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] and Tie-Yan Liu[1]

[1]Microsoft Research AI for Science.

*Corresponding author(s). E-mail(s): liang.he@microsoft.com; pan.deng@microsoft.com;
Contributing authors: 767780791shr@gmail.com; guoqingliu@microsoft.com; haiguang.liu@microsoft.com; chuancao@microsoft.com; fusongju@microsoft.com; lijuwu@microsoft.com; taoqin@microsoft.com; tie-yan.liu@microsoft.com;

Abstract

Protein engineering holds significant promise for designing proteins with customized functions, yet the vast landscape of potential mutations versus limited lab capacity constrains the discovery of optimal sequences. To address this, we present the $\boldsymbol{\mu}$Protein framework, which accelerates protein engineering by combining $\boldsymbol{\mu}$Former, a deep learning model for accurate mutational effect prediction, with $\boldsymbol{\mu}$Search, a reinforcement learning algorithm designed to efficiently navigate the protein fitness landscape using $\boldsymbol{\mu}$Former as an oracle. $\boldsymbol{\mu}$Protein uniquely leverages single-point mutation data to predict optimal sequences with complex, multi-point mutations through its modeling of epistatic interactions and a two-step, multi-round search strategy. In benchmark testing, $\boldsymbol{\mu}$Protein achieved state-of-the-art results, and in wet-lab experiments, it identified high-gain-of-function multi-point mutants for the enzyme $\boldsymbol{\beta}$-lactamase, including variants that increased bacterial growth rate by up to 2000-fold, surpassing the highest known activity level, all while training solely on single-site mutation data. These results demonstrate $\boldsymbol{\mu}$Protein's

capability to discover impactful mutations across vast protein sequence space, offering a robust, efficient approach for protein optimization.

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

1 Introduction

Protein engineering, a cornerstone of biotechnology, focuses on designing proteins with customized functions that advance fields such as technology, agriculture, and medicine [1–5]. By optimizing protein sequences, this field holds vast potential for developing biologic drugs, novel enzymes, and other biotechnological innovations. Central to this endeavor is accurately mapping protein sequences to their functions, which forms the basis for identifying sequences with desired traits. This mapping, known as the fitness landscape, is complex and rugged, as the relationship between protein sequences and functions involves intricate interactions, or epistasis, between amino acid residues.

While high-throughput experimental techniques like Deep Mutational Scanning (DMS)[6] and Multiplex Assays of Variant Effects (MAVEs)[7] have significantly improved our ability to measure fitness for protein variants, they are limited by time and cost. These methods often restrict experiments to single-point mutations, which leaves the combinatorial complexity of multi-mutation variants largely unexplored [4, 8–10]. Predicting the effects of high-order mutation combinations—where the sequence space grows exponentially with each added mutation—is therefore an important goal of fitness landscape modeling. Computational models [4, 9, 11–15] offer a solution by predicting fitness from limited data. However, approaches based on evolutionary insights from Multiple Sequence Alignments (MSAs)[1, 2, 16–18] and protein language models (PLMs) trained on large-scale datasets[5, 18–24] have shown promise but often fall short when dealing with proteins lacking homologs or when capturing specific property-based landscapes.

To address these limitations, we introduce μProtein, a novel framework that combines two synergistic components: μFormer, a deep learning model designed to capture complex epistatic effects accurately, and μSearch, a reinforcement learning-based algorithm for efficiently exploring vast sequence spaces using μFormer's predictions. Together, these components provide a comprehensive, efficient approach to identifying high-fitness mutations in complex fitness landscapes, thus expanding the scope of protein engineering.

Within this framework, μFormer uses a pairwise masked language model and three specialized scoring modules to model fitness landscapes effectively, even for complex scenarios like multi-point mutations and orphan proteins. μSearch complements this by framing protein engineering as a decision-making task, using Proximal Policy Optimization (PPO) with exploration noise to

navigate the rugged fitness landscape, allowing μProtein to outperform existing models and guide experimental validation successfully.

Applying this pipeline, we successfully engineered a β-lactamase enzyme with enhanced activity against cefotaxime, a substrate that has not been extensively explored in protein engineering tasks. By leveraging single-site mutation data and exploring multi-site, high-fitness mutations, we identified 47 variants with improved activity over the wild type, from a pool of 200 samples tested. Notably, some variants exhibited growth rates up to 2000 times higher, highlighting the effectiveness of μProtein in identifying high-performance protein variants and showcasing its potential to advance protein engineering and drug discovery.

2 Results

2.1 Overview of μProtein

μProtein is designed to efficiently explore the vast fitness landscape of proteins and generalize from limited experimental observations, which are typically confined to single-point mutations (Fig. 1a). The framework combines μSearch and μFormer. Given a wild-type sequence, μSearch, the reinforcement learning framework, autonomously searches the multi-point mutational landscape by leveraging μFormer, a pre-trained oracle model that scores the effects of sequences proposed by μSearch (Fig. 1b).

Reinforcement learning allows μSearch to frame protein engineering as a decision-making task, selecting optimal mutations to maximize sequence fitness based on a reward function. Using Proximal Policy Optimization (PPO) with Dirichlet noise to avoid local optima, μSearch navigates the rugged fitness landscape by learning mutation site and type policies through two independent neural networks (Fig. 1c). A multi-round search strategy enables μSearch to propose multi-site mutations.

μFormer, acting as the oracle, provides reward scores for μSearch. μFormer is built from a self-supervised protein language model and three supervised scoring modules (sequence-level, motif-level, and residue-level). It first undergoes pre-training on over 30 million protein sequences from UniRef50 [25] (Fig. 1d). Using pairwise masked language modeling [26] (Fig. 1d, right), the model learns to predict amino acid identities at masked positions based on not only their surrounding residues but also the likelihood of interactions between residue pairs within the sequence. During fine-tuning, the full-length sequence embeddings, residue embeddings, and amino acid probabilities are passed to the three scoring modules (Fig. 1d, left), with the outputs of the three scoring modules 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 protein fitness landscape with improved accuracy and robustness, offering a reliable reward function for μSearch.

While pre-training on diverse proteins and mutational datasets allows μFormer's to generalize to across proteins, experimental data, such as DMS data specific to the protein of interest, can further fine-tune μFormer for improved protein-specific predictions (Fig. 1b).

We validated the μProtein framework through extensive benchmark studies and wet-lab optimization of β-lactamase, showcasing its efficacy in pinpointing high-performance mutants in a broader protein mutational space (Fig. 1e). We believe that μProtein holds broad applicability across diverse protein engineering tasks, addressing both industrial and therapeutic challenges, including applications such as antibody design [27], enzyme optimization [28], and the prediction of drug-resistant mutations [29, 30] (Fig. 1f).

2.2 μFormer accurately predicts fitness scores of proteins

We started by assessing the ability of μFormer in fitness landscape modeling and mutational effect prediction. To this end, we benchmarked it against ten alternative approaches, including MSA-based methods [1, 16–18], language model-based zero-shot methods [5, 24] and learning-based approaches [13–15]. 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 mean Spearman's rank correlation coefficient (ρ) of 0.703, μFormer achieved the best performance on 48 out of 78 test 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 [31, 32]. 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 [32]. μFormer achieves a Spearman ρ above 0.8 for both settings, signifying its effectiveness as a surrogate oracle for navigating protein fitness landscape and guiding protein sequence design.

