Accelerating protein engineering with fitness landscape modeling and reinforcement learning

Haoran Sun[1†], Liang He[1*†], Pan Deng[1*†], Guoqing Liu[1†], Zhiyu Zhao[2], Yuliang Jiang[3], Chuan Cao[1], Fusong Ju[1], Lijun Wu[1], Haiguang Liu[1*], Tao Qin[1*] and Tie-Yan Liu[1]

[1]Microsoft Research AI for Science.
[2]Institute of Automation, Chinese Academy of Sciences, China.
[3]Tsinghua University, China.

*Corresponding author(s). E-mail(s): liang.he@microsoft.com; pan.deng@microsoft.com; haiguang.liu@microsoft.com; taoqin@microsoft.com;
Contributing authors: 767780791shr@gmail.com; guoqingliu@microsoft.com; zyzhao2021@gmail.com; jiangyl22@mails.tsinghua.edu.cn; chuancao@microsoft.com; fusongju@microsoft.com; lijuwu@microsoft.com; tie-yan.liu@microsoft.com;
†These authors contributed equally to this work.

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 leverages single mutation data to predict optimal sequences with complex, multi-amino acid mutations through its modeling of epistatic interactions and a multi-step search strategy. Except from state-of-the-art

performance on benchmark datasets, **μ**Protein identified high-gain-of-function multi-point mutants for the enzyme **β**-lactamase, surpassing the highest known activity level, in wet-lab, trained solely on single mutation data. These results demonstrate **μ**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 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, a process that forms the foundation for identifying sequences with desired traits. This mapping, known as the fitness landscape, is often complex and rugged due to the intricate interactions, or epistasis, between amino acid residues within protein sequences.

High-throughput experimental techniques like Deep Mutational Scanning (DMS)[6] and Multiplex Assays of Variant Effects (MAVEs)[7] have significantly advanced our ability to measure fitness effects for protein variants, particularly by systematically probing single substitutions at every position in a protein. However, these powerful tools face two major limitations. First, they are unable to explore the full combinatorial sequence space, which grows exponentially with the number of mutations. This makes it impractical to experimentally measure high-order mutation combinations, leaving vast regions of the fitness landscape unexplored [4, 8–10]. Second, these methods rely on experimental readout systems that link protein activity to a measurable outcome, such as cell growth, fluorescence, or binding affinity. While effective for certain fitness traits, many protein functions cannot be easily measured using such high-throughput approaches. These constraints further limit the versatility and generality of DMS and MAVEs in addressing diverse protein engineering objectives, underscoring the need for alternative approaches.

Computational models that predict fitness from limited experimental data offer a promising solution, enabling exploration of sequence-function relationships beyond the reach of current experimental techniques [2–5, 10–24]. In this study, we introduce μProtein, a 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. Within this framework, μFormer leverages a pairwise masked language model and

three specialized scoring modules to model fitness landscapes, which hence enable μFormer to handle complex scenarios, such as multiple mutations, indel mutations, and orphan proteins. μSearch complements this by framing protein engineering as a decision-making process, using Proximal Policy Optimization (PPO) with exploration noise to navigate the rugged fitness landscape. Together, these components offer a comprehensive and efficient strategy for identifying high-fitness mutations within complex fitness landscapes, thereby expanding the scope of protein engineering.

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 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 tested samples, with *E. coli* harboring certain variants exhibiting growth rates up to 2,000-fold higher than the wild type.

2 Results

2.1 Overview of μProtein

μProtein is designed to generalize from limited experimental observations and efficiently explore the vast fitness landscapes of proteins (Fig. 1a). The framework combines μSearch and μFormer. Starting with a wild-type sequence, μSearch, the reinforcement learning framework, autonomously navigates the multi-amino acid mutational landscape, guided by μFormer, a pre-trained oracle model that scores the mutational effects of sequences proposed by μSearch (Fig. 1b).

μSearch frames protein engineering as a Markov Decision Process (MDP), where the state represents the current mutant sequence, the action corresponds to a single amino acid mutation. In each episode of the MDP, μSearch sequentially mutates multiple amino acids from the wild-type sequence being engineering. The learning objective of μSearch is to optimize the selection policy for mutation sites and types at each step to achieve the desired properties. Using Proximal Policy Optimization (PPO) with Dirichlet noise to avoid local optima, μSearch navigates the rugged fitness landscape with both a mutation site policy network and a mutation type policy network, efficiently identifying high-fitness mutations (Fig. 1c. Methods).

μFormer, acting as the oracle, provides reward scores for μSearch. μFormer is built from a self-supervised protein language model combined with three supervised scoring modules (sequence-level, motif-level, and residue-level. Methods). 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 by leveraging both local residue context and pairwise interaction likelihoods within the sequence. During fine-tuning, the full-length sequence embeddings, motif embeddings, and residue probabilities are passed to their

respective scoring modules (Fig. 1d, left). The outputs of these modules are then combined to generate the final fitness prediction for the query sequence.

The sequence- and motif-level scoring modules have been independently validated as effective tools for modeling protein fitness [22, 27]. Similarly, the residue-level scorer, which evaluates specific amino acid changes at mutated positions, has shown promise in zero-shot fitness prediction tasks [24]. Building on these foundations, we introduce a scoring module that generalizes the residue-level approach. Unlike the traditional residue-level scorer, which focuses exclusively on mutated residues, our design incorporates unmasked sequences during fine-tuning and aggregates probabilities across all residues in the query sequence. The integration of these modules, each focusing on distinct protein features and further enhanced by the pre-trained protein language model, provides a more accurate and comprehensive representation of protein fitness landscapes (Extended Data Fig. 1 and Supplementary Notes), ultimately offering a reliable reward function for μSearch.

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 [28], enzyme optimization [29], and the prediction of drug-resistant mutations [30, 31] (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 16 alternative approaches, including MSA-based methods [2, 16–18], language model-based zero-shot methods [5, 24] and learning-based approaches [13–15, 32, 33]. We first evaluated all models on ProteinGym [5], a collection of 78 DMS-derived protein mutational effect datasets spanning a diverse set of protein types, sequence lengths, biological functions, and fitness assays.

