Epistatic interactions, where the effect of one mutation depends on others, are particularly challenging to predict from sequence alone but often arise from structural constraints or inter-residue contacts (12; 13).

To address these challenges, this paper presents two distinct yet complementary parameter-efficient approaches for low-N protein fitness prediction, leveraging the power of large pLMs while integrating different sources of information:

1. **ConFit++:** An extension of the Contrastive Fitness Learning framework that focuses on enhancing sequence-based fitness prediction by using a triplet-based contrastive objective to improve fitness ranking and incorporating an attention-based module to model epistasis, all within a parameter-efficient fine-tuning scheme using LoRA on ESM2.
2. **Sequence and Structure Fusion:** An alternative approach that explicitly combines sequence embeddings from a LoRA-tuned ESM2 with structural embeddings derived from AlphaFold-predicted 3D structures using a dedicated CNN encoder. A late-fusion MLP then predicts fitness from the combined representation.

Both methods utilize Low-Rank Adaptation (LoRA) (11) applied to ESM2 to drastically reduce the number of trainable parameters and computational requirements, making them feasible in resource-constrained settings. We evaluate both approaches on relevant subsets of the ProteinGym benchmark (14), demonstrating their effectiveness in low-N scenarios compared to existing methods.

The subsequent sections detail the background on pLMs and LoRA, describe the methodology of each proposed approach (**ConFit++** first, then the **Sequence and Structure Fusion** method), present their respective experimental results, and discuss conclusions and future work.

2 Background

2.1 Protein Language Models and Fitness Prediction

Protein language models (pLMs) are typically large Transformer networks trained on massive datasets of protein sequences. Given a sequence $\mathbf{x} = (x_1, \ldots, x_L)$, where $x_i \in \mathcal{A}$ is the amino acid at position i, the pLM learns a probability distribution $p_\theta(x_i \mid \mathbf{x}_{\setminus i})$ over the possible amino acids at a masked position i, conditioned on the unmasked residues $\mathbf{x}_{\setminus i}$. The model is parameterized by θ. This distribution reflects the model's learned understanding of which residues are evolutionarily plausible in a given sequence context.

The zero-shot prediction of mutation effects is often based on comparing the likelihood of the wild-type residue (x_i^{WT}) versus a mutant residue (x_i^{MT}) at a specific position i, given the wild-type

context:

$$\Delta \log p_\theta(x_i^{\text{WT}} \to x_i^{\text{MT}}) = \log p_\theta(x_i^{\text{MT}} \mid \mathbf{x}_{\setminus i}^{\text{WT}}) - \log p_\theta(x_i^{\text{WT}} \mid \mathbf{x}_{\setminus i}^{\text{WT}}) \tag{1}$$

For a multi-site mutant $\mathbf{x}^{\text{MT}}$ with mutations at positions M, a common additive zero-shot score is $\sum_{i \in M} \Delta \log p_\theta(x_i^{\text{WT}} \to x_i^{\text{MT}} | \mathbf{x}_{\setminus i}^{\text{WT}})$. While these zero-shot scores show correlations with fitness, they are often insufficient for precise prediction and fail to capture epistatic interactions.

2.2 ESM-2 Model

ESM-2 (6) (Evolutionary Scale Modeling) is a state-of-the-art pLM based on the Transformer encoder architecture. It is pre-trained on over 400 million sequences from the UniRef database. The model outputs contextualized embeddings $\mathbf{h}_1, \ldots, \mathbf{h}_L \in \mathbb{R}^D$ for each residue, where D is the embedding dimension. These embeddings capture rich structural and functional information. ESM-2 comes in various sizes, from 8M to 15B parameters. For parameter-efficient fine-tuning, using a moderately sized model like the 150M or 650M parameter variant is a common choice, balancing representational power with computational cost.

2.3 Low-Rank Adaptation (LoRA)

LoRA (11) is a parameter-efficient fine-tuning technique designed to adapt large pre-trained models without modifying all their parameters. It proposes that the weight updates during fine-tuning have a low intrinsic rank. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA freezes W_0 and introduces two trainable low-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$ is the low rank. The forward pass is computed as:

$$h = W_0 x + BAx \tag{2}$$

The parameter update $\Delta W = BA$ is added to W_0. During fine-tuning, only A and B are trained, reducing the number of trainable parameters from $d \times k$ to $d \times r + r \times k$. This significantly decreases GPU memory requirements and allows faster training, making fine-tuning large models feasible even on limited hardware. LoRA is typically applied to the weight matrices in the attention and feed-forward layers of Transformer models.

2.4 Protein Structure Prediction and Encoding

AlphaFold2 (15) predicts protein 3D structures from sequence with high accuracy, providing a powerful source of structural information when experimental structures are unavailable. A protein structure can be represented by its Cα–Cα distance map $D \in \mathbb{R}^{L \times L}$, where D_{ij} is the Euclidean distance between the Cα atoms of residues i and j. Distance maps or derived contact maps (thresholded distance maps) capture crucial spatial relationships between residues. These 2D representations can be processed by convolutional neural networks (CNNs) to extract fixed-length structural embeddings that encode patterns related to compactness, contacts, and secondary/tertiary structure.