2.3 μFormer captures epistatic mutational effect of high-order variants

We next assessed whether μFormer can generalize to multi-point mutants, i.e. high-order variants, while training on single-point mutants only (single-to-multi).

We collected nine sets of DMS assays that include both single-point mutants and high-order mutants and trained prediction models with single-point data for each target protein. The nine proteins span a wide range of functions, including enzymes (β-lactamase TEM-1 [33]) and reporter proteins (GFP [34]) of high interest in synthetic biology, disease-related proteins (Amyloid beta precursor protein APP [35]), signaling proteins (PSD-95 [36], GRB2 [36] and YAP1 [37]), RNA binding proteins (Pab1 [38]), and protein binding proteins (GB1 [39]). 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 can be much greater (Fig. 3a). For instance, while there are 1,045 single-site GB1 mutants for training, the high-order mutants to be tested reached a number of 535,917. 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, 40, 41]. 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.

2.4 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 Data 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 pred-0ictions 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, reducing the antibiotic activities. 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) [42], 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 quantification of TEM-1 single mutants on cefotaxime collected by Stiffler et al. [43]. Then, we evaluated μFormer using a curated dataset comprising 105 ESBLs and 52 confirmed non-ESBLs [42, 44]. 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) [44].

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 [45, 46]. Moreover, when ranked by fitness values, the top 1% (50) TEM-1 mutants from a saturated single mutant DMS study [43] 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.

2.5 Design high-functioning sequences using μProtein

We employed the TEM-1-cefotaxime system to assess μProtein's effectiveness in guiding protein optimization. Using μFormer as the reward function, μSearch explored billions of possible TEM-1 variants containing 2-3 point

mutations, aiming to identify those with enhanced activity against cefotaxime (Extended Data Fig. 4, Methods).

From the sequences explored by the RL search, we identified over 1,200 variants and selected the top 200 predicted variants for validation through *E. coli* growth assays. Additionally, we included 50 randomly generated variants with 2-3 point mutations and 15 variants with 1 to 4 point mutations spanning A40G, E102K, M180T, and G236S, previously tested through minimum inhibitory concentration (MIC) assays [44], for comparative analysis.

Our experiments revealed that *E. coli* carrying 47 different RL-designed TEM-1 variants (23.5% of the RL-designed mutants tested) exhibited superior growth on cefotaxime compared to the wild-type strain (Fig. 5a), indicating heightened enzymatic activity against cefotaxime. In contrast, only 12% of randomly generated TEM-1 variants displayed enhanced activity. This is consistent with previous observations that TEM-1 mutations often increase activity against cefotaxime [43]. Furthermore, RL-designed variants demonstrated greater activity gains compared to random-generated variants (Fig. 5b). Notably, a double-mutation combination, G236S;T261V, surpassed the activity level of the well-known quadruple mutant A40G;E102K;M180T;G236, Swhich exhibits significant activity against cefotaxime relative to wild-type TEM-1 [44]. While G236S has been linked to moderate cefotaxime activity gain in previous studies [44], T261I is a novel mutation not previously reported.

A comparison of relative growth rates revealed a weak correlation between growth in the presence and absence of cefotaxime for hyperactive mutants (Fig. 5c). Ultra-high activity mutants (bottom-right corner of the plot) tended to exhibit reduced growth in the absence of cefotaxime, consistent with the understanding that beneficial mutations under selective pressure may be detrimental in neutral conditions. However, some strains, such as G236S;T261V;K284N, showed growth advantages under both treated and untreated conditions (Fig. 5c).

Notably, the mutation distribution in RL-designed strains differed significantly from that of natural ESBL strains (Fig. 5d). The RL-designed mutants exhibited a more diverse range of mutations. For instance, while the G236S mutation dominates in natural strains, RL-designed mutants frequently featured asparagine and alanine substitutions at G236. Similarly, while T261M is observed in ESBL strains, RL-designed mutants more commonly exhibited isoleucine, valine, and alanine substitutions. All these mutations may alter the hydrogen bond network and, consequently, affect the structure of TEM-1, particularly around arginine at position 241, which is located near the cefotaxime binding pocket and likely influences TEM-1's activity against the antibiotic (Fig. 5e).

Two main factors likely contribute to the observed differences in mutation distribution. First, mutations such as T261A, T261I, and T261V exhibit relatively high activity gains compared to other T261 mutations (Fig. 5f), suggesting that our method favors single-point mutations with positive effects. Second, natural mutants typically arise from single nucleotide transitions (A-T

or C-G mutations), which occur more frequently than transversions *in vivo*. In contrast, RL-designed mutants can explore a broader mutational space, including two-nucleotide mutations (Fig. 5g and Extended Data Fig. 5). Therefore, by combining in silico fitness landscape modeling with μSearch, we are able to design protein variants that may not be achievable through natural evolution or selection. Given that TEM-1 mutants are common pathogenic proteins associated with drug resistance, our findings offer valuable insights into potential future mutations that could arise under prolonged selective pressure.

In conclusion, by leveraging an effective fitness prediction model and an efficient reinforcement learning framework, we successfully navigated the vast mutant space of TEM-1 and identified high-order variants with significantly enhanced activity against cefotaxime. These results represent a significant advancement in exploring and manipulating rugged fitness landscapes.

2.6 μProtein enables generalization to novel 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 [39] 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.

2.7 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 [32], 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 mutational effect modeling.

3 Discussion

In this study, we present μProtein, an integrated framework that leverages the deep learning model μFormer and the reinforcement learning framework μSearch to accurately model and navigate protein fitness landscapes. Our results demonstrate that μProtein achieves superior performance in predicting mutational effects, particularly in challenging cases such as high-order mutations and indels, where most current approaches fall short. This capability addresses key limitations in protein engineering, such as capturing epistatic interactions and generalizing from single-point to multi-point mutations. Together, these advancements offer a powerful tool for protein optimization in fields ranging from synthetic biology to drug discovery.

Our framework addresses some of the limitations of existing models. Current approaches, including those based on multiple sequence alignments (MSAs) and zero-shot language models, often struggle with proteins lacking abundant homologs or with accurately capturing epistatic effects [5, 24]. Unlike these methods, μFormer integrates pre-trained language model representations with specialized scoring modules that enhance its adaptability to protein sequences with limited or sparse experimental data. Furthermore, by fine-tuning with DMS data specific to the protein of interest, μFormer achieves improved predictive accuracy for high-order mutants, outperforming zero-shot approaches and helping to avoid unfavored sequences, a limitation previously noted in similar models [4]. This improvement highlights the effectiveness of phenotype-aware fine-tuning, particularly when applied to proteins like TEM-1, where mutational effects can differ significantly across related phenotypes, such as resistance to different beta-lactam antibiotics [43](Extended Data Fig. 6).

The practical capabilities of μProtein are demonstrated through its application to the TEM-1 beta-lactamase enzyme. Fine-tuning μFormer on single-point mutation data enabled accurate predictions of high-functioning TEM-1 variants against cefotaxime, a beta-lactam antibiotic resistant to the wild-type enzyme. Among high-order mutants, μFormer accurately prioritized mutants from clinical isolates exhibiting resistance, and it successfully identified novel combinations, including a G236S;T261V variant, which exceeded the activity levels of known high-resistance mutants. These results underscore μProtein's potential to guide the identification of clinically relevant mutations, offering insights into evolutionary trajectories that could inform the development of new antibiotic treatments or anticipatory measures against resistance.