Among the tested methods, μFormer demonstrates the best overall performance. When analyzing performance on individual datasets, μFormer consistently outperforms alternative approaches across the majority of cases (Fig. 2a and Supplementary Data Fig. 1). Moreover, μFormer achieves superior results more frequently in terms of Spearman's rank correlation coefficient (ρ) between predicted and observed fitness scores across the tested datasets (Fig. 2b). Notably, for over 50% of the datasets tested, μFormer achieves a Spearman's ρ greater than 0.7, and for 6 datasets, it exceeds 0.9, outperforming all other methods. This highlights μFormer's robust capability to deliver high-quality predictions across diverse fitness landscapes. As some methods failed on specific datasets due to their suboptimal algorithm design, we calculated both the absolute number and the proportion of high-performing datasets for each model to ensure a fair and comprehensive evaluation.

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. 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. Here, we benchmarked μFormer on indel tasks against available approaches. Our method showed the best performance on all 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 [27, 34]. 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 [27]. 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.

We also showed that μFormer could generalize to unseen residues, highlighting its data efficiency and generalization ability (Extended Data Fig. 2 and Supplementary Notes)

2.3 μFormer captures epistatic mutational effect of high-order variants

We next assessed whether μFormer could generalize to multi-point mutants, i.e. high-order variants, while being trained exclusively on single-point mutants (single-to-multi setting). We collected nine DMS datasets that include both single-point mutants and high-order mutants [27, 34–41], and trained μFormer using only single mutation data for each target protein. To be noted, the training datasets (single mutants) range in size from 362 to 5,468, whereas the test datasets (high-order mutants) are substantially larger (Extended Data Table 1).

A key object in this single-to-multi setting is to assess the model's ability to predict epistasis, wherein the combined mutational effects at various sites are not simply the sum of their individual effects [8, 42, 43]. Rather than capturing additive effects (linear relationships between single mutations and higher-order mutations), we sought to determine whether models can capture interdependencies between amino acids that are not directly observed in the single-mutant training data. This capability is critical for understanding and modeling complex protein fitness landscapes.

To this end, we calculated epistatic scores for high-order mutants as the discrepancy between the observed mutational effect values and the sum of the constituent single-mutant effect values: $S_{epi} = Effect_{pos_1,pos_2...pos_n} - \Sigma_{i=1}^{n} Effect_{pos_i}$ (Fig. 3a). Here, the observed S_{epi}

is defined as the observed multi-mutant effect minus the sum of individual observed single-mutant effects. Similarly, the predicted S_{epi} is defined as the calculation between the predicted multi-mutant effect and the predicted single-mutant effects generated by each model. Compared to Augmented DeepSequence and ProteinNPT - models that demonstrated strong performance in the previous section - μFormer more accurately captures the direction of epistasis, and the epistatic scores predicted by μFormer show a stronger correlation with observed S_{epi} across various proteins (Fig. 3b). Additionally, for high-order mutants exhibiting the top 20% absolute epistatic scores in each protein (Fig. 3a), μFormer reduces predictive errors relative to observed values in 8 out of 9 tested proteins compared to the additive model (Fig. 3c). μFormer also achieves the highest Spearman's ρ overall (Fig. 3d). These results collectively demonstrate that μFormer effectively captures epistatic effects, rather than merely estimating the additive contributions of single mutants present in the training data.

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) (Supplementary Data Fig. 2). Interestingly, 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. Thus, the model's ability to capture the relationship between fitness scores and the corresponding variants may have enabled it to effectively predict the outcomes for high-order mutants, despite not being explicitly trained on them.

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 (Supplementary Data Fig. 3), we further investigated if the model could predict top-performing mutants effectively.

We calculated the Top-K recall scores on curated high-order mutant datasets using the single-to-multi setting for μFormer, DeepSequence, and ProteinNPT. This evaluation measures the percentage of top K high-functioning high-order mutants correctly identified among the 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 by detecting ultra-high valued mutants, except the VP1 case (Fig.4a-b). These results together show that μFormer can provide efficient guidance for protein optimization.

To evaluate μFormer in practice, we focused on the acquired hydrolysis activity of TEM-1. TEM-1 is a β-lactamase that degrades β-lactam antibiotics such as ampicillin. Cefotaxime, a β-lactam antibiotic discovered nearly 50 years ago, was initially resistant to degradation by TEM-1, making it effective against bacterial pathogens carrying this enzyme. However, TEM-1 variants capable of hydrolyzing cefotaxime rapidly evolved following the drug's introduction to the market. Among the 222 sequenced clinical variants of TEM-1, 98 are classified as extended-spectrum β-lactamases (ESBLs) [44], indicating activity against extended-spectrum cephalosporins with an oxyimino side chain, including cefotaxime. Most ESBLs (91 out of 98) are high-order mutants of TEM-1 (Supplementary Data Fig. 4), we therefore examined whether μFormer could prioritize high-order ESBL mutants.

We fine-tuned μFormer using a TEM-1 assay performed by Stiffler et al. [45], which measured the fitness effects of approximately 5,000 single-point TEM-1 mutants under cefotaxime selection. Then, we evaluated μFormer using a curated dataset comprising 105 ESBLs and 52 confirmed non-ESBLs [44, 46]. 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) [46]. In contrast, the additive model assigned smaller fitness score gaps between ESBLs and non-ESBLs (Supplementary Data Fig. 5).

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, when ranked by fitness values, the top 1% (50) TEM-1 mutants from the single mutant DMS study [45] 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 μSearch efficiently navigate fitness landscapes of proteins

μSearch is designed to efficiently explore protein fitness landscapes and thus enabling effective protein optimization. To assess the capability of μSearch, we first evaluated it using FLEXS [47], an open-source toolbox for benchmarking biological sequence design algorithms. This toolbox provides an approximate fitness model independent of μFormer, serving as a noisy oracle to guide diverse

exploration algorithms. Additionally, it offers five "ground truth" fitness landscapes spanning DNA, RNA, and proteins [36, 48–51] (Supplementary Notes), enabling comprehensive assessment of exploration algorithms.

We benchmarked μSearch against eight alternative exploration algorithms [47, 52–57] across these diverse fitness landscapes. Following the principles of machine-guided directed evolution [58, 59], we employed a multi-round simulated experimental design. In this setup, each algorithm iteratively queried the approximate fitness model, proposed sequences, and updated the model with ground truth oracle-evaluated samples (Supplementary Notes). Specifically, we conducted ten rounds of in silico search using three different random seeds. In each round, each algorithm proposed 100 candidate sequences, constrained by a computational budget of 5,000 local approximate model calls.

Compared to alternative approaches and a Random Selection baseline, μSearch demonstrated superior sample efficiency on several protein landscapes (Fig. 5a–c), highlighting its effectiveness in navigating complex protein fitness landscapes. However, we observed that this advantage was less pronounced on nucleic acid-based tasks (Supplementary Data Fig. 6). A possible explanation is that the smaller search spaces in these tasks reduce the difficulty of the optimization problem, resulting in comparable performance across different methods (Supplementary Notes).