3 Methodology: Two Approaches

We present two distinct methodological approaches for low-N protein fitness prediction, both leveraging ESM2 and LoRA for efficiency, but differing in how they incorporate fitness signal and potential structural information.

3.1 Method 1: ConFit++ - Contrastive Sequence Learning with Epistasis

ConFit++ extends the idea of Contrastive Fitness Learning to enhance sequence-based fitness prediction. It directly fine-tunes the pLM (ESM2 with LoRA) using a contrastive objective based on relative fitness and incorporates an explicit epistasis modeling component.

3.1.1 Protein Fitness Score from pLM Probabilities

ConFit++ estimates fitness based on the relative change in conditional probabilities assigned by the fine-tuned pLM. For a variant $\mathbf{x}$ derived from wild-type $\mathbf{x}^{\text{WT}}$ with mutations at positions M, the base

fitness score is:

$$\hat{y}_{\theta,\text{base}}(\mathbf{x}) = \sum_{i \in M} \left(\log p_\theta(x_i^{\text{MT}} \mid \mathbf{x}_{\setminus i}^{\text{WT}}) - \log p_\theta(x_i^{\text{WT}} \mid \mathbf{x}_{\setminus i}^{\text{WT}}) \right) \tag{3}$$

where θ are the parameters of the LoRA-fine-tuned ESM2. This formulation leverages the pLM's natural probabilistic output.

3.1.2 Triplet-Based Contrastive Training

To train the model effectively on limited supervised data while prioritizing the learning of relative fitness orderings, we adopt a **triplet ranking loss** framework. Unlike standard regression losses (e.g., Mean Squared Error) that penalize absolute deviations from ground truth values, the triplet loss focuses on preserving the *ordering* of fitness values, which is especially suitable when only relative fitness information is meaningful or robust in low-data regimes.

Each training instance is constructed as a triplet $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$, where:

- $\mathbf{x}^a$ is the **anchor** sequence.
- $\mathbf{x}^p$ is a **positive** example with known fitness $y^p \geq y^a$ (i.e., equally or more fit than the anchor).
- $\mathbf{x}^n$ is a **negative** example with $y^n < y^a$ (i.e., less fit than the anchor).

The training objective is to ensure that the predicted fitness of the positive example $\hat{y}_\theta(\mathbf{x}^p)$ is higher than that of the negative $\hat{y}_\theta(\mathbf{x}^n)$ by at least a fixed margin $\delta > 0$. This is formalized using the hinge-based triplet loss:

$$\mathcal{L}_{\text{triplet}} = \sum_{(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)} \max\left(0, \hat{y}_\theta(\mathbf{x}^n) - \hat{y}_\theta(\mathbf{x}^p) + \delta\right) \tag{4}$$

where $\hat{y}_\theta(\cdot)$ denotes the model's total predicted fitness score, including both additive and epistatic contributions (as described in later sections).

This objective function only incurs loss when the negative is ranked too closely to or above the positive. It is thus a sparse and directional signal, focusing learning on hard or misordered examples. The use of a margin δ introduces a buffer zone that enforces a meaningful separation between fitness scores, improving generalization.

Rationale: The protein fitness landscape is often rugged, with many sequences having similar but not identical fitness values. In low-N settings, accurately predicting absolute fitness is difficult and may be misleading. However, the relative *ranking* of variants is often preserved and more robust across datasets (9). The triplet loss explicitly optimizes for such ranking consistency. By learning a latent space in which fitter proteins are consistently scored higher than less fit ones, the model becomes better at prioritizing beneficial mutations and guiding protein engineering even in data-scarce regimes.

Moreover, this formulation aligns well with evolutionary principles, where selective pressure acts on fitness differences rather than absolute values, reinforcing the biological relevance of the training objective.

3.1.3 Attention-Based Epistasis Modeling

To account for non-additive epistatic interactions between mutations, we incorporate a dedicated **epistasis modeling module** built upon an attention mechanism applied to the **contextual embeddings** of the mutated residues obtained from the fine-tuned ESM2 model. For a given mutant protein sequence $\mathbf{x}$, let the set of mutated positions be denoted by M. We extract the corresponding final-layer contextual embeddings $\{\mathbf{h}_i\}_{i \in M}$ from the LoRA-adapted ESM2 model. These embeddings $\mathbf{h}_i \in \mathbb{R}^d$ capture the semantic and structural context of each residue within the entire protein sequence, effectively encoding the local and global biochemical environment.

These mutation-specific embeddings are then passed into a small **neural epistasis module**, whose purpose is to model the non-additive fitness effects caused by interdependencies between mutations. The module consists of a **multi-head self-attention layer**, followed by a feed-forward neural network

(FFN). The attention mechanism enables each mutated residue i to attend to all other mutated positions $j \in M$, producing a context-aware representation $\mathbf{z}_i$ for each position:

$$Q_i = \mathbf{h}_i W^Q \tag{5}$$

$$K_j = \mathbf{h}_j W^K \tag{6}$$

$$V_j = \mathbf{h}_j W^V \tag{7}$$

$$\alpha_{ij} = \frac{\exp(Q_i K_j^\top / \sqrt{d_k})}{\sum_{k \in M} \exp(Q_i K_k^\top / \sqrt{d_k})} \tag{8}$$

$$\mathbf{z}_i = \sum_{j \in M} \alpha_{ij} V_j \tag{9}$$

Here, $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are trainable projection matrices for the query, key, and value transformations, and d_k is the dimensionality of the key vectors.