Our results also suggest broader applications for μProtein in protein design. The accurate predictions of high-functioning, multi-point mutants illustrate its applicability to diverse engineering tasks, including antibody engineering, enzyme optimization, and peptide therapeutic design. The ability of μProtein to explore multi-site mutations across a large mutational space, even when trained on single-point data, further highlights its utility for industrial and therapeutic applications where protein optimization is essential.

While μProtein exhibits strong performance, there are several directions for improvement. First, integrating 3D structural data could enhance its capacity to capture spatially proximate but sequentially distant epistatic effects, which sequence data alone may not capture as effectively. This enhancement could improve the predictive power of μFormer in cases where structural considerations are particularly relevant, such as antibody design and the engineering of enzymes with highly specific active sites [47]. Additionally, expanding μProtein's training across a range of phenotypes and environmental contexts could enhance its generalization capabilities, potentially allowing for phenotype-aware modeling that supports multitarget fitness predictions within a unified framework. For example, encoding specific phenotypes or environmental factors (e.g., different substrates or conditions) within the model could facilitate simultaneous prediction across related proteins, further extending the utility of μProtein in practical applications.

A consideration for future work is the potential constraint posed by pre-trained language models on sequence length. For proteins with longer sequences (>1024 residues), extending the model's context capabilities could enhance μProtein's performance. Moreover, building models that can predict mutational impacts across a continuum of sequence lengths would be advantageous, especially for proteins with functional flexibility dependent on variable regions or motifs. Exploring efficient architectures that can support such extended context lengths is an important direction for advancing μFormer's adaptability.

Overall, μProtein provides a novel and versatile framework for protein engineering by integrating a powerful fitness prediction model with an efficient reinforcement learning strategy. Our work demonstrates μProtein's potential to significantly enhance protein design efforts, bridging the gap between

sequence prediction and functional optimization. Achieving state-of-the-art performance across a variety of complex tasks, μProtein stands as a valuable tool for achieving tailored protein functionalities, opening new possibilities in biotechnological innovation and discovery.

4 Methods

4.1 Fitness Landscape Modeling via μFormer

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 Pairwise Masked Language Model for Proteins

The masked language model (MLM) [19, 48] is a self-supervised learning technique that utilizes the Transformer encoder [49] 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, we design a novel **pairwise masked language model** (PMLM) for protein sequence pre-training, it considers the dependency among masked tokens, taking into account the joint probability of a token pair [26]. 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:

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

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 $S_{\text{RESI}}(x)$, which characterizes the grammatical validity of residues at each position conditioned on the entire sequence [22, 24, 50];
- The motif-level score $S_{\text{MOTIF}}(x)$, 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, 48, 51];

- 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 [21, 22, 50, 52, 53].

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:

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

where L is the length of the sequence x and $p_\theta(x_i \mid x)$ 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 [21, 24] 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 [48, 51]. 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 [21, 52] 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. 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 [32]. 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 20% 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 [24], 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 [32]. 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 [15] and the CNN [32] models, we used the model architecture and hyperparameters implemented by the FLIP codebase [32]. While for ECNet [14], we reproduced the ECNet model using the open-sourced code provided by Luo et al [14]. Since ECNet both leverages MSAs and labeled data, we retrained the models under the same settings while with the prepared MSAs. We also fine-tuned the ESM-1b models as done in the work [15].

4.2 Fitness Landscape Navigation by μSearch

To explore the vast space of mutations and efficiently identify high-functioning mutants in the protein landscape, we develop a RL method called μSearch, using the μFormer model as a reward function. Similar to previous work [54, 55], we frame the protein engineering problem using a Markov Decision Process (MDP). In our case, the state represents the current mutant sequence, the action signifies a one-point mutation, including the selection of mutation sites and types to mutate within the sequence, and the reward function is given by the μFormer model. During each episode, the agent sequentially mutates a single-site residue until it arrives at a predetermined horizon. The agent's learning goal is to master the optimal selection of mutation sites and types for the desired properties. This learning process alternates between two phases: exploration and updating. During the exploration phase, the agent navigates the mutation space, utilizing a mutation site policy network and a mutation type policy network to generate potentially high-performing mutants. In the updating stage, the μFormer model is used to label the newly produced mutants, and the policy network is driven by the policy loss to propose mutants with superior fitness scores.

4.2.1 PPO Algorithm with Mutation Site and Mutation Type Policy Networks

In our MDP definition, each action corresponds to a single-point mutation which includes the selection of mutation sites and types within the sequence. Therefore, we have designed two distinct policy networks. The first, referred

to as the mutation site policy network, uses both the original (wild-type) sequence and the current mutant sequence as inputs. These inputs are represented by a binary matrix of size $20 \times 2L$[1]. This network identifies the positions for modifications in the current mutant sequence. The second network, named the mutation type policy network, considers the wild-type sequence, current mutant sequence, and the selected mutation site, represented by a binary matrix of size $20 \times 2L + L$. This network determines the specific modifications to be made at the selected mutation site.

We employ the Proximal Policy Optimization (PPO) algorithm [56] to optimize both policy networks, a method recently utilized to train large language models with human preference-based reward models [57, 58]. To reduce the variance of the policy gradient estimate, we incorporate two value networks that estimate the future rewards from a given state. The policy and value networks are updated by the following loss functions:

$$\begin{aligned} &L_{policy}(\theta) = \mathbb{E}_t[\min(r_t(\theta) \cdot A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t)], \\ &\text{where } r_t(\theta) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}, A_t = S(x) - v_\phi, \\ &L_{value}(\phi) = (v_\phi - S(x))^2. \end{aligned} \tag{10}$$

t stands for the timestep within an episode, $r_t(\theta)$ denotes the probability ratio of the current policy $\pi(\theta)$ to the previous policy $\pi_{old}(\theta)$. θ represents the policy network parameters, and ϕ represents the value network parameters. $S(x)$ indicates the μFormer score of generated mutant x. Both the mutation site and mutation type policy networks use the mentioned loss function to update their parameters.

The value function, employed as a baseline or critic, assists in calculating the advantage function A_t. This function evaluates the relative value of each action by identifying the difference between the actual return, $S(x)$, and the expected return derived from the value function. The advantage function helps the policy network differentiate superior actions from the rest, thereby adjusting its probability correspondingly. The policy loss $L_{policy}(\theta)$ is designed to increase the probability of actions with positive advantage functions, while avoiding substantial deviations from the previous policy to prevent performance collapse. The ϵ hyperparameter controls the size of the trust region for each policy update.

4.2.2 Incorporating Dirichlet Noise into Exploration

The mutation space of a protein sequence is vast, with high-profile mutants potentially spread widely within it. To navigate this space, we integrate Dirichlet exploration noise during the mutation generation phase, which helps prevent getting trapped in local optima. Specifically, the Dirichlet noise supplements the probabilities derived from the policy networks, resulting in $P(s, a) = (1 - \epsilon) \cdot \pi_\theta(a \mid s) + \epsilon \cdot \eta$, where η follows a $Dir(0.03)$ distribution

[1] L denotes the length of the protein sequence, and 20 means the number of types of amino acid.

and ϵ is set to 0.25. This added noise ensures that all potential moves can be explored by the agents, while still prioritizing mutations with superior fitness scores from μFormer. During the learning phase, the injected Dirichlet noise may lead to an action with a low probability from the policy network π_θ, potentially causing the original importance ratio to explode. To mitigate this issue, we introduce a new ratio $r_t(\theta) = \frac{P(s,a)}{P_{old}(s,a)}$, based on the mixed distribution, to facilitate a smoother training process.