We further evaluated different exploration algorithms using μFormer as the oracle model in the TEM-1-cefotaxime system, employing a single-round experimental setting. Here, each algorithm was allowed to make 250,000 oracle queries, without any feedback from the ground truth oracle. μSearch identified sequences with higher oracle-predicted fitness scores as the number of model queries increased (Fig. 5d). Remarkably, μSearch achieved fitness scores that no other algorithm was able to reach even after 250,000 queries, with around 50,000 queries (Fig. 5d). By the end of 250,000 queries, μSearch identified significantly more sequences with high predicted scores compared to other algorithms (Figs. 5e-f). Notably, for sequences with predicted scores exceeding 0.3, μSearch identified over 2,000 sequences that are challenging for all other algorithms to detect (Fig. 5f). These results demonstrate that μSearch effectively explores complex fitness landscapes and proposes sequences with potentially optimized properties of interest, particularly when paired with μFormer.

Lastly, to assess computational efficiency, we compared μSearch with random sampling by measuring the number of model queries required to identify mutants exceeding varying fitness thresholds (Supplementary Notes). The results show that μSearch achieves the same or higher fitness levels with significantly fewer model evaluations, particularly at higher thresholds, highlighting its superior efficiency in exploration.

2.6 Design high-functioning sequences using μProtein

After comprehensive benchmarking on μFormer and μSearch, we employed the TEM-1-cefotaxime system to assess μProtein's effectiveness in guiding protein optimization. Using the μFormer model fine-tuned on TEM-1 single mutant

data (see the previous section) as the reward function, μSearch explored millions of possible TEM-1 variants containing 2-3 amino acid changes, aiming to identify those with enhanced activity against cefotaxime (Supplementary Data Fig. 7, Methods).

From the sequences explored through the RL search, we identified over 1,200 variants that exceeded a threshold defined by the median predicted value of known ESBL mutants. From this set, we selected the top 200 variants based on ensemble ranking scores generated by μFormer and validated these variants using *E. coli* growth assays (Methods). For comparative analysis, we also included 50 randomly generated variants with 2-3 amino acid changes, as well as 15 variants with 1 to 4 amino acid substitutions spanning A40G, E102K, M180T, and G236S, which have been previously tested in minimum inhibitory concentration (MIC) assays [46], .

E. coli carrying 47 distinct 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. 6a), indicating enhanced enzymatic activity against cefotaxime. In contrast, 12% of randomly generated TEM-1 variants showed enhanced activity. This is consistent with prior observations that TEM-1 mutations often increase activity against cefotaxime [45]. Furthermore, RL-designed variants demonstrated greater activity gains compared to random-generated variants (Fig. 6b). Notably, a double-mutation combination, G236S;T261V, surpassed the activity level of the well-known quadruple mutant A40G;E102K;M180T;G236S, which exhibits significant activity against cefotaxime relative to wild-type TEM-1 [46]. While G236S has been linked to moderate cefotaxime activity gain in previous studies [46], T261I is a mutation that was not previously reported in the ESBL strains. Further validation experiments demonstrate that these novel high-order mutants provide bacteria with a significant growth advantage compared to their constituent single mutations (Supplementary Data Fig. 8).

Comparison of relative growth rates revealed a weak correlation between growth in the presence and absence of cefotaxime for hyperactive mutants (Fig. 6c). Mutants with ultra-high activity tended to exhibit reduced growth in the absence of cefotaxime (bottom-right corner of the plot), consistent with the understanding that beneficial mutations under selective pressure may be detrimental in neutral conditions [60–62]. However, some strains, such as G236S;T261V;K284N, showed growth advantages under both treated and untreated conditions (Fig. 6c).

Notably, the mutation distribution in RL-designed strains differed significantly from that of natural ESBL strains (Fig. 6d). 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. 6e).

Two 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 (Supplementary Data Fig. 9), suggesting that our method favors single 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. 6f and Supplementary Data Fig. 10). 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.

3 Discussion

In this study, we present μProtein, an integrated framework that combines the deep learning model μFormer with the reinforcement learning algorithm μSearch to model protein fitness landscapes and efficiently explore mutational space. Extensive benchmarking demonstrated μFormer's superior accuracy in predicting fitness landscapes and μSearch's ability to identify optimal variants. We further validated μProtein by optimizing the TEM-1 *beta*-lactamase enzyme. By fine-tuning μFormer on TEM-1 single mutant data, we identified several TEM-1 variants highly active against cefotaxime. Notably, μProtein discovered novel high-order mutants, such as G236S;T261V, which exhibited activity levels exceeding those of known high-activity variants. These findings highlight μProtein's ability to model and search for high-order mutations, addressing the critical challenge of capturing epistatic interactions in protein engineering.

Training data dependency presents a fundamental trade-off in protein engineering, with zero-shot and supervised methods offering distinct advantages. Zero-shot methods are appealing for their ability to generalize without requiring task-specific data, but they often fail to capture desired sequence-function relationships. For proteins like TEM-1 beta-lactamase, where mutational effects can vary significantly across related phenotypes (e.g., resistance to

different beta-lactam antibiotics [45], Supplementary Data Fig. 11), supervised training is particularly valuable. Fine-tuning on phenotype-specific data enables the model to achieve higher accuracy in predicting mutational impacts and high-order interactions.

In comparison with alternative methods on extensive benchmarks, μFormer demonstrated leading overall performance but did not excel across all tasks. This highlights the fact that no single model currently performs optimally for every protein engineering challenge. Addressing task-specific limitations and enhancing generalizability will be critical for future improvements in computational protein design.

Several potential directions may further enhance μProtein's capabilities. Integrating 3D structural data may improve its ability to capture spatially proximate epistatic interactions that are not evident from sequence data alone. This would be particularly beneficial for applications like antibody engineering or enzyme optimization, where structural considerations play a key role [63]. Additionally, expanding the training framework to incorporate diverse phenotypes and environmental contexts could enable multitarget fitness predictions within a unified model. For instance, encoding factors such as substrate specificity, temperature, or pH conditions could broaden μProtein's applicability to more complex protein optimization scenarios.

In this study, we generated approximately 250 TEM-1 mutant strains to assess μProtein's predictions. While this dataset offered valuable insights, its relatively small size, especially insufficient number of negative samples required for a balanced assessment, limited the fully evaluation on model's performance. In addition, multi-round experimental feedback, such as iterative cycles of machine learning-guided directed evolution or active learning, could further refine the model and improve its predictive accuracy in future studies.

