Accelerating protein engineering with fitness landscape modeling and reinforcement learning

Haoran Sun[1], Liang He[1*], Pan Deng[1*], Guoqing Liu[1], Haiguang Liu[1], Chuan Cao[1], Fusong Ju[1], Lijun Wu[1], Tao Qin[1], Tie-Yan Liu[1]

[1]AI for Science, Microsoft Research.

*Corresponding author(s). E-mail(s): lihe@microsoft.com; paden@microsoft.com;
Contributing authors: 767780791shr@gmail.com; guoqingliu@microsoft.com; haiguangliu@microsoft.com; chuancao@microsoft.com; fusongju@microsoft.com; lijuwu@microsoft.com; taoqin@microsoft.com; tyliu@microsoft.com;

Abstract

Protein engineering is essential for a variety of applications, such as designing biologic drugs, optimizing enzymes, and developing novel functional molecules. Accurate protein fitness landscape modeling, such as predicting protein properties in sequence space, is critical for efficient protein engineering. Yet, due to the complexity of the landscape and high-dimensional sequence space, it remains as an unsolved problem. In this work, we present $\boldsymbol{\mu}$Former, a deep learning framework that combines a pre-trained protein language model with three scoring modules targeting protein features at multiple levels, to tackle this grand challenge. $\boldsymbol{\mu}$Former achieves state-of-the-art performance across diverse tasks, including predicting high-order mutants, modeling epistatic effects, handling insertion/deletion mutations, and generalizing to out-of-distribution scenarios. On the basis of prediction power, integrating $\boldsymbol{\mu}$Former with a reinforcement learning framework enables efficient exploration of the vast mutant space. We showcase that this integrated approach can design protein variants with up to 5-point mutations and potentially significant enhancement in activity for engineering tasks. The results highlight $\boldsymbol{\mu}$Former as a powerful and versatile tool for protein design, accelerating the development of innovative proteins tailored for specific applications.

Keywords: deep learning, reinforcement learning, protein mutational effect, protein fitness landscape, epistatic effects, protein language model, protein engineering, protein optimization

1 Introduction

Protein engineering aims to design proteins with desired functions and properties that are important for technology, agriculture, and medicine [1–5]. Efficient protein engineering, achieved through optimization of protein sequences, can greatly benefit the design of biologic drugs, enzymes, and beyond. A promising approach to achieve this goal is to accurate map protein sequences to their corresponding functionalities, ergo enabling efficient search of sequences with desired functions/properties.

However, the mapping between sequences and versatile functions in the space of protein sequences, known as fitness landscape, is complex and rugged. Advanced experimental techniques, such as Deep Mutational Scanning (DMS) [6] and Multiplex Assays of Variant Effects (MAVEs) [7], have been developed to couple functional assays with protein mutation detection. These assays can be measured in high throughput manners, enabling targeted and systematic probing of a wide range of quantification on fitness scores of protein variants around a starting sequence. Nonetheless, quantification from these experiments is still far from providing comprehensive landscapes, considering the vast protein sequence space and complex coupling between residues in proteins [4, 8, 9].

To accurately model a comprehensive landscape of fitness from limited experimental observations, many computational approaches have been proposed. Several methods rely on evolutionary information, such as Multiple Sequence Alignments (MSAs), to capture mutational effects of residues from natural homologous sequences of the proteins to be optimized. These approaches make predictions by utilizing site-wise information [10], pairwise features by a Potts model [1, 11], or hidden representations from a variational autoencoder [2, 12]. While these methods do not rely on experimental observations, they are limited to proteins with abundant homologs. They also struggle to extend to sequences with varying lengths, due to their dependence on fixed-length axial systems built on prepared MSAs. Recently, researchers start to leverage language models for mutational effect prediction and fitness landscape modeling. Language models are computational algorithms widely applied in natural language processing [13, 14]. Numerous efforts have applied language models to proteins by treating protein sequences as 'sentences' and residues as 'words' in natural language, demonstrating their capability in various fields [12, 15–17]. Pre-trained with large corpora of protein sequences collected from databases such as UniRef and Pfam, language models learn the probability of 20 amino acids for each position in given protein sequences, which can be connected to a proxy of mutational effect for each substitution [5, 18]. Although such approaches, known as a 'zero-shot' setting, alleviate reliance on homologs and MSAs, their phenotype-agnostic nature results in a discernible gap between predictions and ground truth, and prevents them from handling landscapes for different types of properties exhibited by the same protein.

Learning-based models that utilize DMS or MAVEs data of proteins for training [4, 9, 19–23] have also been applied to the fitness landscape prediction task, independently or combined with MSA information or language models. However, these models, mostly data-driven, often struggle to make reasonable predictions when experimental data is limited. To address this bottleneck issue, we propose a deep learning framework featuring three learning-based scoring modules that are designed to capture protein mutational effects at different levels of sequence features: single-residue validity, motif-level patterns, and sequence-level semantics. Combining with a pre-trained protein language model, our framework, named μFormer, is capable of modeling protein fitness landscape more accurately and comprehensively, with less dependency on observational data.

We demonstrate that μFormer can handle a variety of challenging scenarios, including limited number of measurements, orphan proteins with few homologs, complicated variants with multiple-point mutations, insertions and deletions, and out-of-distribution predictions. Extensive experiments indicate that μFormer outperform its counterparts on different tasks. More importantly, observing that μFormer excels at high-order mutational effect prediction when trained with single mutants, we deployed μFormer as a general tool navigating protein design in combination with a reinforcement learning protocol supporting efficient sequence search in the vast and rugged fitness landscape. Demonstrating an efficient and comprehensive exploration of mutant space consisting of an astronomical number of sequences with this pipeline, we engineered a beta-lactamase enzyme to hydrolyze a new substrate molecule and identified a number of variants with up to 5-point mutations that potentially exhibit significantly enhancement in activity against the new substrate.

2 Results

2.1 μFormer accurately predicts fitness scores of proteins

μFormer is composed of a self-supervised protein language model and a set of supervised scoring modules (Fig. 1). In the pre-training stage, the protein language model is trained with over 30 million protein sequences collected from UniRef50. By adopting a masked language modeling strategy, the protein language model learns to predict the most likely amino acids at targeted positions given the rest of residues in a protein sequence. Later, in the fine-tuning stage, the embeddings (representative vectors of numerical values) of the full-length query proteins and each residue, as well as the predicted probability for each amino acid in the proteins, are fed into 3 scorer modules respectively:

- The sequence-level scoring module learns to predict protein mutational effects from the embeddings of full-length protein sequences generated by the protein language model. This is motivated by the observation that embedding-derived semantics of protein sequences are relevant to their functions [15, 24, 25].
- The motif-level scoring module aims to extract consequential sequence patterns that are important for protein functions [26, 27].

- The residual-level scoring module calculates the fitness scores of given sequences by averaging the probability values of all amino acids in the sequence, while the probability values reflect the grammatical validity for each position conditioned on the entire sequence [16, 18, 28].

The outputs of the three scoring modules are then combined as the final prediction for the query sequence. We expect that the integration of these modules, each focuses on protein features at different levels and enhanced by the pre-trained protein language model, captures a more comprehensive fitness landscape of query proteins, with a boost in both robustness and accuracy.

To assess μFormer's ability in fitness landscape modeling and mutational effect prediction, we benchmarked it against nine other modeling approaches, including MSA-based methods [1, 10–12], language model-based zero-shot methods [5, 18] and learning-based approaches [21–23]. We first evaluated all models on the ProteinGym [5], a collection of DMS-derived protein mutational effect datasets spanning a diverse set of protein types, sequence lengths, biological functions, and fitness assays. Among all models, μFormer demonstrated the strongest ability in predicting mutational effects across different datasets (Fig. 2a-b). With a median Spearman's rank correlation coefficient (ρ) of 0.724, μFormer achieved the best results on 59 out of 78 test datasets and improved the performance by more than 10% on 24 datasets. Not surprisingly, supervised approaches outperformed both MSA-based and zero-shot models (Fig. 2a), as latter approaches are completely unaware of the biological function alterations associated with mutations. Further analysis indicates that μFormer is not sensitive to variations in training data size and homologous sequence numbers (Fig. 2b), suggesting that our method is a universal tool for protein mutational effect prediction.

Next, we evaluated μFormer's performance on insertion and deletion (indel) prediction. Indels are common mutation types that can lead to drastic changes in protein functions, and designing proteins of different lengths may be expected in specific scenarios. For example, short peptides with varying lengths are typical in therapeutic peptide design. However, indels further complicate the mutational space of proteins, rendering fitness prediction a more challenging task. While MSA-based methods are incapable of scoring indels, approaches of other types may not be able to extend to the scenario of indel prediction due to their designs. For example, the dependency of MSA-derived axial features restricted the application of ECNet on indel prediction. Here, we benchmarked μFormer on indel tasks against four alternative approaches. μFormer consistently outperforms other methods on two benchmark datasets with indel mutations (Fig. 2c): ProteinGym Indel, which collected indel-included DMS results from seven assays [5], and FLIP AAV, which split fitness assay data on capsid protein VP1 with varying strategies to probe model generalization [29, 30]. To be noted, the Mut-Des split assigns naturally occurring mutants (Mut) as training dataset and designed sequences (Des) as test dataset. Hence, performances on Mut-Des and the reverse split Des-Mut indicate models' out-of-distribution prediction ability. μFormer achieves a Spearman ρ above 0.8 for both settings, signifying its effectiveness as a surrogate model for guiding and optimizing protein sequence design.

2.2 μFormer captures epistatic mutational effect of high-order variants.

Due to time and cost constraints, most experimental assays for mutational effect quantification, including the DMS and MAVEs assays, focus on quantification of single-point mutations. Predicting the effect of combinations of mutations, i.e., high-order variants whose space grows exponentially with the increasing number of mutations involved, is therefore an ultimate goal of fitness landscape modeling. Towards this end, we observed that μFormer is capable of predicting the mutational effect of high-order variants from single mutants (single-to-multi).

We first collected nine sets of DMS assays that include both single-site mutants and high-order mutants and then trained prediction models with single-point data for each target protein. The nine proteins span a wide range of functions, including enzymes (beta-lactamase TEM-1 [31]) and reporter proteins (GFP [32]) of high interest in synthetic biology, disease-related proteins (Amyloid beta precursor protein APP [33]), signaling proteins (PSD-95 [34], GRB2 [34] and YAP1 [35]), RNA binding proteins (Pab1 [36]), and protein binding proteins (GB1 [37]). Also, we included the point mutation data from the aforementioned VP1 assay in our analysis. While the size of training data (single-site mutants) varies from 362 to 5468, the number of high-order mutants ($geq2$ mutations) can be much greater(for example, there 535,917 high-order mutants for GB1 protein, which has 1,045 single-site mutants used for training; Fig. 3a). Therefore, the analysis offers a comprehensive and rigorous assessment on high-order mutations prediction.

In comparison to other supervised approaches, representative zero-shot and alignment-based methods, μFormer shows the most robust performance across all tested proteins, with a Spearman ρ ranging from 0.653 to 0.946 (Fig. 3a). To be noted, the GFP dataset is composed of 12,777 double mutants and 37,852 variants with a mutation number spanning from 3 to 15. The accurate prediction of GFP high-order mutants (Spearman $\rho = 0.795$) demonstrates μFormer's effectiveness in generalizing to protein variants with multi-site mutations.