4.2.3 Searching Using Multiple μFormer Models and Ranking candidates

We utilized the TEM-1-cefotaxime system to evaluate the effectiveness of combining μFormer and μSearch. Our goal was to identify TEM-1 variants with 2-3 point mutations demonstrating improved activity against cefotaxime. To achieve this, 6 μSearch searches were independently conducted using different μFormer models. For every search, we set a threshold using the median predicted value of known ESBL mutants, as determined by the currently used μFormer model, and gathered all mutants surpassing this threshold. We aggregated all the potential high-function mutants into a comprehensive list. Subsequently, to determine the final ranking of the mutants on the list, we computed the predicted value for each mutant using the six μFormer models, and determined their relative rank in the list. The final rank was derived from the aggregation of these relative ranks and was used for selection in subsequent wet-lab experiments.

4.3 *E. coli* growth assay

4.3.1 library construction

The plasmid containing the wild-type TEM-1 sequence (pSkunk3-TEM-1) was generously provided by the Ostermeier lab. Amino acid substitutions proposed by μFormer were translated into corresponding DNA mutations, taking into account the codon preference in *E. coli*, and introduced into the plasmid by Suzhou Silicon BioScience. Engineered plasmids were verified by Sanger sequencing to confirm the presence of the intended mutations before being transformed into the SNO31 *E. coli* strain harboring pTS42, which was also provided by the Ostermeier lab. To facilitate downstream sequencing, a unique 6 bp strain-specific barcode was incorporated into each TEM-1 mutant construct.

4.3.2 Antibiotic Selection

Following established protocols from previous studies [43, 59, 60], bacterial cultures carrying various TEM-1 mutant plasmids were grown overnight to saturation at 37°C, then diluted to an OD_{600} of 0.01. These cultures were seeded onto 180mm petri dishes containing LB media supplemented with either 0μg/ml or 0.2μg/ml cefotaxime sodium salt (TargetMol , CAS Number:

63527-52-6). In the initial round, around 50 different constructs were seeded onto the same plate, including the wild-type construct as a control. To ensure robustness, constructs were randomized across two different plates, each with distinct neighboring constructs. After 36-48 hours of incubation at 37°C, bacterial colonies were harvested. The bacteria from each plate were collected into separate tubes for subsequent analysis and sequencing. The top 20 strains (described below) were selected for a second round of experiments where each variant, along with the wild-type, was individually seeded onto 90mm plates.

4.3.3 Sequencing and anlysis

Paired-end primers were designed to sequence the regions flanking the 6 bp barcode on the plasmid. A 3 bp sequencing barcode was incorporated into the sequencing primers to differentiate between plates. Barcoded PCR amplicons were prepared from each plate, pooled, and subjected to paired-end second-generation sequencing. Sequencing raw data was cleaned by removing adapters, trimming reads at the front and tail, and filtering out reads with $> 40\%$ unqualified reads ($< Q15$) using fastp (version 0.23.4). Paired reads were then merged based on at least 30bp of overlap. Sequence alignment was performed by minimap2 (version 2.26) with the following parameter settings: "-a –sr -t 4 –cs -m20 –heap-sort=yes –secondary=no". Aligned data were checked for the presence of the corresponding 6 bp barcode at the correct position. Only sequences with an exact match to the barcode were included in the downstream analysis. The relative abundance of mutant strains compared to tehe wild-type on each plate (log(FoldChange)) was used as a surrogate measure for the activity of TEM-1 mutants against cefotaxime.

The antibiotic selection, sequencing, and analysis were performed by Suzhou Silicon BioScience.

Data Availability.
The FLIP benchmark is publicly available through: https://github.com/J-SNACKKB/FLIP/. The ProteinGym data collections can be accessed via https://github.com/OATML-Markslab/ProteinGym[2]. The list of extended spectrum beta-lactamases can be found on http://bldb.eu/([42]).

Code Availability.
The code to reproduce the results of the paper is publicly available at the following URL: https://github.com/MutonAI/Mu-Former. It will be migrated into an official release later, with notice on the git repository.

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** (1), 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** (10), 816–822 (2018) .

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

[2] Retrieved on 12/30/2022

[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** (4), 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** (8), 801–807 (2014) .

[7] Weile, J. & Roth, F. P. Multiplexed assays of variant effects contribute to a growing genotype–phenotype atlas. *Human Genetics* **137** (9), 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** (22), 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** (48), e2104878118 (2021) .

[10] Judge, A. *et al.* Network of epistatic interactions in an enzyme active site revealed by large-scale deep mutational scanning. *Proceedings of the National Academy of Sciences* **121** (12), e2313513121 (2024). https://doi.org/10.1073/pnas.2313513121 .

[11] Kim, H. Y. & Kim, D. Prediction of mutation effects using a deep temporal convolutional network. *Bioinformatics* **36** (7), 2047–2052 (2020) .

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

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

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

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

[16] Sim, N.-L. *et al.* Sift web server: predicting effects of amino acid substitutions on proteins. *Nucleic Acids Research* **40** (W1), W452–W457 (2012) .

[17] Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nature Biotechnology* **35** (2), 128–135 (2017) .

[18] Frazer, J. *et al.* Disease variant prediction with deep generative models of evolutionary data. *Nature* **599** (7883), 91–95 (2021) .

[19] 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) .

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

[21] 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** (15), e2016239118 (2021) .

[22] Hie, B., Zhong, E. D., Berger, B. & Bryson, B. Learning the language of viral evolution and escape. *Science* **371** (6526), 284–288 (2021) .

[23] Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379** (6637), 1123–1130 (2023) .

[24] 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) .

[25] Suzek, B. E. *et al.* Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31** (6), 926–932 (2015) .

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

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

[28] Lu, H. *et al.* Machine learning-aided engineering of hydrolases for pet depolymerization. *Nature* **604** (7907), 662–667 (2022) .

[29] Hughes, D. & Andersson, D. I. Evolutionary trajectories to antibiotic resistance. *Annual review of microbiology* **71** (1), 579–596 (2017) .

[30] Wang, X., Zhang, H. & Chen, X. Drug resistance and combating drug resistance in cancer. *Cancer drug resistance* **2** (2), 141 (2019) .

[31] Bryant, D. H. *et al.* Deep diversification of an aav capsid protein by machine learning. *Nature Biotechnology* **39** (6), 691–696 (2021) .

[32] Dallago, C. *et al.* FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv* 2021–11 (2021) .

[33] Jacquier, H. *et al.* Capturing the mutational landscape of the beta-lactamase tem-1. *Proceedings of the National Academy of Sciences* **110** (32), 13067–13072 (2013) .

[34] Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533** (7603), 397–401 (2016) .

[35] 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) .

[36] Faure, A. J. *et al.* Mapping the energetic and allosteric landscapes of protein binding domains. *Nature* **604** (7904), 175–183 (2022) .

[37] 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** (42), 16858–16863 (2012) .

[38] 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** (11), 1537–1551 (2013) .

[39] Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current Biology* **24** (22), 2643–2651 (2014) .

[40] Starr, T. N. & Thornton, J. W. Epistasis in protein evolution. *Protein Science* **25** (7), 1204–1218 (2016) .

[41] Sailer, Z. R. & Harms, M. J. High-order epistasis shapes evolutionary trajectories. *PLoS Computational Biology* **13** (5), e1005541 (2017) .

[42] Naas, T. *et al.* Beta-lactamase database (bldb)–structure and function. *Journal of Enzyme Inhibition and Medicinal Chemistry* **32** (1), 917–919 (2017) .