Overall, μProtein offers a versatile framework for protein engineering by integrating a powerful fitness prediction model with an efficient reinforcement learning strategy. Our study demonstrates its ability to bridge the gap between sequence-based predictions and functional protein optimization. By achieving state-of-the-art performance across a variety of tasks, μProtein provides a valuable tool for designing tailored protein functionalities, opening new possibilities for 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, 64] is a self-supervised learning technique that utilizes the Transformer encoder [65] 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 prHsu et al. otein 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 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:

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

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. The final loss function for pre-training is defined as the sum of the masked language modeling loss and the pairwise masked language modeling loss, computed by comparing the ground truths with the predictions from two separate heads:

$$\mathcal{L} = \mathcal{L}_{mlm} + \mathcal{L}_{pmlm}. \tag{3}$$

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:

$$x = \textsc{LayerNorm}(x + \mathrm{MHA}(x)), \tag{4}$$

$$x = \textsc{LayerNorm}(x + \mathrm{FFN}(x)). \tag{5}$$

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 language model using the UR50 dataset, which contains approximately 30 million protein sequences. We adopt the same masking strategy as BERT [19]: for each input sequence, we randomly select 15% of the residues. Of these, 80% are replaced with a special mask token, 10% are substituted with a random amino acid, and the remaining 10% are left unchanged – though the model is still tasked with predicting them and their contributions are included in the loss. This probabilistic behavior enables the model to learn how to refine sequences and identify unfavorable mutations.

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 residue-level score $S_{\text{RESI}}(x)$, which characterizes the likelihood of residues at each position conditioned on the entire sequence [22, 24, 66]. It consists of a multi-layer perceptron (MLP) that processes the hidden representation of each residue to predict amino acid likelihoods. The final score is computed as the average likelihood over all residues.
- 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 [10, 64, 67]. It consists of a residual network with two convolutional blocks, followed by a pooling layer and an MLP to generate motif-level scores.

- 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, 66, 68, 69]. Like the residue-level scorer, it is an MLP-based model, but it generates predictions using the entire sequence representation as input.

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

$$S(x) = S_{\text{RESI}}(x) + S_{\text{MOTIF}}(x) + S_{\text{SEQ}}(x). \tag{6}$$

Residue-level Scoring Module

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

$$S_{\text{RESI}}(x;\ \theta) = \frac{1}{L}\sum_{i=1}^{L} \log P_\theta(x_i \mid x), \tag{7}$$

where L is the length of the sequence x, and $P_\theta(x_i \mid x)$ represents the estimated likelihood/probability of the i-th residue in the mutant-type sequence x, given by the protein LM parameterized by θ. Although this score originates from residue-level probabilities, the logits contribute to the final fitness score as a residue-level feature during optimization, where it no longer strictly follows probability constraints. Since the parameters in this component are pre-trained, they also benefit from both the pre-training and fine-tuning processes.

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 $\sum_{i\in M} \log \frac{P_\theta(x_i^{mt}|x_{/M})}{P_\theta(x_i^{wt}|x_{/M})}$ for mutational effect prediction, where M is the set of mutated positions, and $P_\theta(x_i^{mt} \mid x_{/M})$ and $P_\theta(x_i^{wt} \mid x_{/M})$ are the estimated probabilities for the i-th residues of the mutant-type and wild-type sequences conditioned on the unmasked residues, 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 not rely on the same-length assumption; instead, it sums over all residues (L) rather than only the mutated ones (M), making it suitable for mutations involving insertions or deletions. Moreover, the design leveraging unmasked predictions instead of masked ones allows other residues to account for mutation effects by aggregating information across the entire sequence too.

Motif-level Scoring Module

Motifs, which are consequential sequence patterns, play a crucial role in protein sequence modeling, widely used in bioinformatics research [64, 67]. 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

$$S_{\text{MOTIF}}(x;\ \alpha) = \textsc{Linear}(\textsc{MaxPool}(\textsc{Conv}(x) + x)), \tag{8}$$

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 are 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, 68] 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

$$S_{\text{SEQ}}(x;\ \beta) = \text{MLP}(x_{cls}), \tag{9}$$

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):

$$\mathcal{L} = \sum_{(x,\ y)\in\mathcal{D}} \|S(x;\ \theta, \alpha, \beta) - y\|, \tag{10}$$

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, 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 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 [27]. 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 called ProteinGym v0.1 and was downloaded from https://github.com/OATML-Markslab/ProteinGym on 12/30/2022, and this data version can be accessed via https://huggingface.co/datasets/OATML-Markslab/ProteinGym_v0.1 now. 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. Our experiments were conducted on ProteinGym v0.1 using a different data split scheme, where the data was randomly divided into training, validation, and test sets with a 7:1:2 ratio. We have made the dataset publicly available at https://doi.org/10.6084/m9.figshare.26892355. The performances of all trained models, including the zero-shot models, are reported on the fixed hold-out test set.

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 [27]. 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 supervised models, we use publicly available implementations and retrain them on the same data splits for each dataset. Specifically, for onehot embedding-based Ridge and CNN models, we adopt the architecture and hyperparameters from the FLIP codebase [27]. For ECNet [14], we reproduce the model using the open-source code provided by Luo et al. [14]. Since ECNet utilizes both MSAs and labeled data, we retrain it under the same settings using the prepared MSAs. Additionally, we fine-tune ESM-1b following the approach in [15]. We also include augmented Ridge models built on EVmutation and DeepSequence [15], evaluating both sampled and full-training data versions. For further comparison, we incorporate recent models, including ProteinNPT [32] and ConFit [33], along with its augmented variant leveraging DeepSequence.

4.2 Fitness Landscape Navigation by μSearch

To efficiently explore the vast space of mutations and identify high-functioning mutants in the protein landscape, we develop a reinforcement learning (RL) method called μSearch, which uses the μFormer model as a reward function. Similar to previous work [53, 70], we frame the protein engineering problem using a Markov Decision Process (MDP). In our MDP, the state represents the current mutant sequence, the action signifies a one-amino acid mutation, including the selection of mutation sites and types to mutate within the

sequence, and the reward function is given by the μFormer model. With a protein engineering agent, during each episode of the MDP, the agent sequentially samples actions from a conditional distribution known as the policy (i.e., mutates a single-site residue) until it reaches a predetermined horizon. At the end of an episode, the agent produces a multi-point mutant from the wild-type sequence we are engineering. The objective of reinforcement learning is to optimize the selection policy for mutation sites and types for the desired properties through trial-and-error interactions with the reward function. This learning process alternates between two phases: mutation space exploration, and policy and value network updating. During the mutation space exploration phase, the agent navigates the extensive mutation space, leveraging a mutation site policy network and a mutation type policy network to generate potentially high-performing mutants. In the policy and value network updating stage, the agent uses the μFormer model to evaluate the newly generated mutants and employs update loss functions to improve the policy and value networks, aiming to propose mutants with higher fitness scores in subsequent iterations.