The interdependency between amino acids in protein gives rise to the phenomenon of epistasis, in which the combined mutational effects of the mutations at various sites are not simply the sum of the individual effects [8, 38, 39]. Epistasis poses a significant challenge for predicting high-order mutational effects. Therefore, we designed a series of tests specifically aimed at analyzing the epistasis modeling capability of μFormer. We calculated epistatic scores for high-order mutants as the discrepancy between observed mutational effect values and the sum of constituent single mutant mutational effect values, i.e. $S_{epi} = Effect_{pos_1,pos_2...pos_n} - \Sigma_{i=1}^{n} Effect_{pos_i}$. Compared to Ridge and ECNet, two approaches that demonstrated favorable performances in previous analysis, μFormer more accurately captures the direction of epistasis (Fig. 3b). In addition, the epistatic scores obtained from μFormer predictions exhibit a stronger correlation with observed S_{epi} in various proteins (Fig. 3b). Next, we designed a baseline method that assumes no epistasis in high-order mutants (additive model). In this model, the mutational effects of high-order mutants are estimated as the sum of the effects of single mutants. Using μFormer, we observed a significant improvement over the additive model (Fig. 3c). Moreover, for high-order mutants displaying the top 20%

absolute epistatic scores in each target protein, μFormer reduced predictive errors relative to observed values in 8 out of 9 proteins tested compared to the additive model. While the distribution of mutations and the strength of epistasis vary for each protein, our results collectively confirm that μFormer effectively captures epistatic effects, rather than trivially estimating the sum of single mutants present in the training data (Fig. 3d).

To gain insight into why and how μFormer achieves superior performance under the single-to-multi setting, we visualized the embeddings of both single mutants (training data) and high-order mutants (test data) extracted from μFormer using t-distributed Stochastic Neighbor Embedding (t-SNE) (Extended Data Fig. 1). Interestingly, we observed that the embeddings from both pre-trained and fine-tuned models aggregate according to fitness scores, rather than the number of mutations, in the corresponding variants. This observation provides an explanation for how μFormer can generalize to high-order mutants when trained exclusively with single mutants. The model's ability to capture the relationship between fitness scores and the corresponding variants enables it to effectively predict the outcomes for high-order mutants, despite not being explicitly trained on them. This highlights the robustness and adaptability of the μFormer model in handling complex mutation scenarios.

2.3 μFormer effectively identifies high-functioning variants with high-order mutations

The primary goal of protein engineering is to enhance the desired functionality of proteins of interest. While μFormer is able to distinguish gain-of-function mutations from loss-of-function mutations accurately (Extended Fig. 2), we further investigated if the model could predict top-performing mutants effectively. We calculated the Top-K recall scores of μFormer on the curated high-order mutant datasets with the single-to-multi setting. The assay aims to estimate the recovered percentage of the top K high-functioning high-order mutants from top K predictions (hits). In most cases, with K=100, μFormer provides more than 10 valid hits (Top-100 recall > 0.1), and the average Top-100 recall reaches 0.165 (Fig. 4a). When K scales up to 500, the recall score further improves and reaches an average value of 0.341 (Fig. 4a). This is a remarkable result, as 100 mutants only represent a small percentage (0.02% to 2%) of tested mutants for each protein. For YAP1, GB1, GRB2 and VP1, a subset of high-order mutants (ultra-high value mutants) exhibits fitness values higher than all single-site mutants of the corresponding protein used for training, which poses an additional challenge to modeling. Encouragingly, μFormer makes reasonable predictions for these out-of-distribution variants (Fig. 4a) and detects ultra-high valued mutants (Fig.4b). These results together show that μFormer can provide efficient guidance for protein optimization.

To demonstrate the performance of μFormer in practice, we focused on the acquired hydrolysis activity of TEM-1, a beta-lactamase that degrades beta-lactam antibiotics, such as ampicillin. Cefotaxime, discovered nearly 50 years ago, is a beta-lactam antibiotic resistant to TEM-1, making it effective against bacterial pathogens carrying TEM-1. However, TEM-1 variants active against cefotaxime rapidly evolved after the drug's introduction to the market. Among the 222 sequenced clinical variants of

TEM-1, 98 are marked as extended-spectrum beta-lactamases (ESBLs) [40], indicating their activity against extended-spectrum cephalosporins with an oxyimino side chain, including cefotaxime. Compared to non-ESBL clinical variant types that did not exhibit cefotaxime activities, most ESBLs (91 out of 98) are high-order mutants of TEM-1 (Extended Data Fig. 3). Therefore, we examined whether μFormer could identify high-order mutants with enhanced activities using ESBLs.

We fine-tuned μFormer with mutational effect quantifications of TEM-1 single mutants on cefotaxime collected by Stiffler et al. [41]. Next, we evaluated μFormer using a curated dataset comprising 105 ESBLs and 52 confirmed non-ESBLs [40, 42]. μFormer prioritizes ESBLs with high scores (Fig. 4c) and demonstrates a significant correlation (Spearman $\rho = 0.94$) with a quantification study on 16 TEM-1 mutants (including 11 high-order mutants) using MIC assay (Fig. 4d) [42].

To understand how μFormer achieves accurate predictions on hyperactive mutants, we extracted mutant embeddings of TEM-1 from μFormer and visualized the latent space with t-SNE. On the visualization, an increasing activity of TEM-1 against cefotaxime is clearly observed along the first dimension, with high-functioning variants aggregating on the right (Fig. 4e). Also, the width of bands stratified by mutational effect scores is consistent with the knowledge that the number of mutant sequences decreases with increasing level of functions [43, 44]. Moreover, when ranked by fitness values, the top 1% (50) TEM-1 mutants from a saturated single mutant DMS study [41] are enriched into 2 isolated clusters on the rightmost, along with 92 out of 105 ESBLs (Fig. 4e). Since the embeddings were extracted from the protein representation layer prior to scoring modules, these results indicate that μFormer learns the function of interest of target proteins from sequences, beyond the number of mutations in a mutant sequence.

2.4 Design high-functioning sequences with μFormer and reinforcement learning

Next, we employed the TEM-1-cefotaxime system to investigate if μFormer can effectively guide protein optimization. We designed a reinforcement learning (RL) method to search for TEM-1 variants with 1-5 point mutations that possess enhanced activity against cefotaxime, utilizing μFormer as the reward function to navigate the search (Extended Data Fig. 4a). The RL method enables efficient exploration of the vast mutant space comprising 6×10^{18} sequences, and incorporates Dirichlet noise into the PPO algorithm [45], which is recently used to align language models with human preferences [46, 47], to ensure the candidate diversity (Methods).

With the designed search framework, we recover 82,831 TEM-1 variants with enhanced activities from 1 million mutant candidates (Fig. 5a and Extended Data Fig. 4b-c). Compared to random mutagenesis, the mutation sites discovered by RL searching with μFormer prediction scores are located closer to cefotaxime binding site, and these sites are more closely packed in 3D structure of the beta lactamase (Fig. 5b). The sequences of RL discovered drug-resistant variants, along with experimentally observed TEM-1 variants, are mapped into a reduced 2D space (Fig. 5c). The highly dispersed clusters, with high-order mutants representing local optima or "peaks", are consistent with the rugged nature of fitness landscapes. While ESBLs

spread among diverse "peaks", implying diverse paths towards high-functioning variants in nature, confirmed non-ESBLs and the majority of single mutants aggregate in one cluster, indicating the presence of a local "valley". Based on the ESBL mutants, we characterized the dominant sites within each cluster and mapped these sites onto 3D structures (Fig. 5d). The dominant residues (in their wild type amino acids) are shown in ball-stick representations with sizes proportional to the observed frequencies, while the modelled cefotaxime (see Method) is shown in bond representations. We observed distinct features in the dominant mutation sites for five clusters shown in Figure 5d. Interestingly, although the mutation sites are spread over TEM-1 protein, we observed at least one mutation occurs in the close vicinity of the substrate binding site, implying that both local chemistry environments near the binding site and the global conformation changes are playing roles in cefotaxime binding and catalysis. We also observed that the most frequently observed mutation sites are either in loop region (such as E102, G236 and E237) or in the structured region but close to loops (such as M180 and T261). With an effective fitness prediction model and an efficient reinforcement learning framework, we successfully explored an enormous mutant space of TEM-1 and identified high-order variants with potentially significantly enhanced activity against cefotaxime spread across a rugged fitness landscape, for the first time. Through this example, we demonstrate that μFormer can serve as a powerful tool for predicting drug-resistant variants for given drug targets.

2.5 μFormer enables generalization to unseen residues in target proteins

Considering that saturated single-point mutagenesis assays can be impractical to probe protein fitness landscapes at large scales, we next investigated whether μFormer could generalize to residues unseen by the model. We used a saturated mutational effect assay of GB1 single and double mutants [37] and followed the single-to-multi setting aforementioned. We randomly selected 20%, 40%, 60%, and 80% of GB1 protein residues and trained the models with single mutant data exclusively covering these residues. For μFormer, we decreased the size of the supervised scoring modules to avoid overfitting (denoted as μFormer-SS). Subsequently, all prediction models were evaluated with saturated double mutants. We compared μFormer to Ridge, the model demonstrating favorable performance in general evaluation. With five random repeats for each selection ratio, we found that μFormer exhibited higher data efficiency and less sensitivity to the size of training data, achieving an average Spearman ρ of 0.46 when only 20% of residues were used for training (Fig. 6a). Furthermore, for each ratio and each repeat, we stratified the test dataset based on whether the two mutation sites were present in the training data (Fig. 6b). Under the 20% setting, μFormer reached an average Spearman ρ of 0.36 when neither of the residues were seen (2/2 unseen) and 0.64 when one of the residues was seen (1/2 unseen) (Fig. 6c). These results collectively indicate the data efficiency and generalization ability of μFormer, leading to the conclusion that μFormer can predict protein fitness and guide protein sequence design even with very limited data on the target protein.

2.6 Ablation studies reveal the contribution of each component in μFormer

Lastly, to assess the contribution of each component in μFormer, we conducted a series of ablation studies following the single-to-multi setting. As shown in Fig. 6d, omitting the language model results in the most significant drop in the performance of high-order mutational effect prediction, indicating the crucial role of pre-training in supervised tasks with limited labeled data. Meanwhile, scoring modules at different levels exhibit diverse roles for different target proteins, and combing the three modules results in the most robust performance (Fig. 6d). Moreover, to demonstrate that the multi-level scoring module design, rather than the model size or other factors, determines μFormer's superior performance, we benchmarked μFormer against three baselines on an additional dataset, FLIP GB1 [30], a landscape focusing on four epistatic sites in GB1 protein with high mutational space coverage. μFormer greatly outperforms ECNet, the learning-based method that also utilizes a language model, in all three settings provided by the benchmark dataset (Fig. 6e). Additionally, replacing the language model in ECNet with μFormer's language model (ECNet w/ μFormer encoder) does not substantially improve performance, and a μFormer variation with a model size similar to ECNet (μFormer-S) exhibits a more similar performance to μFormer (Fig. 6e). Thus, we conclude that the design of μFormer is essential for the state-of-the-art performance in accurate fitness modeling.

3 Discussion

Previous studies have demonstrated that sequence-based protein language models can lead to promising results in protein research, such as enzyme function prediction [48], antibody design [49], and enzyme optimization [50]. In this study, we developed a sequence-based protein language model framework, μFormer, which can generalize to the property predictions for diverse types of proteins. We further demonstrated its performance in enzyme optimizations through efficient fitness prediction. Our model, with the protein language model and the three scoring modules targeting protein features at residue/motif/sequence levels, is adaptable to various fitness prediction scenarios. It achieves state-of-the-art performance on nearly all tasks, including high-order mutants, epistatic effects, high-functioning sequences, indel mutations, and more. We also highlight the data efficiency and generalization ability of μFormer, as well as the significance of μFormer's model design in achieving its performance.