[43] Stiffler, M. A., Hekstra, D. R. & Ranganathan, R. Evolvability as a function of purifying selection in tem-1 β-lactamase. *Cell* **160** (5), 882–892 (2015) .

[44] 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** (5770), 111–114 (2006) .

[45] Maynard Smith, J. Natural selection and the concept of a protein space. *Nature* **225** (5232), 563–564 (1970) .

[46] Orr, H. A. The distribution of fitness effects among beneficial mutations. *Genetics* **163** (4), 1519–1526 (2003) .

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

[48] Wu, L. *et al.* Sproberta: protein embedding learning with local fragment modeling. *Briefings in Bioinformatics* **23** (6) (2022) .

[49] Vaswani, A. *et al.* Attention is all you need. *Advances in Neural Information Processing Systems* **30** (2017) .

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

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

[52] Kulmanov, M. & Hoehndorf, R. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics* **36** (2), 422–429 (2020) .

[53] 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) .

[54] Wang, Y. *et al.* Self-play reinforcement learning guides protein engineering. *Nat. Mac. Intell.* **5** (8), 845–860 (2023). URL https://doi.org/10.1038/s42256-023-00691-9. https://doi.org/10.1038/S42256-023-00691-9 .

[55] Angermueller, C. *et al.* *Model-based reinforcement learning for biological sequence design* (2020). URL https://openreview.net/forum?id=HklxbgBKvr.

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

(2017) .

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

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

[59] Firnberg, E., Labonte, J. W., Gray, J. J. & Ostermeier, M. A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular Biology and Evolution* **31** (6), 1581–1592 (2014) .

[60] Gonzalez, C. E. & Ostermeier, M. Pervasive pairwise intragenic epistasis among sequential mutations in tem-1 β-lactamase. *Journal of Molecular Biology* **431** (10), 1981–1992 (2019) .

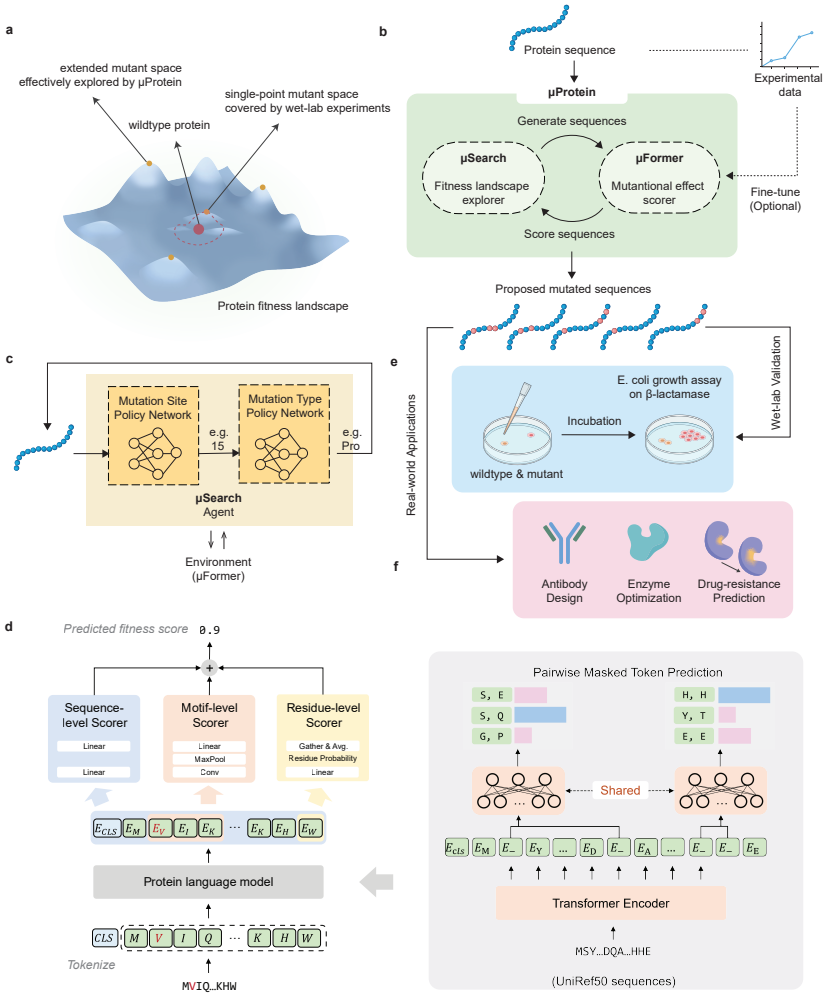


Fig. 1 Overview of μProtein. **a)** μProtein efficiently explores the vast protein fitness landscape, generalizing from limited experimental data, typically confined to single-point mutations. **b)** μProtein workflow. **c)** Overview of μSearch, a reinforcement-learning algorithm that uses a two-step, multi-round search strategy for fitness landscapes exploration. **d)** μFormer's two-step mutational effect prediction approach: (1) pre-training a masked protein language model (LM) on a large database of unlabeled proteins and (2) integrating three scoring modules with a small set of new parameters to predict fitness scores using mutated protein sequences with known fitness. μFormer employs a pairwise masked language model (PMLM) to capture dependencies between masked tokens, focusing on residual, motif, and sequence-level scores for enhanced fitness prediction. **e)** Wet-lab experiment validation of μProtein. **f)** Potential applications of μProtein in real-world scenarios.

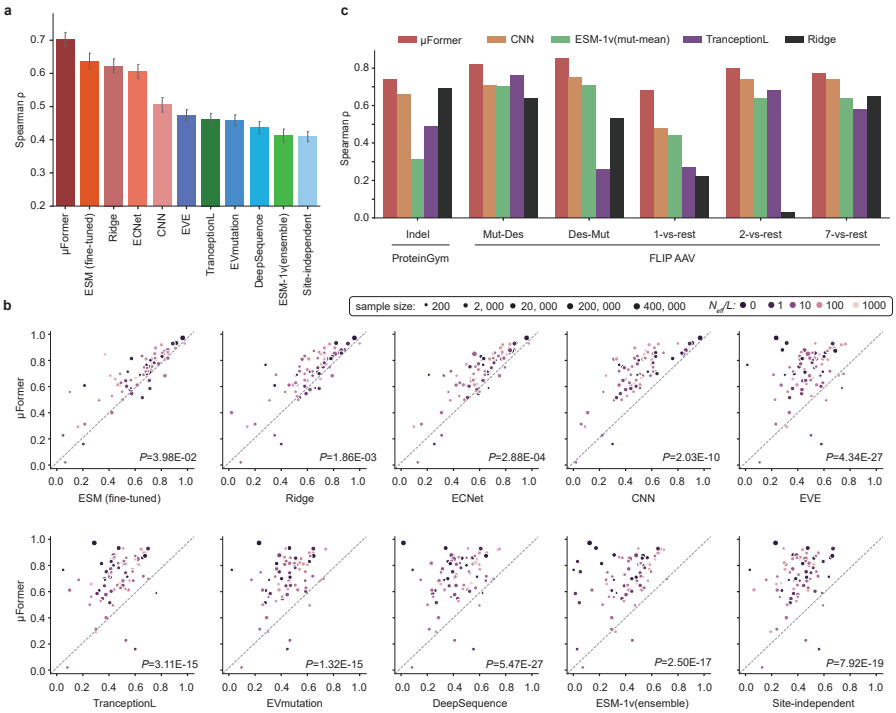


Fig. 2 Quantitative comparison of μFormer with the state-of-the-art mutational effect prediction approaches. a-b) μFormer outperforms alternative approaches on ProteinGym benchmark. a) Spearman ρ statistics on 78 ProteinGym datasets of μFormer and 10 alternative approaches. Red: learning-based approaches, including μFormer, ESM(fine-tuned), 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. Error bar: standard deviation. b) Pairwise comparison between μFormer and other approaches. Each data point represents one 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. [31] and Dallago et al. [32] 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.