4.2.1 PPO Algorithm with Mutation Site and Mutation Type Policy Networks

In μSearch's MDP definition, each action corresponds to a single 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 $20 \times 2L$, where L denotes the length of the protein sequence and 20 means the number of types of amino acid. 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. The two networks work sequentially. First, the mutation site policy network identifies potential sites for mutation in the sequence. Then, the mutation type policy network selects the specific amino acid change to be made at the chosen site.

We employ the Proximal Policy Optimization (PPO) algorithm [71] to optimize both policy networks, a method recently utilized to train large language models with human preference-based reward models [72, 73]. 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:

$$
\begin{aligned}
&L_{policy}(\theta) = \mathbb{E}_t[\min(r_t(\theta) \cdot A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t)], \\
&\text{where } r_t(\theta) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}, A_t = S(x) - v_\phi, \\
&L_{value}(\phi) = (v_\phi - S(x))^2.
\end{aligned} \tag{11}
$$

In this equation, t represents the step (timestep) in an episode, where each step corresponds to a single mutation being performed. The term $r_t(\theta)$ is the probability ratio of the current policy $\pi(\theta)$ to the previous policy $\pi_{old}(\theta)$, and v_ϕ denotes the value function. $S(x)$ is the μFormer score of the generated mutant x. The parameters θ and ϕ refer to the policy and value network parameters, respectively. Both the mutation site and mutation type policy networks use this loss function to update their model parameters.

The value function v_ϕ serves as a baseline or critic, aiding in the computation of the advantage function A_t. This function assesses the relative value of each action by calculating the difference between the actual return, $S(x)$, and the expected return derived from the value function. The advantage function A_t allows the policy network to differentiate superior actions from the rest, adjusting its probability correspondingly. The policy loss $L_{policy}(\theta)$ is designed to increase the likelihood of actions with positive advantage functions while minimizing substantial deviations from the previous policy to prevent performance collapse. The hyperparameter ϵ 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 and ϵ is set to 0.25. This added noise ensures that all potential moves can be explored by the agent, while still prioritizing mutations with superior fitness scores from μFormer. During the updating phase, the injected Dirichlet noise may lead to an action with a low probability from the policy network π_θ, potentially resulting in numerical optimization challenges when estimating the original importance weights, we introduce a 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 with Multiple μFormer Models and Ranking based Selection for Experimental Validation

To identify TEM-1 variants with 2–3 amino acid mutations that exhibit enhanced activity against cefotaxime, μProtein integrated the μFormer model with the μSearch algorithm. Six separate μSearch runs were performed using μFormer models trained with different random seeds. In each run, we set a

threshold determined by the median predicted value of known ESBL mutants (as scored by the specific μFormer model used in that search). Variants exceeding this threshold were identified as high-potential candidates. For the TEM-1 system, a total of one million mutant sequences were computationally screened by the μFormer reward model across these six runs.

The results from the six independent searches were compiled into a comprehensive list of over 1,200 unique mutants. For each mutant in this list, we calculated its predicted scores across all six μFormer models and determined its relative rank within each search. The final ranking of the mutants was obtained by aggregating these relative ranks across the six searches. This automated ranking process was used to select the top 200 variants for experimental validation, balancing predictive confidence and wet-lab cost constraints. For the TEM-1 system, we conducted only one round of wet-lab experimental screening.

This approach ensures that the selection process is fully automated, free from manual curation, and leverages consensus predictions across multiple independent μFormer models for robust candidate prioritization.

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 introduced into each TEM-1 mutant construct.

4.3.2 Antibiotic Selection

Following established protocols from previous studies [45, 74, 75], 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 analysis

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 [76] (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 the 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 at http://data.bioembeddings.com/public/FLIP/. The ProteinGym v0.1 data collection can be accessed at https://huggingface.co/datasets/OATML-Markslab/ProteinGym_v0.1. The dataset under our split scheme is available at https://doi.org/10.6084/m9.figshare.26892355. The list of extended-spectrum beta-lactamases can be found at http://bldb.eu/([44]). The fitness scores for beta-lactamase variants against cefotaxime, validated by our wet-lab experiments, are provided in the supplementary files.

Code Availability.
The code to reproduce the results of the paper is publicly available at the following URL: https://github.com/microsoft/Mu-Protein (https://www.doi.org/10.5281/zenodo.15836168). The data split scheme is available at https://doi.org/10.6084/m9.figshare.26892355.

Acknowledgments.
We gratefully acknowledge Dr. Ostermeier for generously providing detailed information on the plasmids pSkunk3-TEM-1 and pTS42, and for sharing the pTS42 plasmid itself. We also extend our thanks to Tianbo Peng for developing the demonstration webpage and Jingyun Bai for creating the graphic illustrations. Lastly, we appreciate Dr. Bonnie Kruft and Dr. Usman Munir for their invaluable support in program management and coordination.

Authors' contributions.
Conceptualization, L.H., H.S., and P.D.; methodology and modeling, L.H., H.S., G.L., Z.Z., Y.J., 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] Miton, C. M. & Tokuriki, N. Insertions and deletions (indels): a missing piece of the protein engineering jigsaw. *Biochemistry* (2022) .

[2] 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) .

[3] 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) .

[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] 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 .

[10] 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) .

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

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

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

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

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

[32] Notin, P., Weitzman, R., Marks, D. & Gal, Y. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. *Advances in Neural Information Processing Systems* **36**, 33529–33563 (2023) .

[33] Zhao, J., Zhang, C. & Luo, Y. *Contrastive fitness learning: Reprogramming protein language models for low-n learning of protein fitness landscape*, 470–474 (Springer, 2024).

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

[35] 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) .

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

[37] 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) .

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

[39] 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) .

[40] 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) .

[41] 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) .

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

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

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

[45] 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) .

[46] 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) .

[47] Sinai, S. *et al.* Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv preprint* (2020) .

[48] Barrera, L. A. *et al.* Survey of variation in human transcription factors reveals prevalent dna binding changes. *Science* **351** (6280), 1450–1454 (2016) .