The output vectors $\{\mathbf{z}_i\}_{i \in M}$, which encode inter-mutational interactions in an attention-weighted fashion, are aggregated (e.g., via mean pooling or learned weighted pooling) and passed through a feed-forward neural network ϕ_{epi}, which serves as a **learned function approximator for epistasis**. This network models the nonlinear mapping from contextual mutation embeddings to a scalar **epistatic correction term**, capturing deviations from purely additive effects:

$$s_{\text{epi}}(\{\mathbf{h}_i\}_{i \in M}) = \phi_{\text{epi}}(\{\mathbf{z}_i\}_{i \in M})$$

Finally, the overall predicted fitness score $\hat{y}_\theta(\mathbf{x})$ is computed by combining the base model's prediction $\hat{y}_{\theta,\text{base}}(\mathbf{x})$, which captures the additive contribution of individual mutations, with the learned epistatic correction:

$$\hat{y}_\theta(\mathbf{x}) = \hat{y}_{\theta,\text{base}}(\mathbf{x}) + s_{\text{epi}}(\{\mathbf{h}_i\}_{i \in M}) \tag{10}$$

Rationale: Traditional additive models assume independence between mutations, which is often violated in real proteins where mutations interact nonlinearly due to structural and functional constraints. By training a neural network ϕ_{epi} to operate on attention-augmented embeddings, we enable the model to learn complex, higher-order epistatic effects that arise from the co-dependence of mutations in spatially or functionally coupled regions of the protein. This results in more accurate modeling of the protein fitness landscape, especially in cases where interactions between residues significantly alter function.

3.1.4 Efficient Fine-Tuning with LoRA on ESM-2

The ESM2 model (150M parameters used in our experiments) is adapted using LoRA with rank $r = 4$ applied to the query and value projection matrices in all Transformer layers. This dramatically reduces the number of trainable parameters to approximately 310K, enabling fine-tuning with limited data and compute resources while preserving most of the pre-trained knowledge.

3.1.5 MSA Context Integration

To further enhance protein-specific evolutionary context, we integrate information from Multiple Sequence Alignments (MSAs). An MSA-based generative model (e.g., a VAE like DeepSequence) is trained on homologous sequences retrieved for the wild-type protein. This model provides a likelihood estimate $p_{\text{MSA}}(\mathbf{x})$ for any variant $\mathbf{x}$ within that protein family. The MSA-based score is:

$$\hat{y}_{\text{MSA}}(\mathbf{x}) = \log p_{\text{MSA}}(\mathbf{x}) - \log p_{\text{MSA}}(\mathbf{x}^{\text{WT}}) \tag{11}$$

The final ConFit++ prediction is a weighted combination:

$$\hat{y}(\mathbf{x}) = \alpha \hat{y}_\theta(\mathbf{x}) + (1 - \alpha) \hat{y}_{\text{MSA}}(\mathbf{x}) \tag{12}$$

where $\alpha \in [0, 1]$ balances the contribution of the pLM-based score and the MSA-based score.

3.1.6 Regularization

To prevent catastrophic forgetting, a KL divergence regularization term is added to the loss function, encouraging the fine-tuned model's output probabilities for wild-type residues at masked positions to stay close to the original pre-trained model's probabilities:

$$\mathcal{L}_{\text{KL}}(\theta, \theta_0) \approx \sum_{i=1}^{L} \text{KL}\left(p_\theta(x_i^{\text{WT}} | \mathbf{x}_{\setminus i}^{\text{WT}}) \,||\, p_{\theta_0}(x_i^{\text{WT}} | \mathbf{x}_{\setminus i}^{\text{WT}})\right) \tag{13}$$

The total objective function for training ConFit++ is:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{triplet}}(\theta) + \lambda \mathcal{L}_{\text{KL}}(\theta, \theta_0) \tag{14}$$

where λ is a regularization weight.

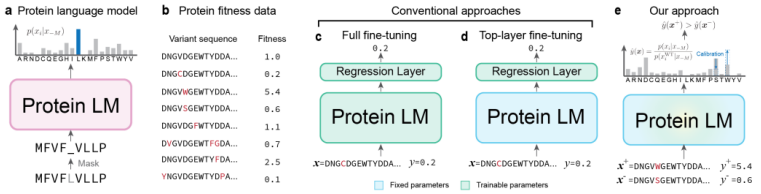

Figure 1: Overview of ConFit. (a) A protein language model (pLM) predicts the probability of amino acid at position given other unmasked positions. (b) Example of protein fitness data in which the variants of a wildtype (WT) sequence are experimentally characterized with fitness values. (c-d) Conventional 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.

Figure 1: Conceptual ConFit training pipeline. The ConFit++ pipeline builds upon this by incorporating ESM2, LoRA, triplet loss, and an attention-based epistasis module within the contrastive fine-tuning step, replacing the pairwise loss.