We propose a prediction task to assess μFormer's performance in practice. While the mutational effect of TEM-1 against ampicillin has been widely studied, no approaches have focused on TEM-1's gain-of-activity against cefotaxime. A major challenge for the latter task arises from the fact that high-order TEM-1 mutants can lead to a drastic increase in cefotaxime hydrolysis activity, while the experimental data (such as DMS) covers only single mutants. This necessitates a strong generalization ability of the prediction method for both high-order mutants and out-of-distribution prediction. We show that μFormer effectively recover high-order, high-functional mutants against cefotaxime in the clinical TEM-1 isolates dataset we curated. Additionally, we have identified variants with an activity up to 10,000 times higher than

that of the wildtype TEM-1. Both findings serve as robust evidence of μFormer's capabilities. Furthermore, we reveal how μFormer can guide efficient mutant search. By coupling μFormer with a reinforcement learning framework, we have explored a protein mutant space consisting of 6×10^{18} sequences and identified rich TEM-1 variants with up to 5-point mutations that are expected to possess enhanced activity against cefotaxime. The results highlight the effectiveness of our pipeline in discovering promising protein variants that can be further investigated and validated experimentally, thus paving the way for designing new proteins tailored for specific applications.

In μFormer, supervised scorer modules demand experimental data for the protein of interest to make predictions, which may limit its application due to the scarcity or inaccessibility of experimental data in practice. While μFormer can be easily applied in its zero-shot mode, we believe that the "fine-tune" mode offers improved accuracy in fitness landscape prediction. As Biswas et al. [4] mentioned, zero-shot modeling based purely on protein language models can guide us away from unfavored sequences but cannot help us identify high-functioning sequences efficiently. Meanwhile, Stiffler et al. [41] showed that TEM-1's mutational effects on ampicillin and cefotaxime, both of which are beta-lactams, are poorly correlated (Spearman $\rho = 0.03$, Extended Fig. 4a). Notably, residues crucial for hydrolysis of ampicillin can be detrimental for cefotaxime degradation (Extended Fig. 4b). These evidences jointly support the necessity of phenotype-aware fine-tuning for target proteins.

Both structure and sequence-based methods have been proposed for protein fitness landscape modeling. We focus on sequence features and protein language models, aside from their success in other studies, for the following reasons: 1) Accessible sequence data is much larger than high quality structural data, which can greatly benefit model training and, therefore, its performance. 2) Most 3D structures, derived from either experimental or computational approaches (e.g., Alphafold2 [51]), are not sensitive to point mutations [52], making it challenging to link mutations to functional alterations. 3) Many protein design tasks, such as antibody designs, flexible regions of proteins that may not have a stable 3D structure. However, it is well noted that structural information may better inform the epistatic effects between residues, especially for those within spatial neighborhood but sequentially apart. As such, integrating structural features into the pipeline is expected in the future.

Despite the outstanding performance of μFormer, we anticipate improvements in accuracy and generalizability of our method through several aspects. First, although our method has demonstrated higher accuracy in capturing high-order mutational epistatic effects compared to other approaches, there remains a gap between predictions and observations. In addition to the 3D structural information mentioned above, incorporating a module explicitly predicting the strength of epistatic effects will enhance performance. Second, to further reduce the need for training data, on way is to develop a unified model for different kinds of phenotypes across multiple proteins by encoding phenotypes and incorporating phenotype representations into the pipeline. For instance, a single model may be capable of predicting TEM-1's fitness on various substrates, provided that ampicillin, cefotaxime, and other molecules are encoded and exposed to the machine learning model. Lastly, the reliance on pre-trained language models with a context constraint on sequence length may limit presented model's

performance for longer sequences (>1024 aa). This makes it necessary to develop a pre-trained model capable of handling longer sequence contexts.

Protein fitness landscape modeling is becoming more and more important in recent years, thanks to the advancement in both experimental measurements and computational quantification. Deep learning models start to demonstrate their potentials in accurately predicting the fitness of proteins in the vast sequence space. Many models depend on large training data, μFormer utilizes a pre-trained large model on protein sequences, and achieves outstanding performances in multiple diverse prediction tasks. It is foreseeable that deep learning models will further accelerate the research on protein fitness, and contribute to various fields including biologic drug design, protein vaccine optimization, and protein engineering.

4 Methods

4.1 Fitness Landscape Model

Our proposed μFormer is a deep learning solution for mutation effect prediction, i.e., predicting the fitness score of a mutated protein sequence. Accurate predictions are achieved in two steps: first, we pre-train a masked protein language model (LM) using a large database of unlabeled protein sequences; second, we introduce three scoring modules (each with a small set of new parameters) into the pre-trained protein LM for the final fitness score prediction and train all parameters using a set of mutant protein sequences with measured fitness scores. Figure 1 provides an overview of the prediction model.

In this section, we briefly introduce the masked protein LM and the three scoring modules.

4.1.1 Protein Language Model

The masked language model (MLM) [13, 26] is a self-supervised learning technique that utilizes the Transformer encoder [53] to learn representations for unlabeled sequences. During training, a random subset of tokens (typically 15%) in the input sequence are masked, and the model is trained to recover the masked tokens, i.e., to minimize the negative log likelihood of masked tokens:

$$\mathcal{L}_{mlm} = \mathbb{E}_{x \in \mathcal{D}} \left(\mathbb{E}_M \sum_{i \in M} \left(-\log P_\theta(x_i \mid x_{/M}) \right) \right), \tag{1}$$

However, protein sequences differ significantly from natural language sentences. Instead of conveying meaning through syntactic and semantic relationships between words, a protein sequence consists of a linear arrangement of amino acids, each with unique physicochemical properties. Together, the amino acids linked sequentially determine a protein's three-dimensional structure and function. The collective effect of these residues as a whole reflects the sequence's function, making it essential to learn protein sequence representations, in order to capture the inter-residue co-variation within the sequences. Conventional masked language models for proteins

model masked tokens (i.e., amino acid residues) by conditioning on unmasked tokens only, while processing each token independently. In contrast, the pairwise masked language model (PMLM) considers the dependency among masked tokens, taking into account the joint probability of a token pair [54]. One crucial distinction between natural language sentences and protein sequences is that the joint probability cannot be determined by the independent probability of each token. In other words,

$$P(x_i, x_j \mid x_{/M}) \neq P(x_i \mid x_{/M}) \times P(x_j \mid x_{/M})$$

for the i-th and j-th positions, which represent masked tokens or residues in M. This aspect is essential for capturing the co-evolutionary information between elements within a protein sequence. This model is applied to generate a more accurate pre-trained encoder for protein sequence representation.

The loss functions we adopted for protein language model pre-training include the following:

$$\mathcal{L}_{pmlm} = \mathbb{E}_{x \in \mathcal{D}} \left(\mathbb{E}_M \sum_{i,\, j \in M} \left(-\log P_\theta(x_i, x_j \,|\, x_{/M}) \right) \right), \tag{2}$$

where $\mathcal{D}$ represents the set of input sequences, X is a sequence in $\mathcal{D}$, $x_{/M}$ represents the masked version of x where the indices of the masked tokens are M, x_i is the i-th token in the sequence x, and θ denotes the parameters to be learnt.

Each Transformer encoder layer consists of two modules: multi-head self-attention (MHA) and position-wise feed-forward (FFN), which are connected through residual connections and layer normalization, as shown below:

$$x = \textsc{LayerNorm}(x + \mathrm{MHA}(x)), \tag{3}$$

$$x = \textsc{LayerNorm}(x + \mathrm{FFN}(x)). \tag{4}$$

The MLM head and the pairwise MLM head, which predict the masked tokens (i.e., residues in our problem) and the masked token pairs, are both two-layer MLPs that maps the hidden vector at each position to a probability distribution over all possible residues and residue pairs respectively.

In μFormer, we pre-train the protein LM with the UR50 dataset with about 30 million proteins.

4.1.2 Scoring Modules

Motivated by biological insights, we introduce three scoring modules on top of the pre-trained protein LM to predict the fitness score for a protein sequence, which focus on different levels of granularity of the protein sequence, as illustrated in Figure 1:

- The residual-level score $S_{\textsc{Resi}}(x)$, which characterizes the grammatical validity of residues at each position conditioned on the entire sequence;
- The motif-level score $S_{\textsc{Motif}}(x)$, which aims to capture the local sequence information around a residue beyond the residue granularity, considering that motif is widely used in biological sequence modeling for different tasks [9];

- The sequence-level score $S_{\text{SEQ}}(x)$, is motivated by the observation that the semantics of protein sequences are relevant to their properties/functions on the whole sequence granularity [16, 28].

The final fitness score $S(x)$ is calculated by

$$S(x) = S_{\text{RESI}}(x) + S_{\text{MOTIF}}(x) + S_{\text{SEQ}}(x). \tag{5}$$

Residual-level Scoring Module

This module calculates the fitness score for a mutant-type sequence by averaging the log-likelihood of the residual at each position, as follows:

$$S_{\text{RESI}}(x;\theta) = \frac{1}{L}\sum_{i=1}^{L} \log p_\theta(x_i|x), \tag{6}$$

where L is the length of the sequence x and $p_\theta(x_i|x)$ is the estimated likelihood/probability of the i-th residue in the mutant-type sequence x by the protein LM p_θ.

As can be seen, this score only depends on the mutant-type sequence and does not explicitly rely on its wild-type sequence. Notably, previous models [15, 18] often use the term $\sum_{i\in M} \log \frac{p(x^i_{mt})}{p(x^i_{wt})}$ for mutational effect prediction, where M is the set of mutated positions, and $p(x^i_{mt})$ and $p(x^i_{wt})$ are the estimated probabilities for the i-th residues of the mutant-type and wild-type sequences, respectively. An obvious limitation of this kind of models is that they assume the same length of the mutant-type and wild-type sequences and thus cannot handle mutants resulted from indel (insertion/deletion) operations. In contrast, our formulation does make the same-length assumption, making it suitable for mutations with insertions or deletions.

Motif-level Scoring Module

Motifs, which are consequential sequence patterns, play a crucial role in protein sequence modeling, widely used in bioinformatics research [26, 27]. In this work, we leverage convolutional neural networks (CNNs) to capture the local features of protein sequences. Specifically, we employ a convolutional module with max pooling and skipped connection to quantify the fitness from the local feature perspective. The score is computed as

$$S_{\text{MOTIF}}(x;\alpha) = \textsc{Linear}(\textsc{MaxPool}(\textsc{Conv}(x) + x)), \tag{7}$$

where x denotes the input sequence (of vectors) to the convolutional module, Conv and MaxPool denote the convolution operation and max pooling operation, respectively, and α is the parameters to be learnt in this module. The skipped connection is used to combine the output of the convolutional module with the input sequence, which helps to preserve the information from the original sequence. The score, S_{MOTIF}, is calculated by applying a linear transformation to the output of the convolutional module.