[49] Chaudhury, S., Lyskov, S. & Gray, J. J. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics* **26** (5), 689–691 (2010) .

[50] Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive aav capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **366** (6469), 1139–1143 (2019) .

[51] Lorenz, R. *et al.* ViennaRNA package 2.0. *Algorithms for molecular biology* **6** (1), 26 (2011) .

[52] Angermueller, C. *et al. Model-based reinforcement learning for biological sequence design* (2019).

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

[54] Brookes, D. H., Park, H. & Listgarten, J. Conditioning by adaptive sampling for robust design. *arXiv preprint arXiv:1901.10060* (2019) .

[55] Hansen, N. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772* (2016) .

[56] Kirjner, A. *et al.* *Improving protein optimization with smoothed fitness landscapes* (OpenReview.net, 2024). URL https://openreview.net/forum?id=rxlF2Zv8x0.

[57] Wang, Y. *et al.* Self-play reinforcement learning guides protein engineering. *Nature Machine Intelligence* **5** (8), 845–860 (2023) .

[58] Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell systems* **12** (11), 1026–1045 (2021) .

[59] Qiu, Y., Hu, J. & Wei, G.-W. Cluster learning-assisted directed evolution. *Nature computational science* **1** (12), 809–818 (2021) .

[60] De Visser, J. A. G. & Krug, J. Empirical fitness landscapes and the predictability of evolution. *Nature Reviews Genetics* **15** (7), 480–490 (2014) .

[61] Tokuriki, N. & Tawfik, D. S. Stability effects of mutations and protein evolvability. *Current opinion in structural biology* **19** (5), 596–604 (2009) .

[62] Lenski, R. E., Barrick, J. E. & Ofria, C. Balancing robustness and evolvability. *PLoS biology* **4** (12), e428 (2006) .

[63] 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) .

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

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

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

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

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

[69] 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) .

[70] 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 .

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

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

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

[74] 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) .

[75] 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) .

[76] Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34** (18), 3094–3100 (2018) .

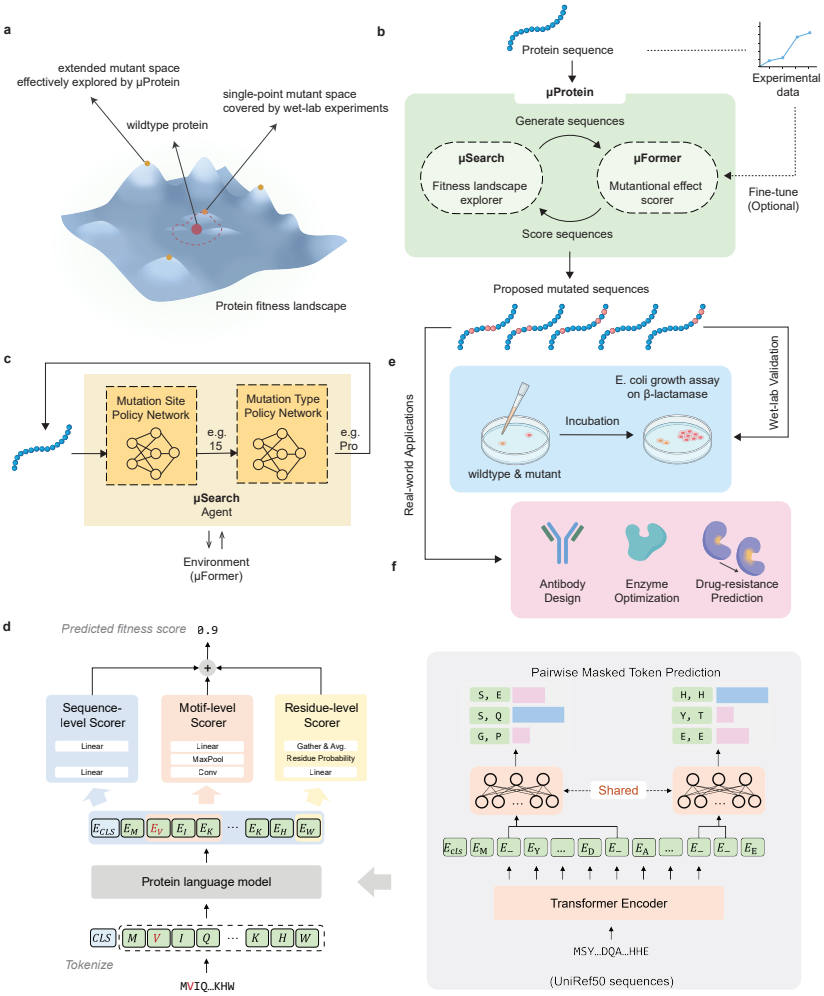


Fig. 1: Overview of μProtein. **a)** μProtein efficiently explores the vast protein fitness landscape, generalizing from limited experimental data, typically confined to single-point mutations. **b)** μProtein workflow. **c)** Overview of μSearch, a reinforcement-learning algorithm that uses a multi-step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