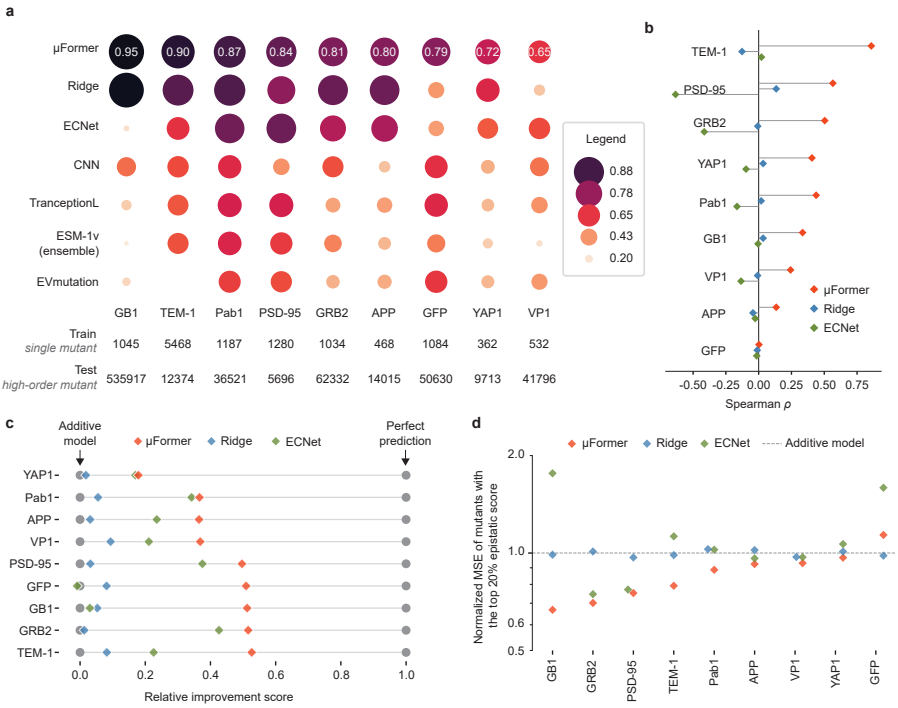


Fig. 3 μFormer models the effects of high-order mutants and epistasis effectively. a) Spearman ρ between predicted and experimentally measured effects for high-order mutants of 9 proteins with diverse functions. The size and darkness of the dots are proportional to the score. Sizes of training data and 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 top 20% absolute epistatic scores in each protein.

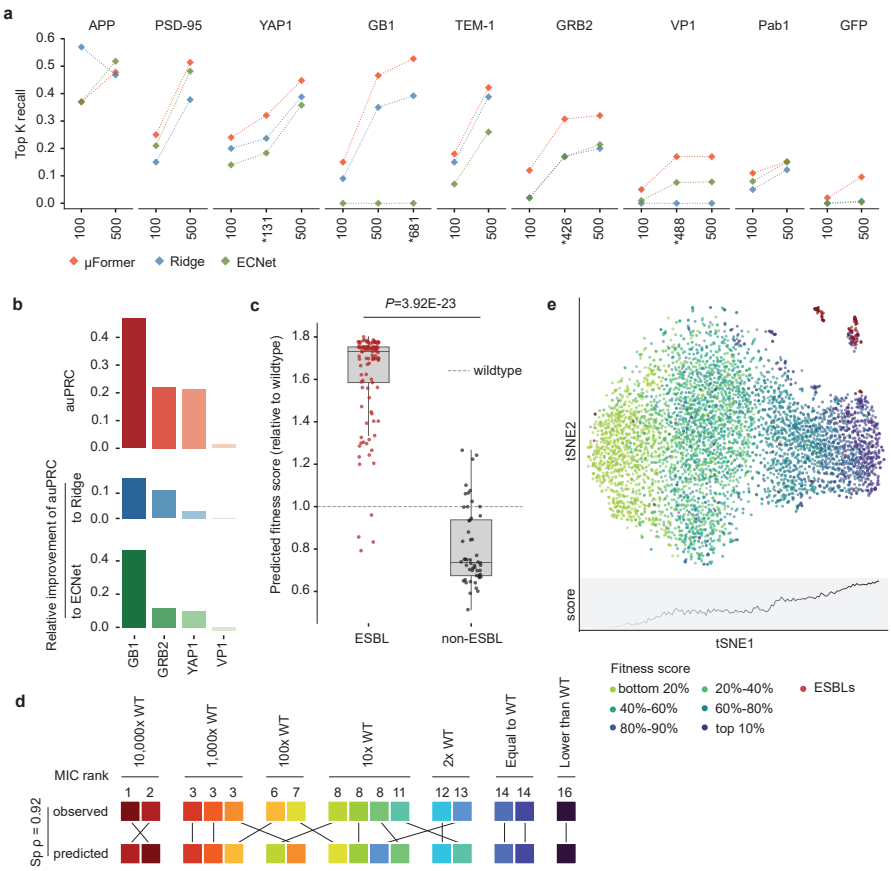


Fig. 4 μFormer effectively identifies high-functioning variants with multi-point mutations. a) Recall score of μFormer (red), Ridge (blue) and ECNet (green) for the Top K high-order mutants of 9 different proteins ranked by fitness values. K is equal to 100, 500, or customized values marked with asterisks. 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 [44]. 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 [43]. The bottom panel shows the average fitness score along PC1.