Sequence-level Scoring Module

It has been verified [15, 24] that the embedding of the [CLS] token in a natural language sentence is a good representation of the whole sentence. Therefore, in this work, the sequence-level scoring module in μFormer takes the embedding of the [CLS] token of a protein sequence as its representation. A Multi-Layer Perceptron (MLP) is employed to map the sequence representation to a fitness score, which is defined as

$$S_{\text{SEQ}}(x;\beta) = \text{MLP}(x_{cls}), \tag{8}$$

where x_{cls} denotes the embedding of the [CLS] token generated by the pre-trained protein LM and β is the parameters to be learnt for this scoring module.

4.1.3 Training

While the parameters θ of the protein LM is pre-trained using proteins without annotations, we need a set of proteins with measured fitness scores to train the newly introduced parameters (i.e., α and β) and to refine θ.

Let $\mathcal{D}$ denote a dataset consisting of pairs (x, y), where x represents a sequence of the wild type or mutant protein, and y is the fitness score of the sequence. All the parameters are trained to minimize Mean Absolute Error (MAE):

$$\mathcal{L} = \sum_{(x,\,y)\in\mathcal{D}} |S(x;\theta,\alpha,\beta) - y|, \tag{9}$$

where $S(x;\theta,\alpha,\beta)$ is the final fitness score obtained by summing three scoring modules.

To prevent overfitting the training data, which is likely derived from limited observations in biological experiments, the scoring modules are designed with significantly fewer parameters than the large, pre-trained encoder. Meanwhile, protein sequences typically consist of hundreds of amino acids, each with 20 possible alternatives. This necessitates a comparable number of parameters to effectively capture the information they contain.

4.1.4 Implementation

The protein language models used in this study were pre-trained on the UR50 (release 2018_03) dataset. The base model was configured with a hidden size of 768, feed forward dimension of 3072, 12 encoder layers, and 12 attention heads. Additionally, a larger model was pre-trained with a hidden size of 1280 and 34 encoder layers. The base model was trained with a maximum length of 512, while the larger model was trained with a maximum length of 1024. Both models utilized non-learnable positional encoding. During pre-training, the base model used the Adam optimizer with parameters $(0.9, 0.98)$, a peak learning rate of 0.0003, and a clip norm of 1.0. The learning rate was scheduled with a polynomial decay function, gradually decreasing after a warm-up period of $20,000$ steps. The larger model followed a similar hyperparameter configuration, with the peak learning rate set to 0.0001. Two models, μFormer-S and μFormer-L, are built on these two pre-trained models to assess the performance, with parameter sizes of 89 million and 670 million, respectively. Since μFormer can

be utilized for zero-shot prediction without training, we denote the zero-shot version of the model as μFormer-Zero. μFormer-S and μFormer-L are trained using the same set of hyper-parameters across all datasets for a maximum of 300 epochs, with a hidden size of 256 and a motif scoring layer of 1. This indicates the potential for further performance improvement by tuning hyper-parameters for specific datasets.

4.1.5 Datasets

FLIP [30]. The data is downloaded from Github: https://github.com/J-SNACKKB/FLIP/tree/main/splits/. We apply the original train/valid/test splits for evaluation, including the zero-shot models.

ProteinGym [5]. The data is downloaded from https://github.com/OATML-Markslab/ProteinGym. The substitution dataset is composed of approximately 1.5 million variants from 87 assays. Considering modeling efficiency, assays with target proteins longer than 1024 amino acids are removed, leaving 78 assays in the final analysis. The indel dataset is composed of approximately 300,000 mutants from 7 assays. For evaluation, we randomly select 10% of records from each assay as a test set, for all trained models including zero-shot models. The performances are reported on the fixed hold-out test set for a fair comparison.

4.1.6 Evaluation Protocol

For the supervised learning setting, we apply the original data split scheme for the FLIP benchmark, and evaluate the metrics fairly on the same hold-out test set. For the datasets from ProteinGym, we randomly select 10% of records from each dataset as a hold-out test set to evaluate all models rigorously, including the zero-shot models. The performance numbers are reported on the fixed hold-out test set for a fair comparison. Spearman's rank correlation scores are recorded for each dataset.

4.1.7 Baselines

The multiple sequence alignments (MSAs) of the target wild-type sequences are prepared in advance using HHblits, by searching against the Uniclust30 database. For the ProteinGym datasets, the author provided MSAs are utilized. During our experiments, all models requiring evolutionary information features, including site-independent, EVE, EVmutation, DeepSequence, and ECNet, are trained on the same MSAs.

For the Tranception [5] model, we downloaded weights of the largest model (denoted as Tranception-L) from official website for zero-shot prediction. To generate zero-shot predictions using ESM-1v [18], we evaluated the ensemble results of 5 checkpoints downloaded from the official website. For the supervised setting of ESM-1v, we compared the results for the mean over subset, which was referred to as 'mut mean' in the FLIP benchmark [30]. This setting resulted in better overall performance than other settings evaluated on the benchmark. In this setting, fitness scores were calculated by regressions on sequence representations that were mean-pooled over the residues

in the mutated region.

For the supervised models, we adopt the public implementations and retrained the models on the same data splits for each dataset. Especially, for the Ridge [23] and the CNN [30] models, we used the model architecture and hyperparameters implemented by the FLIP codebase [30]. While for ECNet [22], we reproduced the ECNet model using the open-sourced code provided by Luo et al [22]. Since ECNet both leverages MSAs and labeled data, we retrained the models under the same settings while with the prepared MSAs.

4.2 Reinforcement Learning

To explore the vast space of mutations and identify high-functioning mutants in the protein landscape, we developed a reinforcement learning (RL) method that utilizes the μFormer model as a reward function. In each episode, the RL agent sequentially mutates a single-site residue until reaching a fixed horizon, such as 5. The goal of the agent is to learn how to select mutation sites and types to optimize protein sequences. The algorithm alternates between two phases: exploration and learning. During the exploration phase, we employ a mutation site policy network and a mutation type policy network to generate potentially high-functioning mutants, aided by Dirichlet noise. During the learning phase, we use the μFormer model to label the generated mutants, and update the policy networks to provide mutants with higher fitness scores.

4.2.1 Policy Networks for Mutation Site and Mutation Type Selection

The RL agent mutates a single-site residue at each step of an episode. This involves selecting a mutation site in the sequence and choosing a mutation type to apply. To facilitate this, we develop two policy networks. The first one, called the mutation site policy network, takes in both the wild-type sequence and the current mutant sequence, which is represented by a binary matrix of size $20 \times 2L$[1]. The network determines where changes should be made in the current mutant sequence. The second one, called the mutation type policy network, takes in the wild-type sequence, current mutant sequence, and the selected mutation site, which is represented by a binary matrix of size $20 \times 2L + L$. The network determines the specific changes to be made at the selected mutation site. To optimize both policy networks, we used the Proximal Policy Optimization (PPO) algorithm [45], which is recently used to align language models with human preference [46]. To stabilize the training of the policy networks, we train two value networks that estimate the expected return of a given state. During training, the policy network is updated based on the difference between the observed μFormer reward and the predicted average value from the value networks, which is also referred to as the advantage function. This helps the policy network to learn which actions are more likely to lead to higher rewards and adjust its behavior accordingly. The loss

[1] 20 represents the number of types of amino acid and L is the length of the protein sequence.

functions for updating both the policy network and value network are as follows:

$$\begin{aligned} &L_{policy}(\theta) = \mathbb{E}_t[\min(r_t(\theta) \cdot A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t)], \\ &\text{where } A_t = \frac{\sum_{i=1}^{N} S_i(x)}{N} - v_\phi, \\ &L_{value}(\phi) = (\frac{\sum_{i=1}^{N} S_i(x)}{N} - v_\phi)^2. \end{aligned} \tag{10}$$

$r_t(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$ denotes the probability ratio of the current policy π_θ and the old policy $\pi_{\theta_{old}}$, where θ denotes the parameters of either the mutation site policy network or the mutation type policy network. To improve the robustness of the reward function, we utilize a collection of μFormer models and calculate their average value as the reward, denoted as $S_i(x)$ for the mutant sequence x from the i-th μFormer model. The advantage function A_t is obtained by subtracting the observed μFormer reward from the estimated value v_ϕ. The policy loss $L_{policy}(\theta)$ is designed to increase the probability of actions with positive advantage functions, while avoiding significant deviations from the old policy to prevent performance collapse. The hyperparameter ϵ controls the size of the trust region for each policy update. On the other hand, the value loss $L_{value}(\phi)$ is a mean-squared error loss that ensures the value network accurately predicts the expected final reward.

4.2.2 Adding Dirichlet Noise into Exploration

To explore the vast mutation space and discover diverse high-profile mutants, we incorporate a Dirichlet exploration noise into the generation of mutants during the exploration phase, which helps prevent getting trapped in local optima. Specifically, the Dirichlet noise is added to the probabilities from the policy networks, such that $P(s,a) = (1-\epsilon) \cdot \pi_\theta(a|s) + \epsilon \cdot \eta$, where $\eta \sim dirichlet(0.03)$ and $\epsilon = 0.25$. This noise ensures that all moves could be attempted, while still favoring mutations with higher fitness scores. During the learning phase, the Dirichlet noise introduced may result in an action with low probability from the policy network π_θ, which can cause the importance ratio $r_t(\theta)$ to explode. To address this issue, we define a new ratio $r_t(\theta) = \frac{P(s,a)}{P_{old}(s,a)}$ to smooth the training process.

4.3 Result analysis

The complex structure of TEM1 and cefotaxime was constructed based on the PDB structures TEM-1 (PDB ID: 1TEM) and cefotaxime in complex with CTX-M-9 S70G (PDB ID: 3HLW). The protein structures were aligned to obtained the translation-rotation matrix, which is applied to the cefotaxime coordinates to get a structural model of the TEM1-cefotaxime complex for downstream analysis. The distances between the center of cefotaxime and each residue were calculated using UCSF Chimera 1.16.

Data Availability.
To be updated soon.

Code Availability.
To be updated soon.

Acknowledgments.
We would like to extend our heartfelt thanks to Dr. Ostermeier for his invaluable support and generosity in providing information on plasmids pSkunk3-TEM-1 and pTS42, as well as for kindly sharing the pTS42 plasmid with us.
We thank Tianbo Peng for his support on building the demo webpage for this paper.

Authors' contributions.
Conceptualization, L.H., H.S., and P.D.; methodology and modeling, L.H., H.S., G.L., F.J., and L.W.; data curation, L.H., H.S., and P.D.; result interpretation, P.D., H.L., L.H., and C.C.; writing-original draft, L.H., P.D., H.S., and G.L.; writing-review, H.L., T.Q., C.C.; supervision, T.L.. All authors read and approved the final manuscript.

Ethics approval.
Not applicable.

Consent to participate.
Not applicable.

References

[1] Gray, V. E., Hause, R. J., Luebeck, J., Shendure, J. & Fowler, D. M. Quantitative missense variant effect prediction using large-scale mutagenesis data. *Cell systems* **6**, 116–124 (2018).

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

[3] Miton, C. M. & Tokuriki, N. Insertions and deletions (indels): a missing piece of the protein engineering jigsaw. *Biochemistry* (2022).

[4] Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M. & Church, G. M. Low-n protein engineering with data-efficient deep learning. *Nature methods* **18**, 389–396 (2021).

[5] Notin, P. *et al.* *Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval*, 16990–17017 (PMLR, 2022).

[6] Fowler, D. M. & Fields, S. Deep mutational scanning: a new style of protein science. *Nature methods* **11**, 801–807 (2014).

[7] Weile, J. & Roth, F. P. Multiplexed assays of variant effects contribute to a growing genotype–phenotype atlas. *Human genetics* **137**, 665–678 (2018).