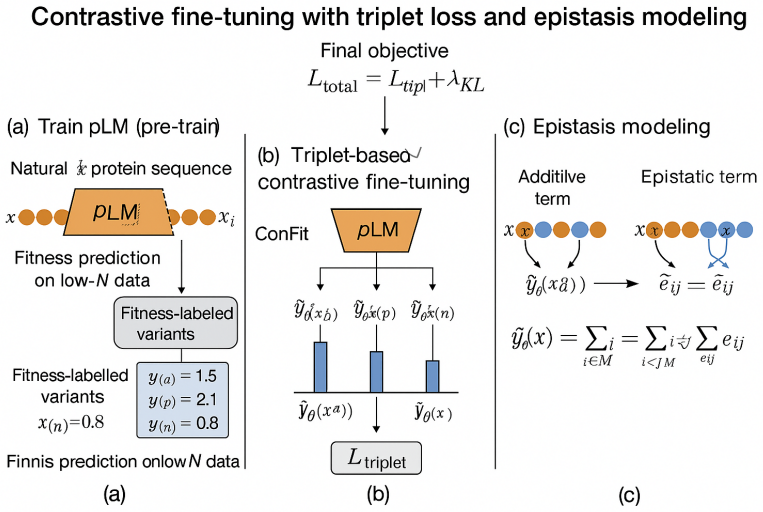

Figure 2: Illustrative overview of the contrastive fine-tuning process with triplet loss and the integration of an attention-based module for epistasis modeling within ConFit++. ESM2 serves as the backbone, adapted via LoRA.

3.2 Method 2: Sequence and Structure Fusion

Alternatively, we investigate a different approach that explicitly combines sequence and structural information in a late-fusion architecture. This method does not directly modify the pLM's probabilistic outputs but rather uses its fine-tuned embeddings as sequence features and integrates them with features derived from 3D structures.

3.2.1 Sequence Embedding with LoRA-tuned ESM2

Similar to ConFit++, the ESM2 model (650M parameters in this specific implementation) is fine-tuned using LoRA (rank $r = 4$). However, the fine-tuning objective here is typically a direct regression loss (e.g., MSE) on the fitness values, or a contrastive loss depending on the specific implementation details, but the key difference is *how* the final prediction is made. After fine-tuning, a fixed-length sequence embedding $\mathbf{e} \in \mathbb{R}^{d_e}$ for any variant $\mathbf{x}$ is extracted by global average pooling of the final layer residue embeddings:

$$\mathbf{e}(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^{L} \mathbf{h}_i^{\text{final}}(\mathbf{x}) \tag{15}$$

where $\mathbf{h}_i^{\text{final}}(\mathbf{x})$ is the contextual embedding for residue i of sequence $\mathbf{x}$ from the final Transformer layer.

To integrate structural information into the fitness prediction framework, we compute structural embeddings from AlphaFold2-predicted 3D protein models. Specifically, for a subset of representative variants (typically 10 per assay), 3D structures are predicted using AlphaFold2. From each predicted structure, we extract a **Cα–Cα distance matrix** $D \in \mathbb{R}^{L \times L}$, where L is the sequence length, and each entry D_{ij} represents the Euclidean distance between the Cα atoms of residues i and j.

To enhance numerical stability and emphasize local structural neighborhoods, we *clip* distances at a maximum value of 20 Å and then *normalize* the entire matrix to the interval $[0, 1]$, forming a contact-like 2D map. This results in a bounded representation that maintains spatial information while attenuating the effect of distant residue pairs.

Structural Encoder: A shallow Convolutional Neural Network (CNN) is employed to encode the 2D normalized distance matrix D. The CNN acts as a feature extractor that identifies spatial patterns associated with local packing and global folding motifs. The architecture typically consists of:

- Several 2D convolutional layers with small kernel sizes (e.g., 3×3), allowing for localized receptive fields;
- Nonlinear activation functions (e.g., ReLU) after each convolution;
- Batch normalization to stabilize and accelerate training;
- A global pooling layer (e.g., Global Average Pooling) that collapses the spatial dimensions and aggregates global structural features;
- A final linear (fully-connected) layer that projects the pooled features into a fixed-dimensional vector $\mathbf{s} \in \mathbb{R}^{d_s}$.

Formally, the structural embedding is computed as:

$$\mathbf{s}(D) = \text{Linear}(\text{GlobalAvgPool}(\text{CNN}(D))) \tag{16}$$

Rationale: The CNN operates over the 2D topology of the contact-like map, making it effective at learning spatially localized motifs that are crucial for protein folding and stability. Because it processes relative distance patterns rather than raw coordinates, the CNN encoder is *invariant to rigid-body transformations* (rotations and translations), a desirable property for structural analysis. This enables the model to learn features such as β-sheets, α-helices, or dense residue clusters that may be associated with functional or fitness-related traits. The global pooling and projection step ensures that the resulting structural embedding $\mathbf{s}(D)$ is a compact, permutation-invariant summary of the full 3D structure, suitable for downstream fusion with sequence-based representations.

Incorporating such structural priors is particularly valuable when epistatic interactions arise from spatially proximal residues that are distant in sequence but close in 3D space, allowing the model to go beyond purely sequential context.