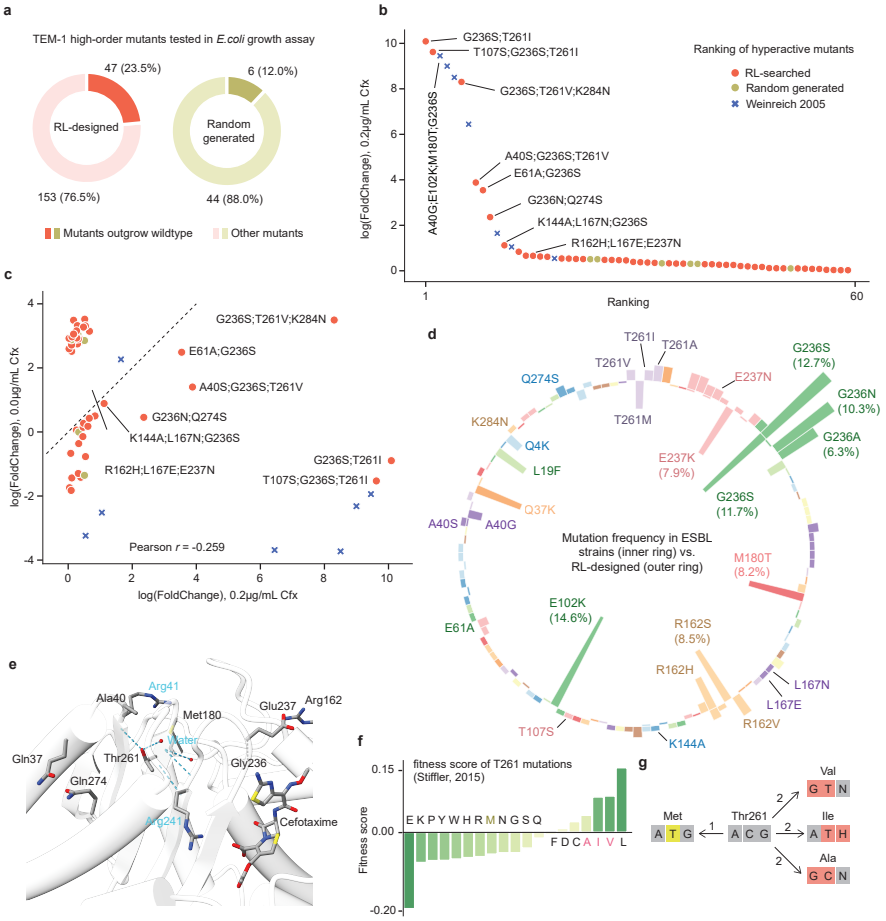


Fig. 5 Design high-functioning sequences with μProtein. a) Percentage of TEM-1 high-order mutants exhibiting superior growth in E. coli compared to wild-type, for both RL-designed and randomly generated sequences. b) Fold change in growth of TEM-1 mutants compared to wild-type E. coli in the presence of 0.2 μg/ml cefotaxime, ranked by performance. Only mutants that outgrew the wild-type are shown. c) Comparison of fold change in growth of TEM-1 mutants relative to wild-type E. coli under two conditions: 0.2 μg/ml cefotaxime and 0 μg/ml cefotaxime. d) The hydrogen bond network involving residue Thr261, with cefotaxime in stick representation. Thr261 forms hydrogen bonds with two water molecules and residues Arg41 and Arg241. e) Distribution of mutations in extended-spectrum beta-lactamases (ESBLs) (inner ring) compared to RL-designed mutants (outer ring). f) Fitness scores of various TEM-1 T261 mutations, as quantified by Stiffler et al [43]. g) Illustration depicting nucleotide mutations at position T261 in natural mutants (ESBL strains, left) and RL-designed mutants (right). Yellow indicates nucleotide mutations observed in natural mutants, while red denotes nucleotide mutations found in RL-designed mutants. The numbers 1/2 represent the number of nucleotide mutations required for the amino acid substitution. H denotes A/C/T, and N denotes A/T/C/G.

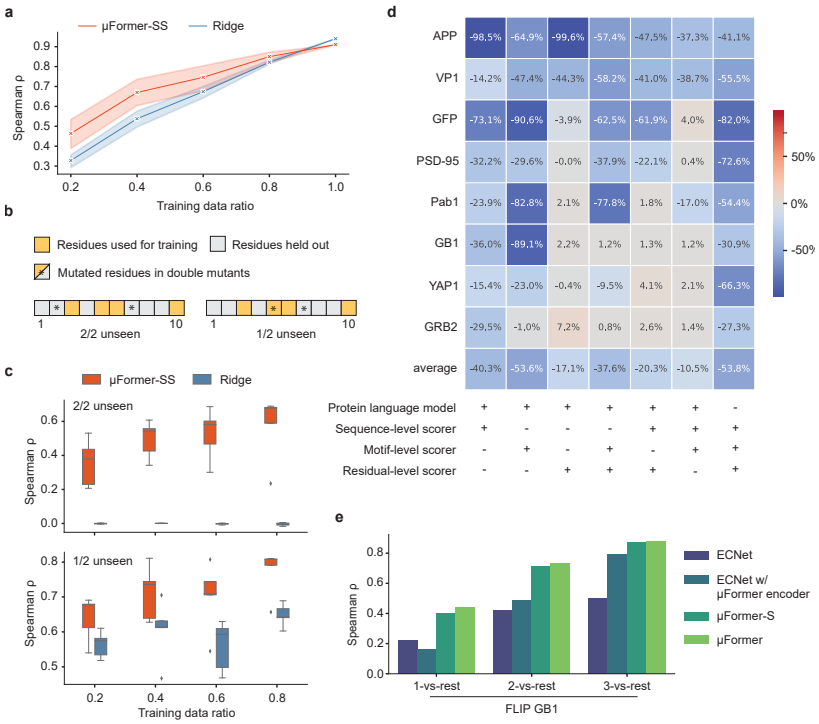
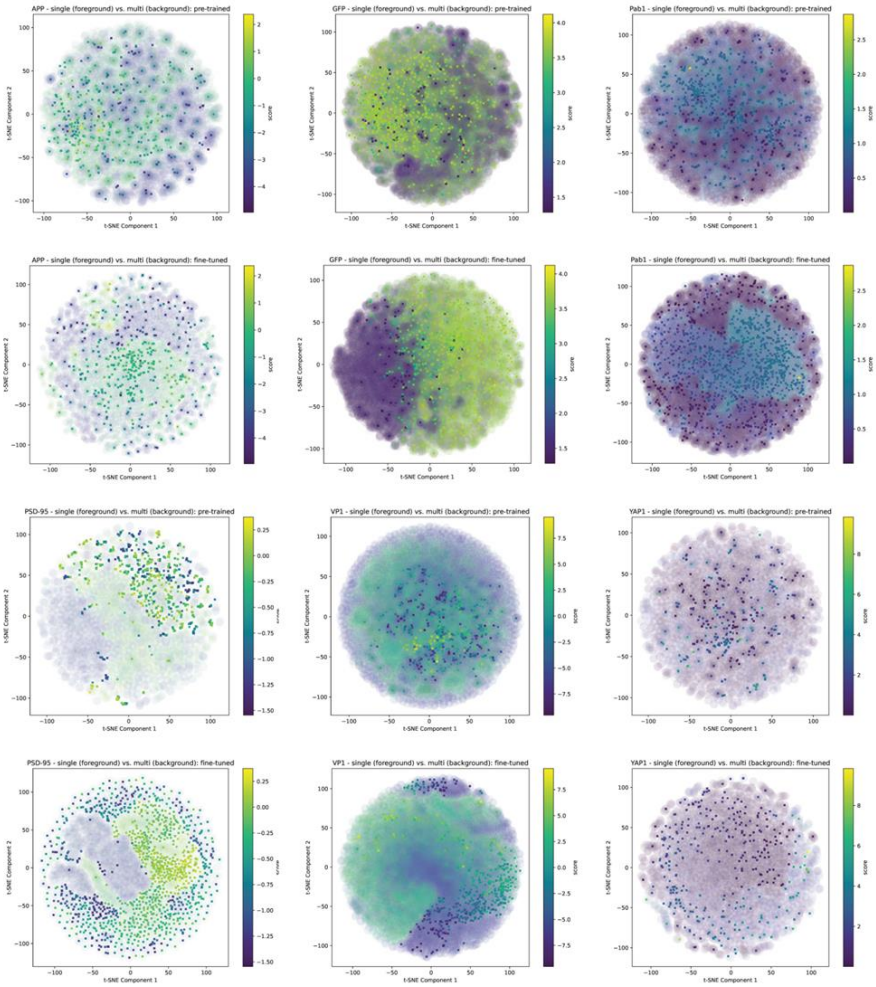
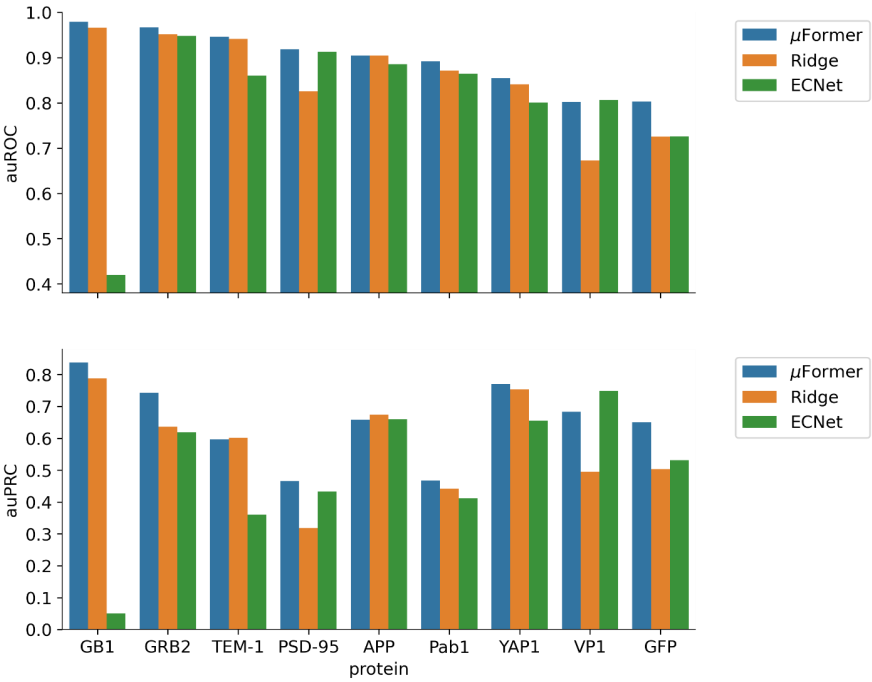