[8] Wittmund, M., Cadet, F. & Davari, M. D. Learning epistasis and residue coevolution patterns: Current trends and future perspectives for advancing enzyme engineering. *ACS Catalysis* **12**, 14243–14263 (2022).

[9] Gelman, S., Fahlberg, S. A., Heinzelman, P., Romero, P. A. & Gitter, A. Neural networks to learn protein sequence–function relationships from deep mutational scanning data. *Proceedings of the National Academy of Sciences* **118**, e2104878118 (2021).

[10] Sim, N.-L. *et al.* Sift web server: predicting effects of amino acid substitutions on proteins. *Nucleic acids research* **40**, W452–W457 (2012).

[11] Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nature biotechnology* **35**, 128–135 (2017).

[12] Frazer, J. *et al.* Disease variant prediction with deep generative models of evolutionary data. *Nature* **599**, 91–95 (2021).

[13] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Floridi, L. & Chiriatti, M. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines* **30**, 681–694 (2020).

[15] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021).

[16] Hie, B., Zhong, E. D., Berger, B. & Bryson, B. Learning the language of viral evolution and escape. *Science* **371**, 284–288 (2021).

[17] Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).

[18] Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems* **34**, 29287–29303 (2021).

[19] Kim, H. Y. & Kim, D. Prediction of mutation effects using a deep temporal convolutional network. *Bioinformatics* **36**, 2047–2052 (2020).

[20] Shanehsazzadeh, A., Belanger, D. & Dohan, D. Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443* (2020).

[21] Yang, K. K., Lu, A. X. & Fusi, N. Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv* 2022–05 (2022).

[22] Luo, Y. *et al.* Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications* **12**, 1–14 (2021).

[23] Hsu, C., Nisonoff, H., Fannjiang, C. & Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology* 1–9 (2022).

[24] Kulmanov, M. & Hoehndorf, R. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics* **36**, 422–429 (2020).

[25] Elnaggar, A. *et al.* Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225* (2020).

[26] Wu, L. *et al.* Sproberta: protein embedding learning with local fragment modeling. *Briefings in Bioinformatics* **23** (2022).

[27] Bileschi, M. L. *et al.* Using deep learning to annotate the protein universe. *Nature Biotechnology* 1–6 (2022).

[28] Hie, B., Zhong, E., Bryson, B. & Berger, B. Learning mutational semantics. *Advances in Neural Information Processing Systems* **33**, 9109–9121 (2020).

[29] Bryant, D. H. *et al.* Deep diversification of an aav capsid protein by machine learning. *Nature Biotechnology* **39**, 691–696 (2021).

[30] Dallago, C. *et al. Flip: Benchmark tasks in fitness landscape inference for proteins.*

[31] Jacquier, H. *et al.* Capturing the mutational landscape of the beta-lactamase tem-1. *Proceedings of the National Academy of Sciences* **110**, 13067–13072 (2013).

[32] Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).

[33] Seuma, M., Faure, A. J., Badia, M., Lehner, B. & Bolognesi, B. The genetic landscape for amyloid beta fibril nucleation accurately discriminates familial alzheimer's disease mutations. *elife* **10**, e63364 (2021).

[34] Faure, A. J. *et al.* Mapping the energetic and allosteric landscapes of protein binding domains. *Nature* **604**, 175–183 (2022).

[35] Araya, C. L. *et al.* A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. *Proceedings of the National Academy of Sciences* **109**, 16858–16863 (2012).

[36] Melamed, D., Young, D. L., Gamble, C. E., Miller, C. R. & Fields, S. Deep mutational scanning of an rrm domain of the saccharomyces cerevisiae poly (a)-binding protein. *Rna* **19**, 1537–1551 (2013).

[37] Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current biology* **24**, 2643–2651 (2014).

[38] Starr, T. N. & Thornton, J. W. Epistasis in protein evolution. *Protein science* **25**, 1204–1218 (2016).

[39] Sailer, Z. R. & Harms, M. J. High-order epistasis shapes evolutionary trajectories. *PLoS computational biology* **13**, e1005541 (2017).

[40] Naas, T. *et al.* Beta-lactamase database (bldb)–structure and function. *Journal of enzyme inhibition and medicinal chemistry* **32**, 917–919 (2017).

[41] Stiffler, M. A., Hekstra, D. R. & Ranganathan, R. Evolvability as a function of purifying selection in tem-1 β-lactamase. *Cell* **160**, 882–892 (2015).

[42] Weinreich, D. M., Delaney, N. F., DePristo, M. A. & Hartl, D. L. Darwinian evolution can follow only very few mutational paths to fitter proteins. *science* **312**, 111–114 (2006).

[43] Maynard Smith, J. Natural selection and the concept of a protein space. *Nature* **225**, 563–564 (1970).

[44] Orr, H. A. The distribution of fitness effects among beneficial mutations. *Genetics* **163**, 1519–1526 (2003).

[45] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[46] Introducing ChatGPT. https://openai.com/blog/chatgpt [Accessed: Aug. 2023].

[47] Ouyang, L. *et al.* Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **35**, 27730–27744 (2022).

[48] Yu, T. *et al.* Enzyme function prediction using contrastive learning. *Science* **379**, 1358–1363 (2023).

[49] Hie, B. L. *et al.* Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology* (2023).

[50] Lu, H. *et al.* Machine learning-aided engineering of hydrolases for pet depolymerization. *Nature* **604**, 662–667 (2022).

[51] Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).

[52] Buel, G. R. & Walters, K. J. Can AlphaFold2 predict the impact of missense mutations on structure? *Nature Structural & Molecular Biology* **29**, 1–2 (2022).

[53] Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).

[54] He, L. *et al.* Pre-training co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527* (2021).

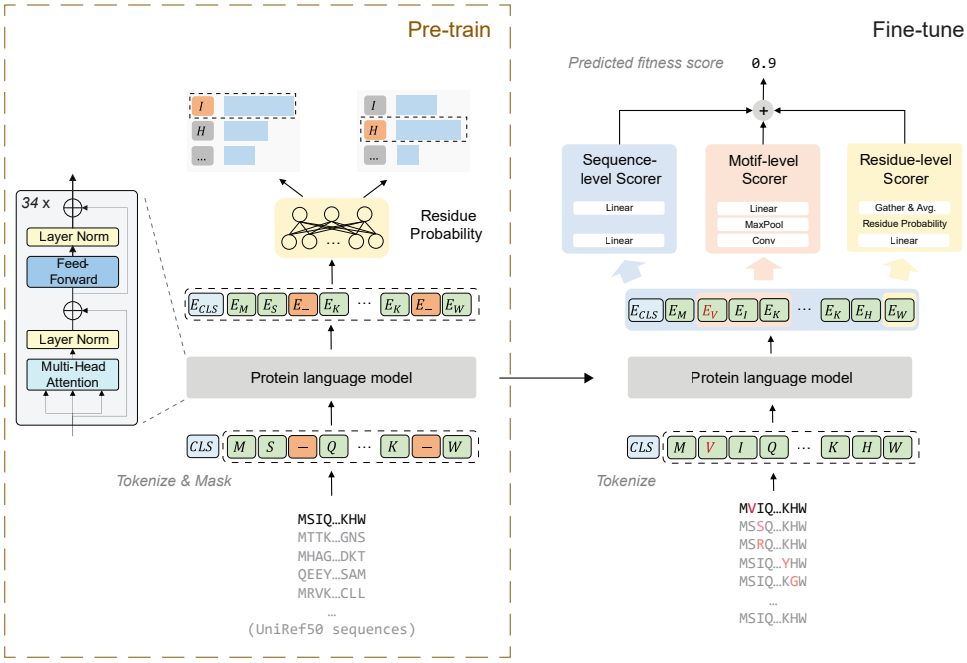


Fig. 1 An Overview of μFormer. μFormer is a two-step solution for mutational effect prediction, i.e., predicting the fitness score of a mutated protein sequence: first, we pre-train a masked protein language model (LM) using a large database of unlabeled proteins; second, we introduce three scoring modules (with a small set of new parameters) into the pre-trained protein LM for fitness score prediction and train all parameters using a set of mutated protein sequences with measured fitness scores. **Left)** Conventional masked language models for proteins model masked tokens (i.e., amino acid residues) by conditioning on unmasked tokens only, while processing each token independently. In contrast, μFormer exploits the pairwise masked language model (PMLM) which considers the dependency among masked tokens, taking into account the joint probability of a token pair. **Right)** Motivated by biological insights, we introduce three scoring modules on top of the pre-trained protein LM to predict the fitness score for a protein sequence, which focus on different granularities of the protein sequence: 1) The residual-level score which characterizes the grammatical validity of residues at each position conditioned on the entire sequence; 2) The motif-level score which aims to capture the local sequence information around a residue beyond the residue granularity, considering that motif is widely used in biological sequence modeling for different tasks; 3) The sequence-level score is motivated by the observation that the semantics of protein sequences are relevant to their properties/functions on the whole sequence granularity.

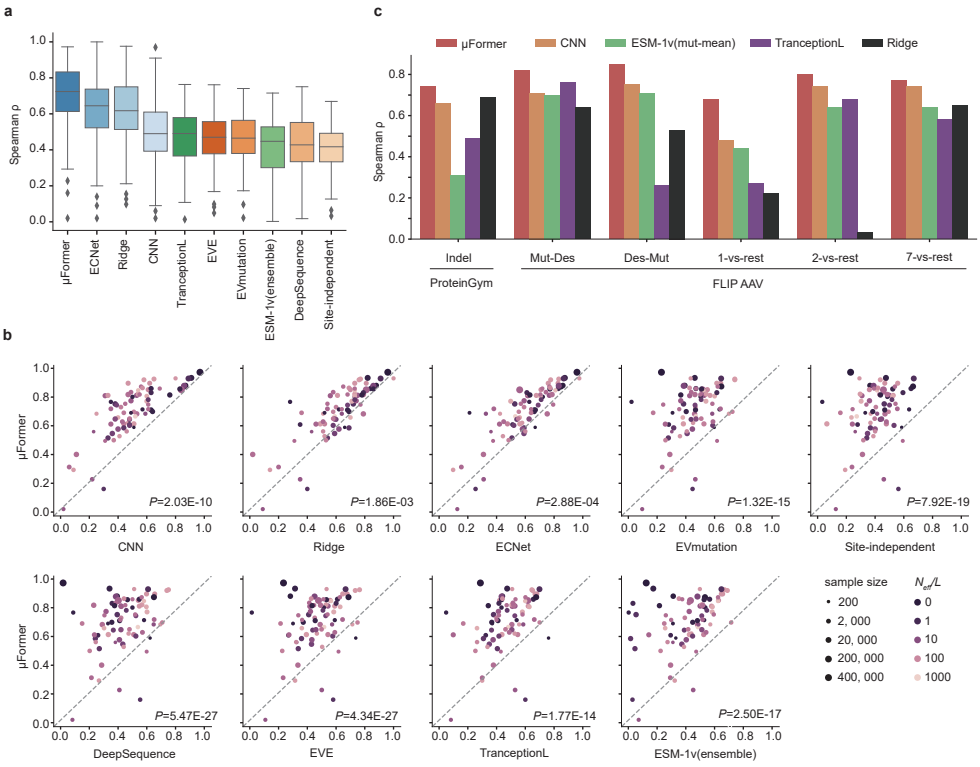


Fig. 2 Quantitative comparison of μFormer with the state-of-the-art mutational effect prediction approaches. a-b) μFormer outperforms alternative approaches on ProteinGym benchmark. a) Spearman ρ statistics on 78 ProteinGym datasets of μFormer and 9 alternative approaches. Blue: learning-based approaches, including μFormer, Ridge, ECNet, and CNN. Green: zero-shot approaches, including Tranception-L and ESM-1v. Blue: MSA-based approaches, including EVE, EVmutation, DeepSequence, and Site independent. Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; diamond points, outliers. b) Pairwise comparison between μFormer and other approaches. Each data point represents 1 ProteinGym dataset. The dot size is proportional to the size of training data logarithmically. The darkness is proportional to the number of homologous sequences of the target protein, with darker color representing less homologous sequences. Spearman ρ is used to quantify the performance of corresponding approaches. P values are computed with the one-sided rank-sum test. c) μFormer outperforms alternative approaches on indel-included benchmark datasets. ProteinGym Indel: cross-protein indel-included mutational effect dataset collected by Notin et al. [5]. FILP AAV: VP1 indel and substitution mutational effect dataset measured and processed by Bryant et al. [29] and Dallago et al. [30] respectively. Mut: naturally occurring mutants. Des: designed sequences. Mut-Des: All natural sequences are assigned to train; all designed sequences are assigned to test. Des-Mut: All designed sequences are assigned to train; all designed sequences are assigned to test. n-vs-rest: Mutants with mutation sites less than or equal to n are assigned to train, the rest of the data with high-order mutations are assigned to test.