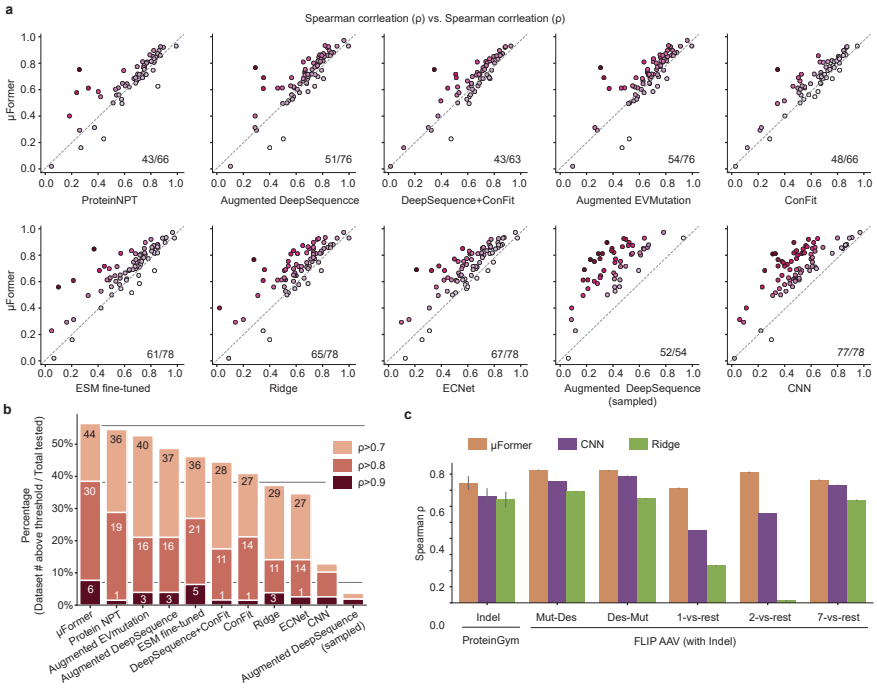


Fig. 2: Quantitative comparison of μFormer with alternative mutational effect prediction approaches. a) Pairwise comparison between μFormer and other approaches. Each data point represents one ProteinGym dataset. Spearman ρ is calculated to quantify the performance of corresponding approaches. The lower-right corner displays the number of datasets where μFormer outperformed the corresponding method compared to the total number of tested datasets. The Ridge model refers to the one trained on one-hot embeddings without augmentation. b) The number and percentage of datasets achieving high Spearman correlation for each model. The y-axis represents the percentage of datasets exceeding different Spearman ρ thresholds for each model, while the numbers on the bars represent the absolute counts of such datasets. Thresholds of Spearman $\rho = 0.7$, 0.8, and 0.9 were evaluated. Darker bars indicate higher Spearman thresholds. 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]. Results from 7 experiments are collectively shown. FLIP AAV: VP1 indel and substitution mutational effect dataset measured and processed by Bryant et al. [34] and Dallago et al. [27] respectively. Results from 10 replications with different random seeds are shown. 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. Center: mean. Error bar: standard error.

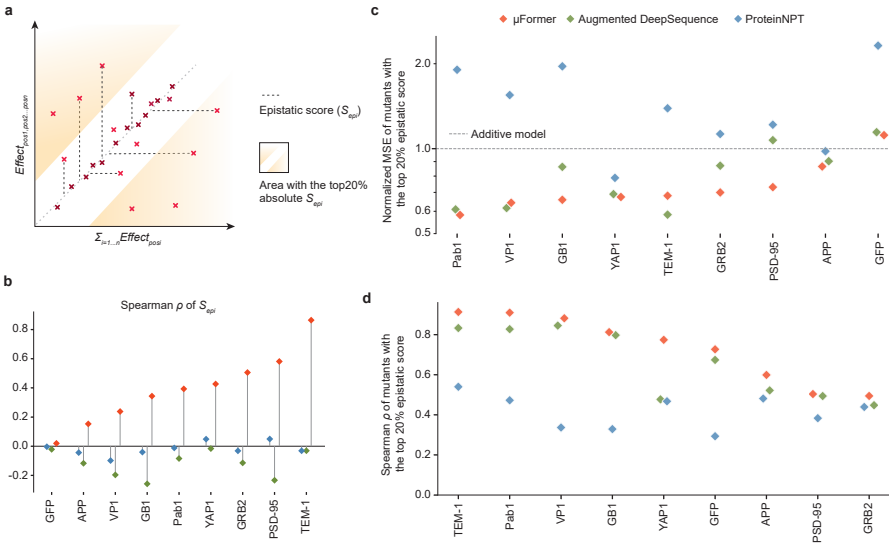


Fig. 3: μFormer models the epistasis effects in high-order mutants effectively. a) Schematic representation of the calculation of epistatic scores S_{epi}. High-order mutants with the top 20% absolute S_{epi} values are highlighted in the shaded regions.b) Spearman ρ between predicted and observed S_{epi} across proteins. c) Normalized mean squared error (MSE) of μFormer, ProteinNPT and Augmented DeepSequence for predictions of high-order mutants with the top 20% absolute S_{epi} values. Compared to the additive model (an epistasis-free estimation), μFormer reduces predictive errors in 8 out of 9 tested proteins. d) Spearman ρ of μFormer, ProteinNPT and Augmented DeepSequence for predictions of high-order mutants with the top 20% absolute S_{epi} values.

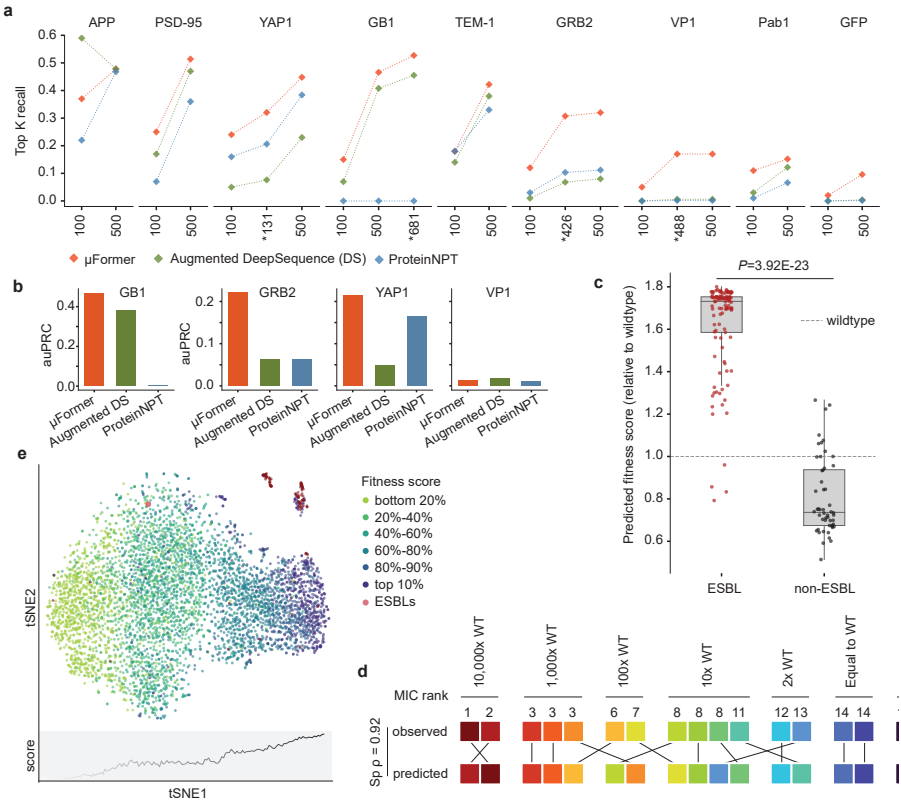


Fig. 4: μFormer effectively identifies high-functioning variants with multi-point mutations. a) Recall score of μFormer, Augmented DeepSequence and ProteinNPT for the Top K high-order mutants of 9 different proteins ranked by fitness values. K is equal to 100, 500, or customized values marked with asterisks. 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 and alternate methods in classifying ultra-high value mutants in GB1 (n=681), GRB2 (n=426), YAP1 (n=488), and VP1 (n=131). c) The predicted fitness scores of extended-spectrum β-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. n = 157. 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 [46]. 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 [45]. The bottom panel shows the average fitness score along PC1.