Fig. 6 Analysis of μProtein. a) Ablation study to evaluate the importance of each component in μFormer. The change in performance after removing various components from the model relative to a full model is shown. Negative numbers (blue) indicate a loss of performance and positive numbers (red) indicate 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.

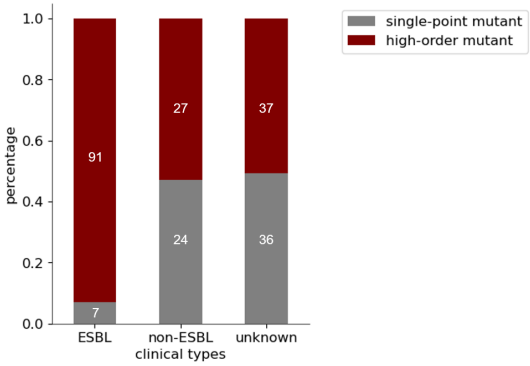


Extended Data Figure 1. Visualization of embeddings on single mutants and high-order mutants (muti mutants) using t-SNE. First row: Illustration of embeddings on APP, GFP, and Pab1 extracted from pre-trained μFormer model. Second row: Illustration of embeddings on APP, GFP, and Pab1 extracted from fine-tuned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

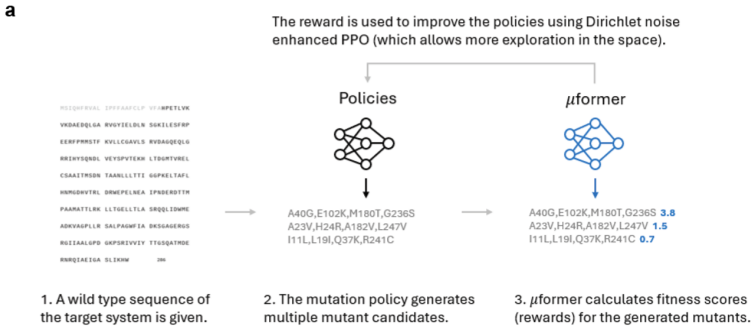
embeddings on PSD-95, VP1, and YAP1 extracted from pre-trained μFormer model. Second row: Illustration of embeddings on PSD-95, VP1, and YAP1 extracted from fine-tuned μFormer model. The color scheme illustrates the fitness scores, with lighter shades indicating higher scores. Blurred background markers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



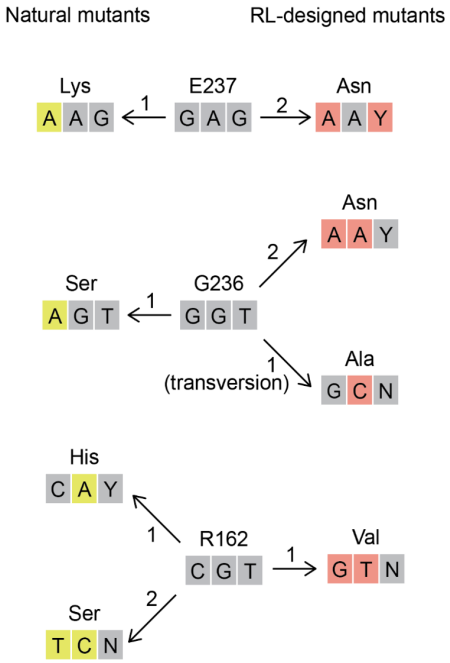
Extended Data Figure 2. μFormer accurately distinguish gain-of-function mutations from loss-of-function mutations. Upper: the area under the receiver operating characteristic (auROC) score for different models on various proteins. Bottom: the area under precision-recall curve (auPRC) score for different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



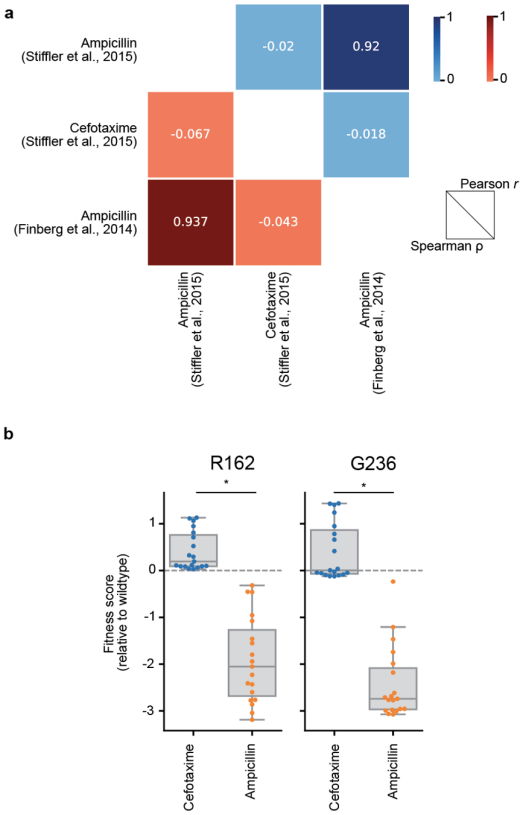
Extended Data Figure 3. Percentage of single-point mutants and high-order mutants in each clinical type of TEM-1. The count number for each group is marked on the plot.



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.



Extended Data Figure 5. Nucleotide mutations in natural mutants (ESBL strains, left) and RL-designed mutants (right). Yellow indicates nucleotide mutations observed in natural mutants, while red denotes nucleotide mutations found in RL-designed mutants. The numbers 1/2 represent the number of nucleotide mutations required for the amino acid substitution. Y denotes C/T, and N denotes A/T/C/G.



Extended Data Figure 6. 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.