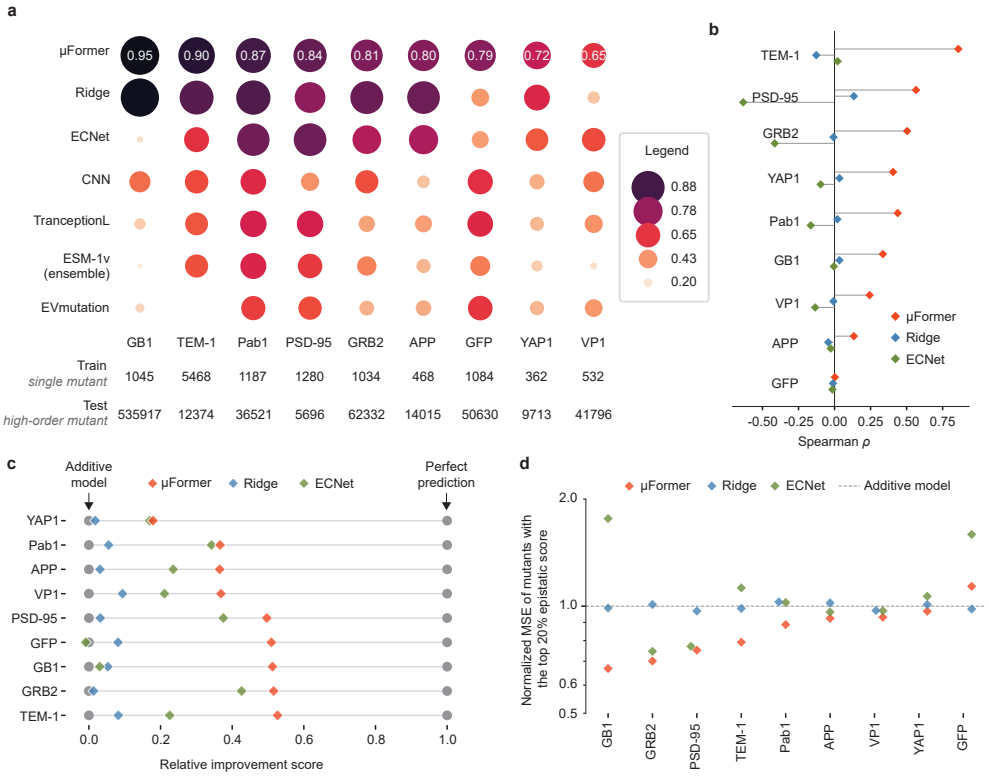


Fig. 3 μFormer models the effects of high-order mutants and epistasis effectively. a) Spearman ρ between predicted and experimentally measured effects for high-order mutants of 9 proteins with diverse functions. The size and darkness of the dots are proportional to the score. Sizes of training data and test data for each protein are listed at the bottom. b) Spearman ρ between predicted and experimentally measured epistatic scores for high-order mutants of 9 proteins with diverse functions. Red: μFormer. Blue: Ridge. Green: ECNet. c) Predicted fitness values from μFormer, Ridge and ECNet in comparison to epistasis-free estimation (additive model) and experimentally measured observations (perfect prediction). The performance of the additive model is normalized to 0, and that of perfect prediction is normalized to 1. d) Mean squared error (MSE) of μFormer, Ridge and ECNet in comparison to the additive model for predictions of high-order mutants displaying the top20% absolute epistatic scores in each protein.

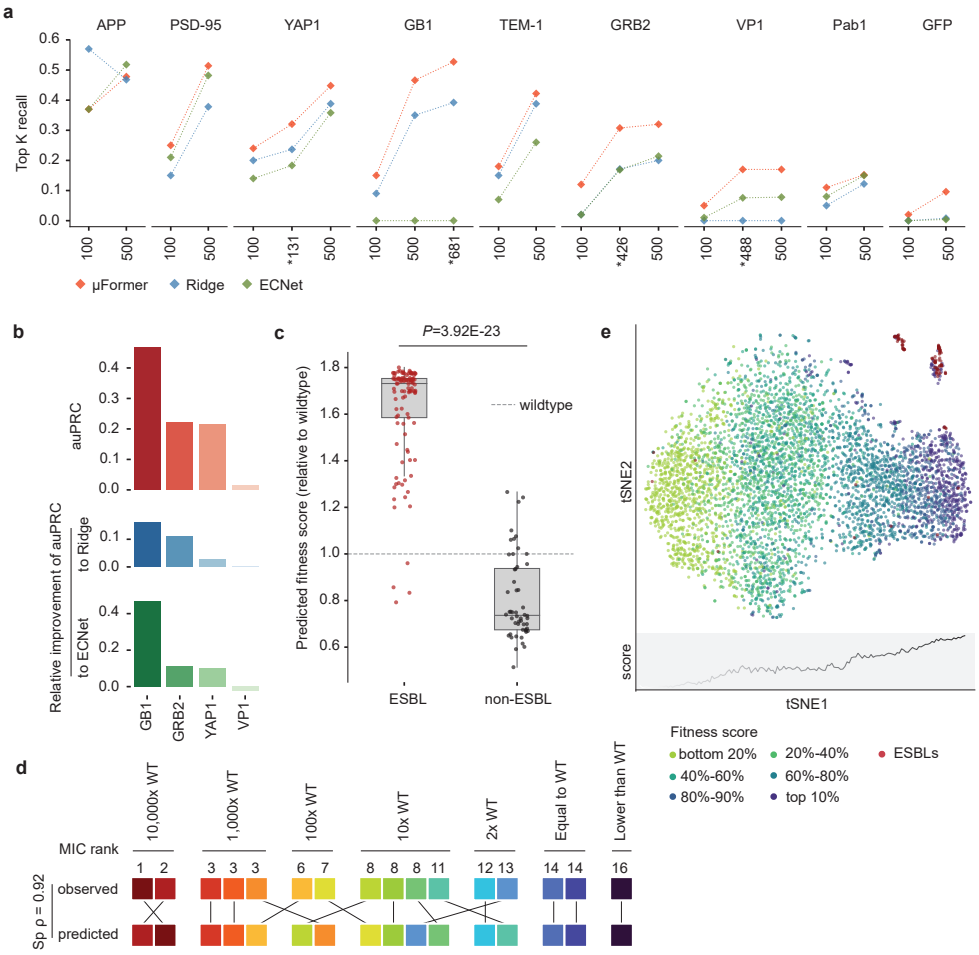


Fig. 4 μFormer effectively identifies high-functioning variants with multi-point mutations. a) Recall score of μFormer (red), Ridge (blue) and ECNet (green) for the Top K high-order mutants of 9 different proteins ranked by fitness values. K is equal to 100, 500, or customized values marked with asterisks. The customized values are determined by the number of high-order mutants with fitness values higher than all values in the training data (ultra-high value mutants). b) The area under precision-recall curve (auPRC) score for μFormer in classifying ultra-high value mutants in GB1 (n=681), GRB2 (n=426), YAP1 (n=488), and VP1 (n=131) (top). The relative improvements of μFormer compared to Ridge (middle) and ECNet (bottom) are also shown. c) The predicted fitness scores of extended-spectrum beta-lactamases (ESBLs) are significantly higher than those of non-ESBL clinical variants. The scores are normalized to wildtype (which equals 1). P values were computed with the one-sided rank-sum test. Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; diamond points, outliers. d) Predicted fitness values from μFormer are highly correlated (Spearman $\rho = 0.92$) with the minimum inhibitory concentration (MIC) measured by Weinreich et al [42]. e) t-SNE visualization of TEM-1 mutant embeddings extracted from μFormer. Red dots represent ESBLs. Other dots, colored by quantile ranks of experimentally measured fitness values, represent TEM-1 single mutants generated by Stiffler et al [41]. The bottom panel shows the average fitness score along PC1.

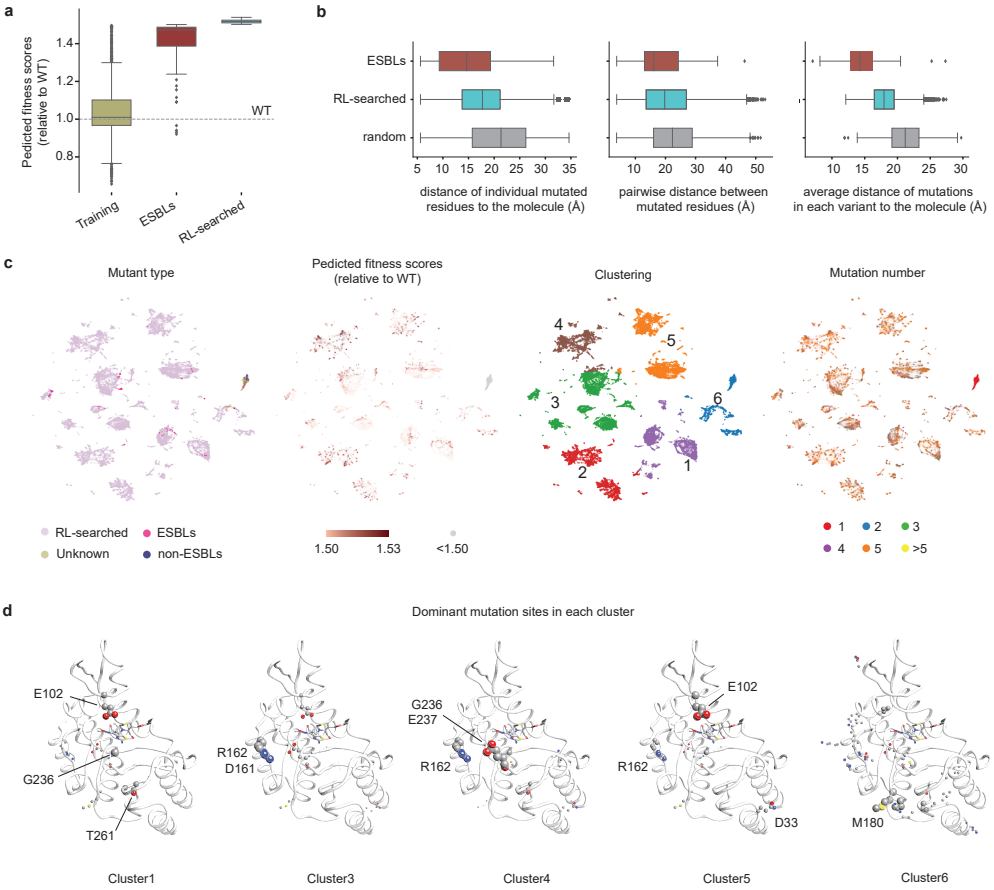


Fig. 5 Design high-functioning sequences with μFormer and reinforcement learning. a) Predicted fitness scores of single mutants (training dataset), ESBLs, and RL-searched mutants. The score of wild-type TEM-1 was normalized to 1. Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers. b) Quantitative mutation properties in ESBLs, RL-searched mutants, and multi-point mutants generated by random mutagenesis. From left to right: the distribution of distances of individual mutated residues to the docked small molecule, the distribution of pairwise distances between mutated resides, and the distribution of average distances of all mutations in each variant to the docked small molecule. Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers. c). t-SNE visualization of TEM-1 mutant embeddings extracted from μFormer. The overlaid labels, from left to right, indicate mutant types, predicted fitness scores, clusters, and the number of mutations in each variant, respectively. Clustering was performed using the K-Means clustering approach.