3.2.2 Late-Fusion Regression

The sequence embedding $\mathbf{e}$ and the structural embedding $\mathbf{s}$ are concatenated to form a joint representation $[\mathbf{e} \parallel \mathbf{s}] \in \mathbb{R}^{d_e+d_s}$. This combined vector is fed into a small Multi-Layer Perceptron (MLP) to predict the final fitness score $\hat{y}$:

$$\hat{y} = W_2 \cdot \text{Dropout}(\text{ReLU}(W_1[\mathbf{e} \parallel \mathbf{s}] + \mathbf{b}_1)) + \mathbf{b}_2 \tag{17}$$

where $W_1 \in \mathbb{R}^{h \times (d_e+d_s)}$, $\mathbf{b}_1 \in \mathbb{R}^h$, $W_2 \in \mathbb{R}^{1 \times h}$, $\mathbf{b}_2 \in \mathbb{R}$, and h is the hidden layer size.

3.2.3 Training Procedure

The overall training framework is designed to integrate complementary sources of information—sequence-derived features from a fine-tuned protein language model and spatial features from 3D protein structures—into a unified predictive model for protein fitness.

Decoupled Fine-Tuning of ESM2: The sequence encoder is based on the pre-trained ESM2 protein language model, which is fine-tuned independently using *Low-Rank Adaptation (LoRA)* on the fitness prediction task. This approach ensures that only a small number of additional parameters (low-rank matrices) are trained, preserving the generalizable knowledge encoded in the original ESM2 while adapting it to task-specific signal. LoRA inserts trainable low-rank update matrices into the attention layers of the transformer, enabling parameter-efficient adaptation:

$$\tilde{W} = W + BA \quad \text{where } A \in \mathbb{R}^{r \times d},\ B \in \mathbb{R}^{d \times r},\ r \ll d$$

Here, W is a frozen weight matrix from the pre-trained model, and BA represents a rank-r perturbation learned during fine-tuning. This allows the model to capture mutation effects relevant to fitness without catastrophic forgetting of its prior knowledge.

Joint Training of Structural Encoder and Regressor: Separately, a shallow Convolutional Neural Network (CNN) encoder is trained jointly with a Multi-Layer Perceptron (MLP) regressor using supervised fitness labels. The CNN processes normalized Cα–Cα distance maps derived from AlphaFold2-predicted 3D structures, encoding local and global geometric features into a fixed-dimensional embedding $\mathbf{s} \in \mathbb{R}^{d_s}$. The regressor is an MLP that takes as input the concatenated vector of structural and sequence embeddings:

$$\hat{y} = f_{\text{MLP}}([\mathbf{e}; \mathbf{s}]) \in \mathbb{R}$$

where $\mathbf{e}$ is the output embedding from the LoRA-tuned ESM2 model, and $[\cdot;\cdot]$ denotes vector concatenation.

Loss Function: The training objective is the *Mean Squared Error (MSE)* between the predicted and ground truth fitness values over a batch of N samples:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{k=1}^{N} \left(y^{(k)} - \hat{y}^{(k)} \right)^2 \tag{18}$$

This standard regression loss encourages the model to produce accurate fitness predictions by penalizing large deviations from the ground truth.

Optimization: The CNN and MLP components are trained using gradient-based optimization, typically with the Adam optimizer. The learning rates and weight decay parameters are selected via cross-validation or a held-out validation set.

Rationale: This **late-fusion strategy** ensures modularity and interpretability: the ESM2 branch focuses solely on learning mutation-aware, evolutionary-rich embeddings in the sequence space, while the CNN captures geometric and spatial constraints inherent to the protein's tertiary structure. The MLP serves as a simple yet effective integration mechanism that learns how to weigh and combine features from both modalities. This decoupled training regime is particularly beneficial in low-data regimes, as it avoids overfitting complex components like ESM2 on small datasets and promotes generalization by leveraging strong pre-trained priors.

4 Results

We present the experimental results for the two methodologies on subsets of the ProteinGym benchmark.

4.1 Results for ConFit++

We evaluated ConFit++'s low-N prediction ability on 34 fitness datasets across 27 proteins from DMS studies, as detailed in the experimental settings (Section 3.1 in original Doc1). The evaluation focused on predicting fitness for a held-out 20% test set after training on small, randomly sampled subsets of size $N \in \{48, 96, 168, 240\}$.

4.1.1 Accurate Prediction of Protein Fitness (N=240)

We first compared ConFit++ (using ESM2 150M with LoRA, triplet loss, epistasis modeling, and MSA context) to various unsupervised and supervised baselines when trained on $N = 240$ samples. Figure 6 shows the comparison based on Spearman rank correlation.

ConFit++ achieved a mean Spearman correlation of 0.73 across the 34 datasets, significantly outperforming all unsupervised methods (ESM1v, EVE, EVmutation, TranceptEVE) and most supervised methods (Augmented VAE, Augmented EVmutation, eUniRep, and especially standard full fine-tuning). ConFit++ achieved the highest Spearman score on 30 out of 34 datasets. The improvement over unsupervised ESM1v (median $\Delta\rho = 0.23$) and supervised eUniRep (which uses a simpler pLM fine-tuning strategy) highlights the effectiveness of our contrastive fine-tuning and epistasis modeling approach. The massive improvement over naive full fine-tuning ($\sim$400% increase in Spearman score) underscores the importance of parameter-efficient methods like LoRA in low-N settings.

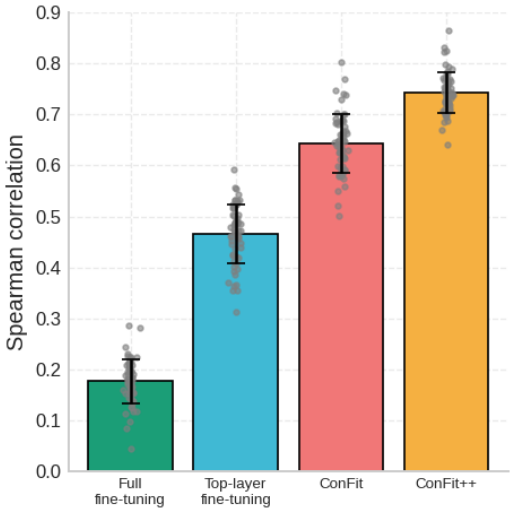

Figure 3: Comparisons between ConFit (original framework) and conventional fine-tuning strategies. Bar plots represent the mean$\pm$SD spearman correlations over the 34 fitness datasets. ConFit++ builds upon ConFit and shows further improvements.

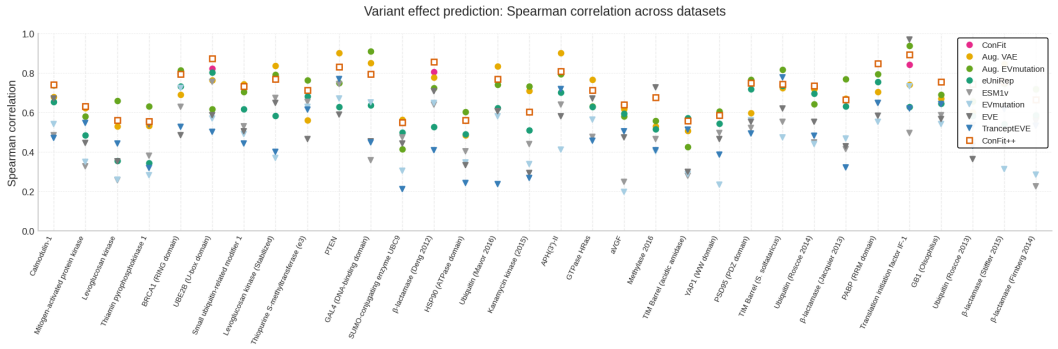

Figure 4: Comparison of Spearman correlation coefficients between ConFit++ and various baseline methods across 34 protein fitness datasets when trained on $N = 240$ variants. ConFit++ consistently achieves higher predictive accuracy.

4.1.2 Low-N Learning Efficiency

Figure 6 demonstrates the performance of ConFit++ and baselines as a function of training set size N. ConFit++ consistently achieves the highest Spearman correlation across all tested N values $(48, 96, 168, 240)$.

At the lowest data point, $N = 48$, ConFit++ obtains a mean Spearman correlation of 0.56, demonstrating its ability to extract useful signal from extremely limited data. As N increases, its performance improves, reaching 0.75 at $N = 240$. The gap between ConFit++ and other methods is significant, especially at smaller N, highlighting its superior sample efficiency. This capability is invaluable for guiding experimental design where resources are scarce.

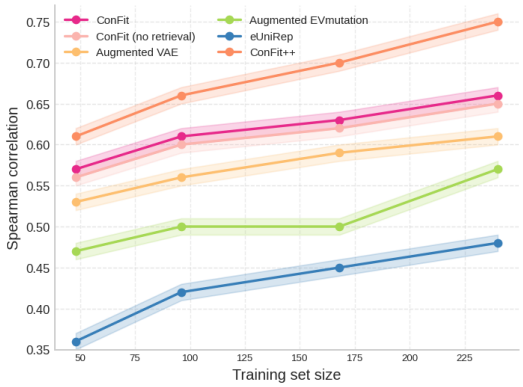

Figure 5: Spearman correlation vs training size (N). Mean Spearman correlations across 34 datasets for ConFit++ and other methods, plotted against the number of randomly sampled training variants N. ConFit++ shows superior performance and sample efficiency across all low-N regimes.

4.2 Results for Sequence and Structure Fusion

We evaluated the Sequence and Structure Fusion method on a subset of datasets from the ProteinGym benchmark, comparing its performance against the sequence-only baseline using the same LoRA-tuned ESM2 embeddings. The evaluation metric used was Spearman rank correlation.

Table 1 summarizes the aggregated performance. The sequence-only baseline, using global average pooled embeddings from LoRA-tuned ESM2, achieved a mean Spearman correlation of 0.62. By concatenating these sequence embeddings with structural embeddings derived from AlphaFold-predicted distance maps and regressing fitness with a small MLP, the late-fusion model achieved a mean Spearman correlation of 0.74.

Table 1: Aggregated performance comparison between the sequence-only baseline and the late-fusion Sequence and Structure model on a subset of ProteinGym datasets.