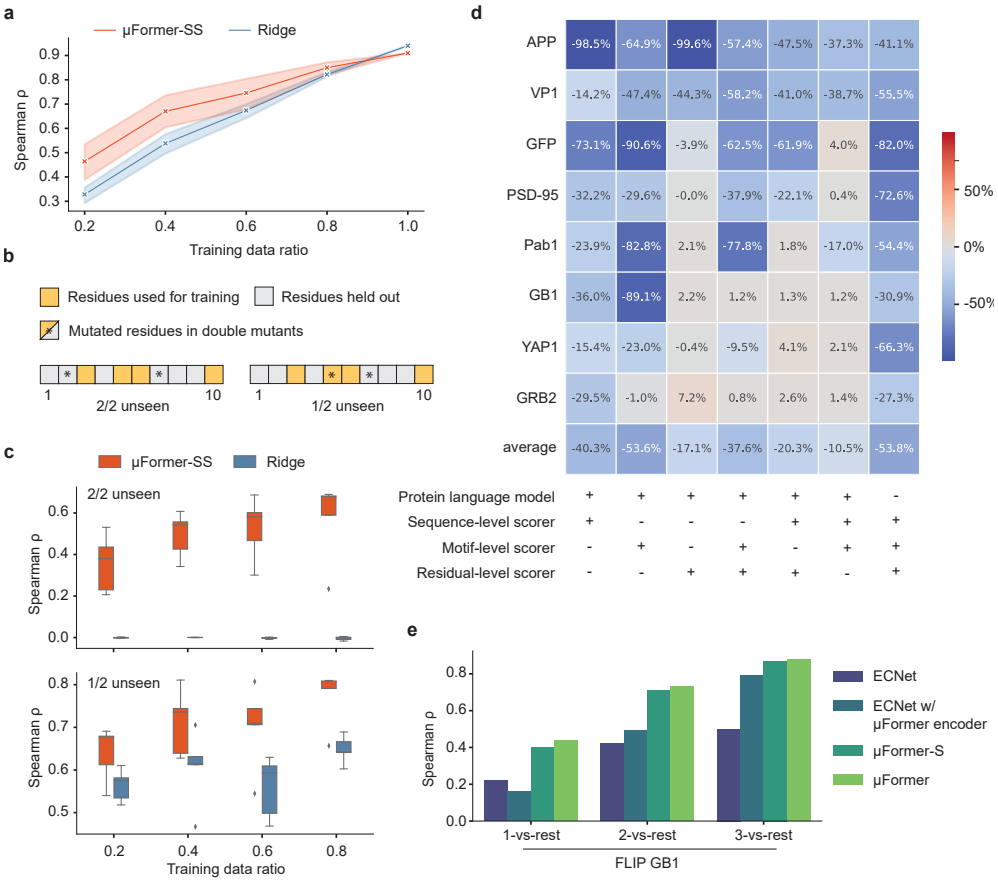
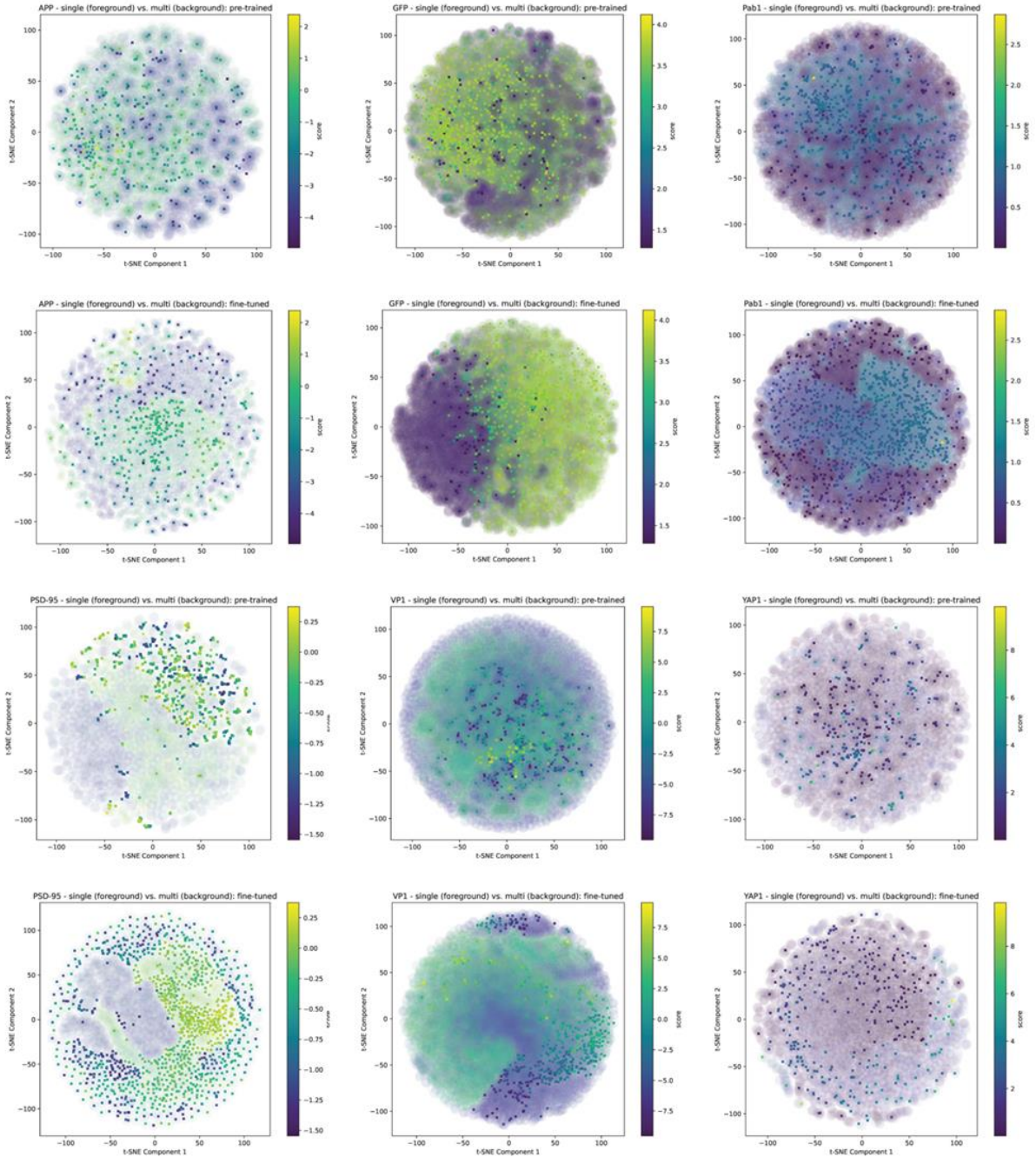
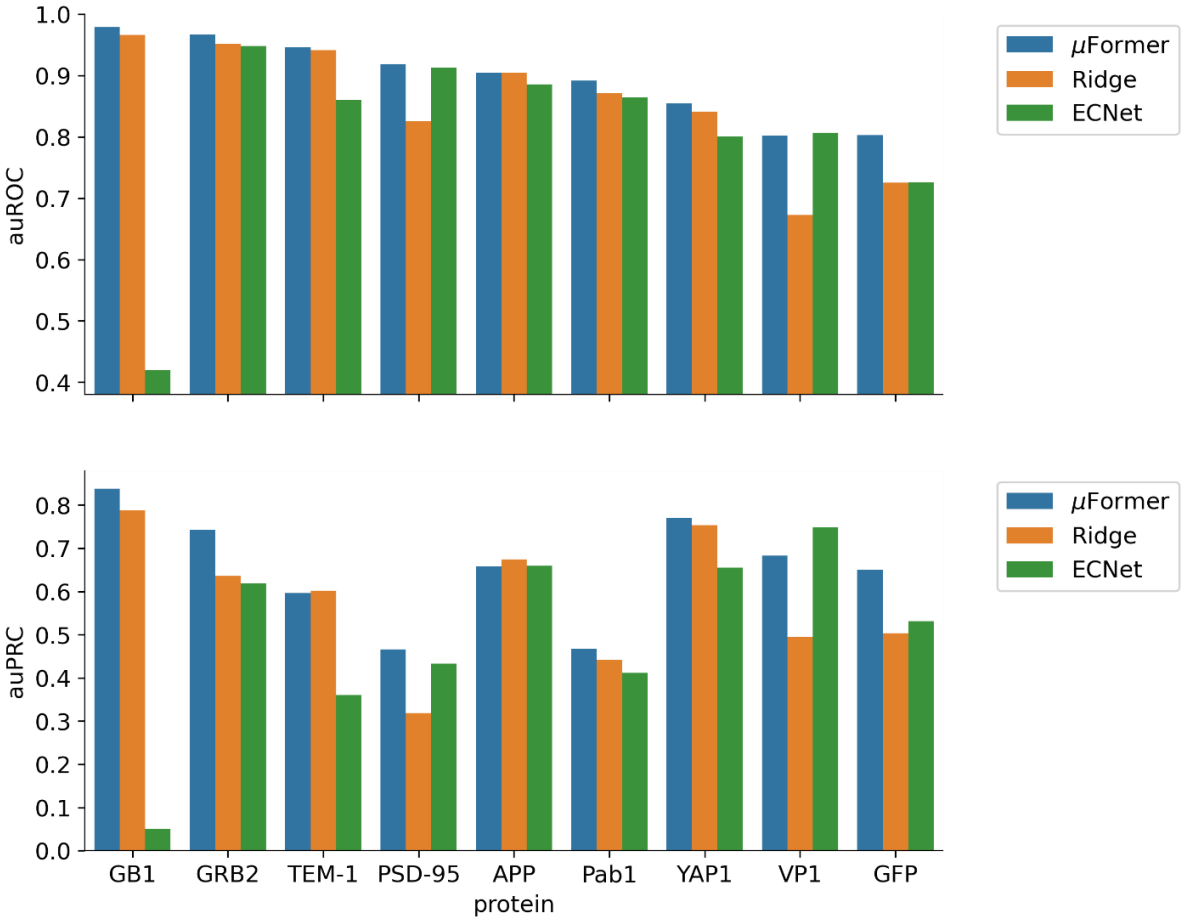


Fig. 6 Analysis of μFormer. a) Ablation study to evaluate the importance of each component in μFormer. The change in performance after removing various components from the model relative to a full model is shown. Negative numbers (blue) indicate a loss of performance and positive numbers (red) indicate an improvement in performance. The last row displays the average performance change over 9 proteins. The plus/minus signs at the bottom indicate the presence/removal of the corresponding component. b) Spearman ρ statistics on 3 FLIP GB1 datasets of μFormer, ECNet, and their variants. ECNet w/ μFormer encoder replaces the language model in ECNet with μFormer's language model. μFormer-S is a μFormer variation with a model size similar to ECNet. c) Performance of μFormer (red) and Ridge (blue) on GB1 double mutants with varying training data size. Here, μFormer is a μFormer variation with a smaller supervised scorer module size (μFormer-SS). Training data ratio indicates the number of residues used for training versus the total number of amino acids in GB1. The training data size equals 209, 418, 627, 836, and 1045 for 20%, 40%, 60%, 80%, and 100%, respectively. All scores were evaluated on GB1 saturated double mutants (n=535, 917). Shades: standard deviation. d) Illustration of test data split, using a protein of 10 residues and the 40% setting as an example. 2/2 unseen: neither of the mutated residues in double mutants are seen by the model. 1/2 unseen: one and only one of the mutated residues in double mutants are seen by the model. e) Performance of μFormer (red) and Ridge (blue) on different splits of GB1 double mutants. Training data split criteria are the same as in c). Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers.