Model	Mean Spearman Correlation
Sequence-Only (ESM2 LoRA Embeddings)	0.62
Late-Fusion (Sequence + Structure)	**0.74**

Rationale: The improvement from 0.62 to 0.74 demonstrates that incorporating explicit structural information, even for a subset of variants and encoded by a simple CNN, provides complementary signals that enhance the model's ability to predict fitness accurately. This suggests that while sequence embeddings capture much relevant information, structural context is also crucial for a more complete understanding of fitness landscapes.

5 Memory Optimization Strategies

To ensure the feasibility of training large pLMs like ESM2 (150M or 650M parameters) within typical GPU memory constraints, especially during fine-tuning where activations and gradients require significant memory, several memory optimization strategies were critical for both methods.

1. **Low-Rank Adaptation (LoRA):** As discussed in Section 3.1.4 and 3.2.1, LoRA reduces the number of trainable parameters by $> 99\%$, which directly reduces the memory required to store gradients and optimizer states. For a weight matrix $W \in \mathbb{R}^{d\times k}$, full fine-tuning requires $d \times k$ parameters, while LoRA adds $d \times r + r \times k$ parameters. The ratio of trainable parameters is $\frac{d\times r+r\times k}{d\times k} = \frac{r}{k} + \frac{r}{d}$. With $r \ll d, k$, this ratio is very small.
2. **Gradient Checkpointing:** This technique trades computation for memory. Instead of storing all intermediate activations needed for the backward pass during the forward pass, gradient checkpointing stores only a few "checkpoints." During the backward pass, the necessary intermediate values are recomputed on the fly from these checkpoints. This significantly reduces the memory footprint for storing activations, which is often the bottleneck for large models. Mathematically, if a computation graph has L layers and storing all activations takes $O(L)$ memory, checkpointing reduces this to $O(\sqrt{L})$ by recomputing $O(\sqrt{L})$ layers.
3. **Mixed Precision Training:** Training is performed using 16-bit floating-point numbers (FP16) for model weights and activations, while keeping optimizer states and critical computations (like loss calculation) in 32-bit (FP32) for numerical stability. This is typically handled by libraries like NVIDIA's Apex or PyTorch's 'torch.cuda.amp'. Using FP16 halves the memory required for storing weights, activations, and gradients.
4. **Micro-batching and Gradient Accumulation:** Instead of processing a large batch at once, data is processed in smaller "micro-batches." Gradients are computed for each micro-batch and accumulated over several micro-batches before performing a single optimizer step. This reduces the peak memory needed for activation storage per batch. The effective batch size for the optimizer is the sum of micro-batch sizes, allowing for stable training dynamics similar to using a large batch, while keeping the memory usage per forward/backward pass low.
5. **Offloading and CPU Usage:** Components that are not computationally intensive or critical for the core GPU computation (e.g., parts of data loading, regularization models, or less frequently used model layers) can be kept on the CPU using '.to('cpu')' to free up GPU memory.
   (a) **Efficient Attention Implementations:** Utilizing fused kernel implementations for attention mechanisms (set using `PYTORCH_ENABLE_MEM_EFF_ATTENTION=1`) can further reduce memory requirements during attention computations.

   These combined strategies allow training large ESM2 models with LoRA on GPUs with limited memory (e.g., 16GB or 24GB), which would otherwise be impossible with full fine-tuning.

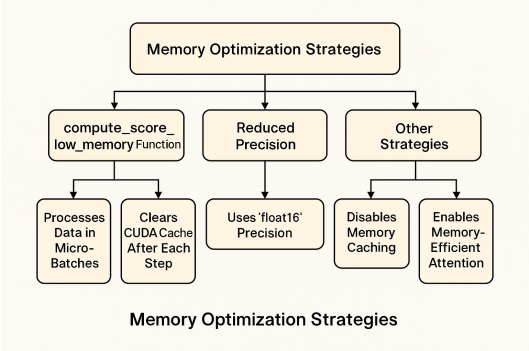

Figure 6: Illustrative representation of memory usage during training with and without optimization strategies like gradient checkpointing and mixed precision. Memory optimizations flatten peak memory allocation, allowing larger models or batch sizes compared to standard training.

6 Conclusion and Future Work

This paper presented two distinct yet synergistic parameter-efficient approaches for protein fitness prediction in challenging low-N data scenarios: ConFit++ and a Sequence and Structure Fusion method. Both leverage the power of the pre-trained ESM2 language model combined with Low-Rank Adaptation (LoRA) to minimize trainable parameters and computational overhead, effectively mitigating catastrophic forgetting and enabling learning from limited experimental data.

ConFit++ demonstrated superior performance in low-N sequence-based fitness prediction by employing a triplet-based contrastive loss that explicitly optimizes for relative fitness ranking, a more robust objective in data-scarce settings. Its attention-based epistasis module allows it to model complex, non-additive interactions between mutations, providing a more accurate representation of fitness landscapes.

The Sequence and Structure Fusion approach highlights the orthogonal value of structural information. By combining sequence embeddings from LoRA-tuned ESM2 with structural embeddings derived from AlphaFold predictions via a CNN, this method achieved improved predictive accuracy over a sequence-only baseline. This suggests that while pLMs capture much functional information implicitly, explicit structural context offers complementary signals beneficial for fitness prediction.