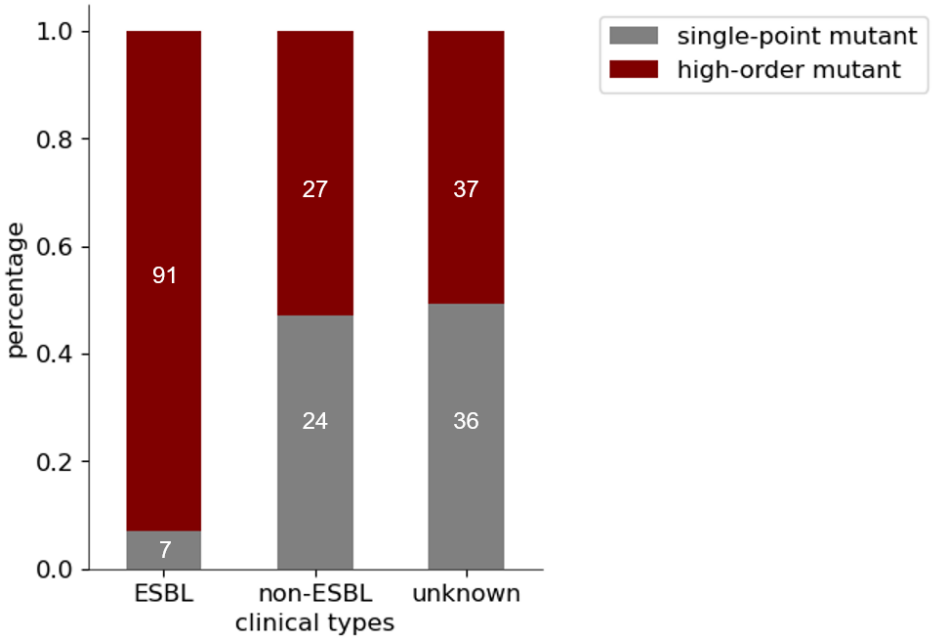


Extended Data Figure 1. Visualization of embeddings on single mutants and high-order mutants (muti mutants) using t-SNE. First row: Illustration of embeddings on APP, GFP, and Pab1 extracted from pre-trained μFormer model. Second row: Illustration of embeddings on APP, GFP, and Pab1 extracted from fine-tuned μFormer model. Third row: Illustration of

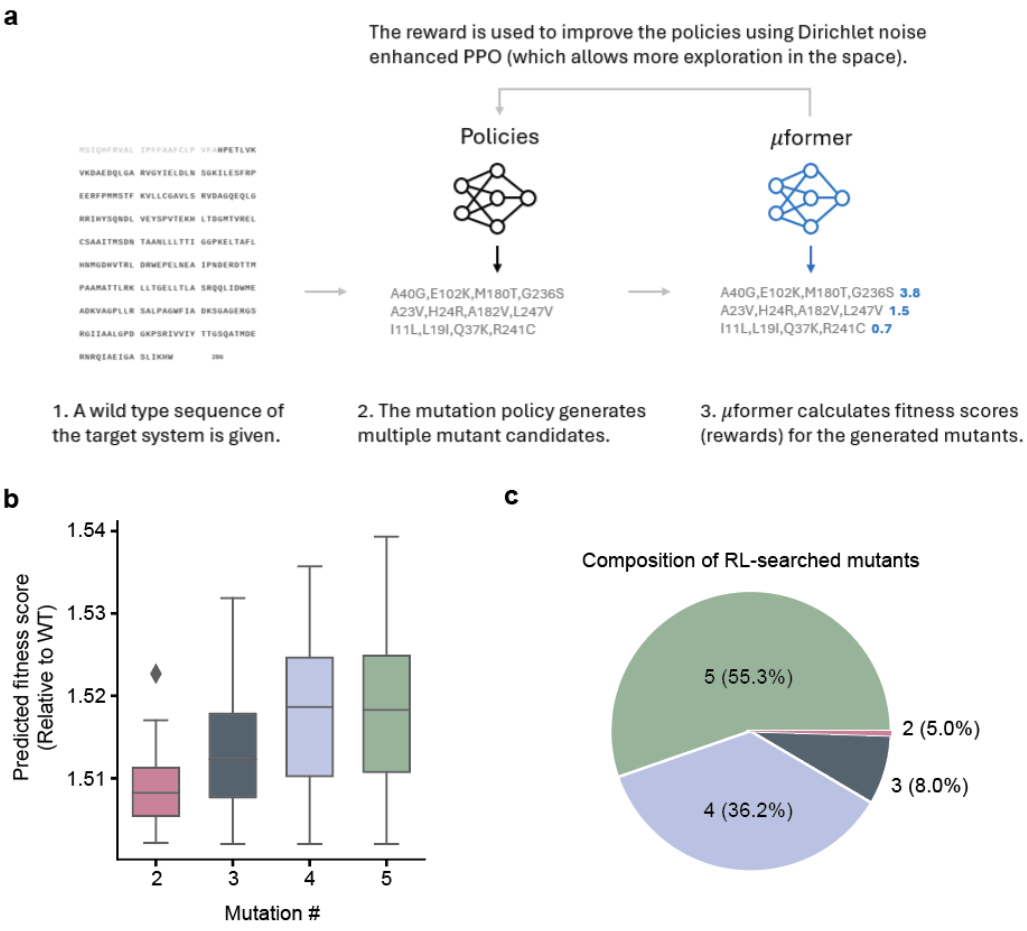
embeddings on PSD-95, VP1, and YAP1 extracted from pre-trained μFormer model. Second row: Illustration of embeddings on PSD-95, VP1, and YAP1 extracted from fine-tuned μFormer model. The color scheme illustrates the fitness scores, with lighter shades indicating higher scores. Blurred background markers represent high-order mutants, while clear foreground markers denote single mutants.



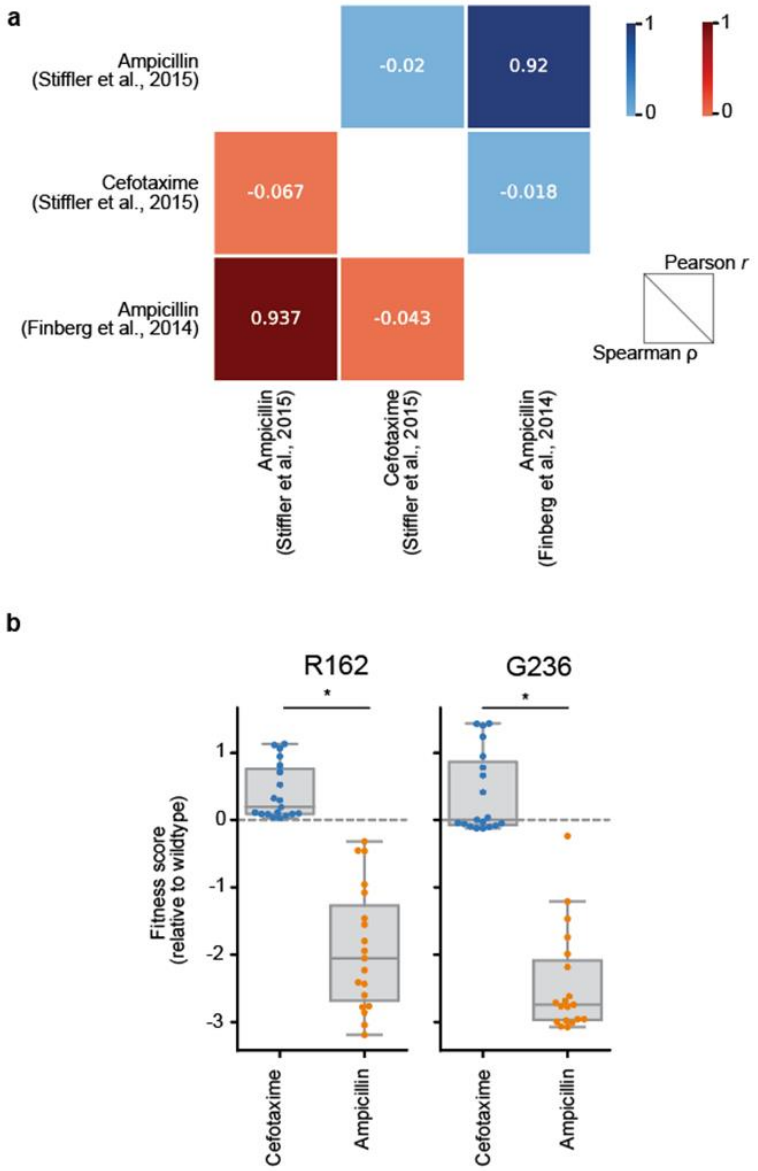
Extended Data Figure 2. μFormer accurately distinguish gain-of-function mutations from loss-of-function mutations. Upper: the area under the receiver operating characteristic (auROC) score for different models on various proteins. Bottom: the area under precision-recall curve (auPRC) score for different models on various proteins.



Extended Data Figure 3. Percentage of single-point mutants and high-order mutants in each clinical type of TEM-1. The count number for each group is marked on the plot.



Extended Data Figure 4. Reinforcement learning supports efficient and comprehensive exploration of the vast protein mutant space. a) Diagram of reinforcement learning-based search pipeline. **b)**. Predicted fitness scores of *n*-point mutants. The score of wildtype TEM-1 is normalized to 1. Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers. **c)** Composition mutants recovered by reinforcement research. The number *n* indicates the number of mutations in each mutant.



Extended Data Figure 5. Correlation between TEM-1's mutational effects on ampicillin and cefotaxime is weak. a) Spearman ρ and Pearson *r* representing the correlation between TEM-1's mutational effects on cefotaxime and ampicillin. The data on ampicillin was collected from 2 different studies. **b)** For residues R162 and G236, the same set of mutations exhibits contrasting effects on protein's activity against cefotaxime and ampicillin.