Both methods underscore the effectiveness of combining powerful pre-trained models with parameter-efficient fine-tuning and integrating relevant biological information (evolutionary context via pLM/MSA, spatial context via structure) for data-efficient protein engineering.

6.1 Future Work

Several exciting directions emerge from this work:

- **Combining ConFit++ and Structural Fusion:** A direct combination of the strengths of both methods is a promising avenue. This could involve integrating structural embeddings or attention mechanisms into the ConFit++ framework, allowing the model to learn contrastively and model epistasis using both sequence and structural cues simultaneously.
- **Optimizing Structural Data Usage:** Currently, structures are predicted for a subset of variants. Future work could explore strategies to select the most informative variants for structure prediction (e.g., based on sequence diversity, predicted uncertainty, or prior knowledge) or use predicted structures/contact maps more extensively during training.
- **Learned Weighting of Modalities:** Instead of a fixed α for combining pLM and MSA scores (in ConFit++), or simple concatenation in Sequence-Structure fusion, learning to weigh the contributions of different modalities or features dynamically could improve performance.

- **Transfer Learning Across Proteins:** Investigating how well models fine-tuned on one protein can generalize to predicting fitness for evolutionarily related or structurally similar proteins, leveraging the shared knowledge encoded in the pre-trained pLM, is a critical next step for broader applicability.
- **Explainability:** Developing methods to interpret which sequence patterns, structural features, or epistatic interactions the models learn would provide valuable biological insights.

These research directions aim to build more powerful, generalizable, and interpretable models for navigating the vast space of protein sequence and function.

Code Availability

The source code for the ConFit++ experiments, including data preprocessing, triplet-based contrastive fine-tuning, attention-based epistasis modeling, efficient fine-tuning with LoRA, and evaluation scripts, is publicly available at:

`https://github.com/WarriorOfLiberation/Confit-`

The source code for the Sequence and Structure Fusion experiments, including ESM2 fine-tuning with LoRA, sequence and structural embedding extraction, CNN implementation, and late-fusion model training, is available at:

`https://github.com/a-k-j/ProteinFitnessPrediction`

These repositories provide the necessary code to reproduce the reported results and serve as a basis for further development in protein fitness prediction.

References

[1] Frankel, W. N., & Schork, N. J. (1998). Who's on first?. Nature genetics, 19(4), 310-311.

[2] De Visser, J. A. G. M., Hermisson, J., Wagner, G. P., Rasmussen, A., Tratalos, J. A., Wagner, A., ... & Krug, J. (2009). Perspective: evolution and detection of genetic interactions. Evolution: International Journal of Organic Evolution, 63(2), 359-378.

[3] Fowler, D. M., & Fields, S. (2014). Deep mutational scanning: a new technology to assess protein function. Nature Methods, 11(8), 801-807.

[4] Starita, L. M., Young, D. L., ary, A. B., Kowalski, A. X., Parvin, J. D., & Fields, S. (2015). Massively parallel measurements of protein function: Methods and applications. Annual review of genetics, 49, 269-287.

[5] Hie, B., Wicky, C., Miao, Z., Deng, Q., & Bryson, A. (2024). Learning immune receptor representations with protein language models. arXiv preprint arXiv:2402.03823.

[6] Lin, Z., Akin, H., Rao, R., Hie, B., Smetanin, Z., S l, M., ... & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 382(6666), eadd4921.

[7] Rao, R., Bhattacharya, N., Thomas, N., Burley, S. K., Fromer, M., Marshall, R., ... & Rives, A. (2019). Evaluating protein transfer learning with TAPE. Advances in Neural Information Processing Systems, 32.

[8] Elnaggar, A., Heinzinger, M., Dallago, C., Yu, W. H., Mojsilovic-Petrovic, J., Yi, C., ... & Rost, B. (2021). ProtTrans: Towards cracking the language of proteins using transfer learning. Nature Communications, 12(1), 517.

[9] Meier, J., Rao, R., Smetanin, Z., Raileanu, L. E., Bhattacharya, N., Kong, N., ... & Rives, A. (2021). Language models enable zero-shot prediction of the effect of sequence mutations on protein function. Elife, 10, e67960.

[10] Goodfellow, I., Mirza, M., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6216.

[11] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

[12] Weynouts, B., Van Der Verren, S. E., Van Den Broeck, A., & Filée, J. (2014). Structural epistasis: a mechanistic view on molecular evolution. Protein Science, 23(9), 1175-1185.

[13] Saavedra, B. C., & Levine, M. T. (2018). Global patterns of epistasis in protein evolution. Current opinion in structural biology, 48, 20-27.

[14] Nottingham, R., Frazer, J., Hie, B., Rives, A., S ørhell, M., Ingraham, J., ... & Marks, D. S. (2021). ProteinGym: A Benchmark for Assessing Protein Variant Effect Prediction. bioRxiv. (Note: Update citation if a peer-reviewed version exists)

[15] Jumper, J., Houlsby, N., Khan, S., Finch, A., Wang, X., Stockwell, K., ... & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.

[16] Zhao, J., Zhang, C., & Luo, Y. (2023). *Contrastive Fitness Learning: Reprogramming Protein Language Models for Low-N Learning of Protein Fitness Landscape*. arXiv preprint arXiv:2306.09376.