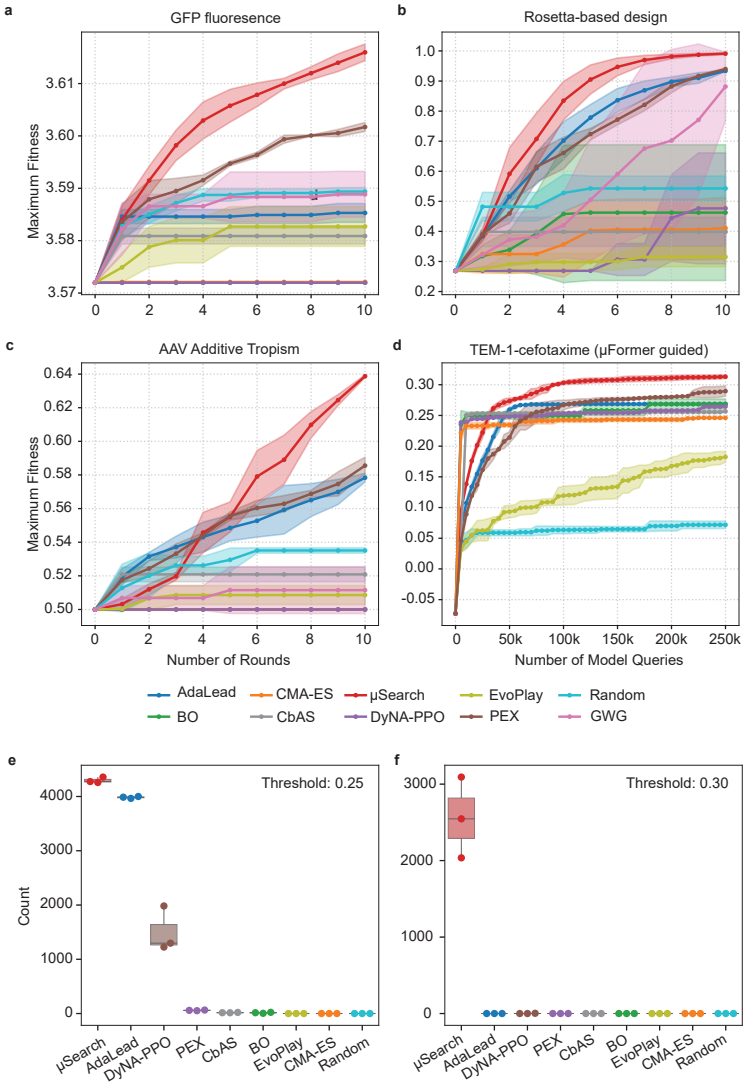


Fig. 5: Quantitative comparison of μSearch with prevalent fitness landscape exploration algorithms. a-c) Compare μSearch with eight leading methods: AdaLead, DyNA-PPO, CbAS, CMA-ES, Bayesian Optimization, PEX, GWG, and EvoPlay, as well as a Random Selection baseline on searching the landscapes of GFP Fluorescence (a), Rosetta-based Design (b), or AAV additive tropism (c) using a multi-round search setting. Maximum fitness: the highest fitness score achieved up to the current round. Center: mean. Error bands: standard deviation. d) Compare μSearch with alternative methods using the single-round search setting and μFormer as the oracle. A total of 250,000 oracle queries were made for each method. GWG was excluded due to its incompatibility with μFormer (It requires joint optimization with an updatable oracle, whereas μSearch is decoupled from the oracle, and integrating GWG would require fundamental pipeline changes). Center: mean. Error bands: standard deviation. e-f) The numbers of the mutants that surpass a fitness threshold of 0.25 (e) or 0.3 (f) in the setting for (d). Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers. Three

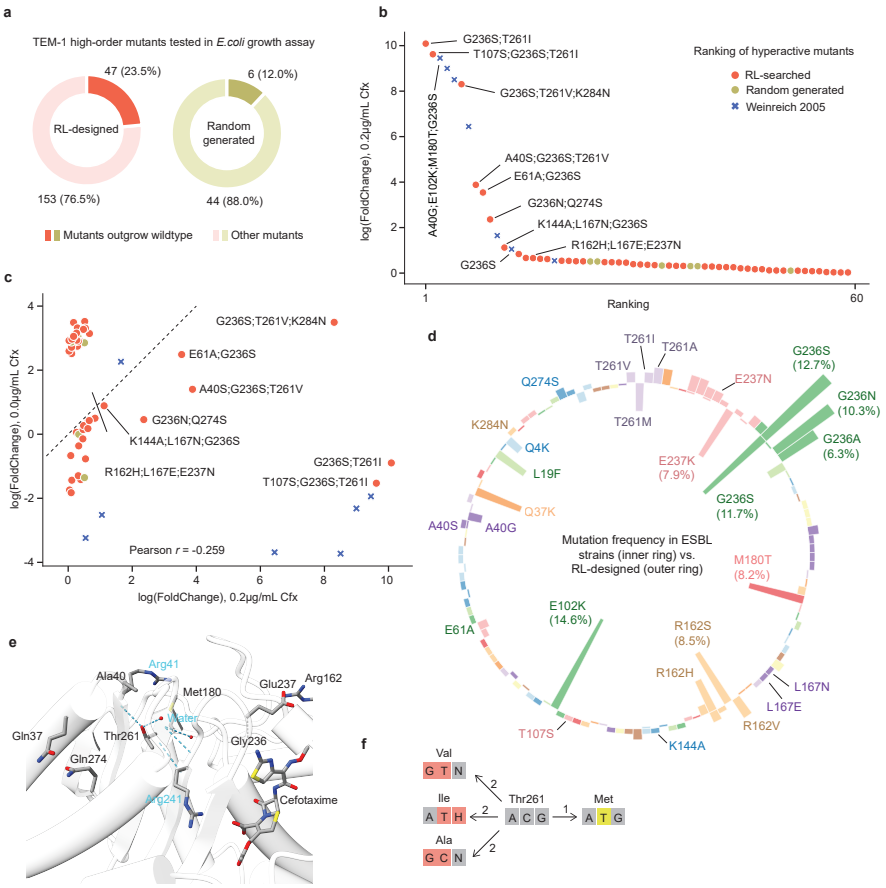
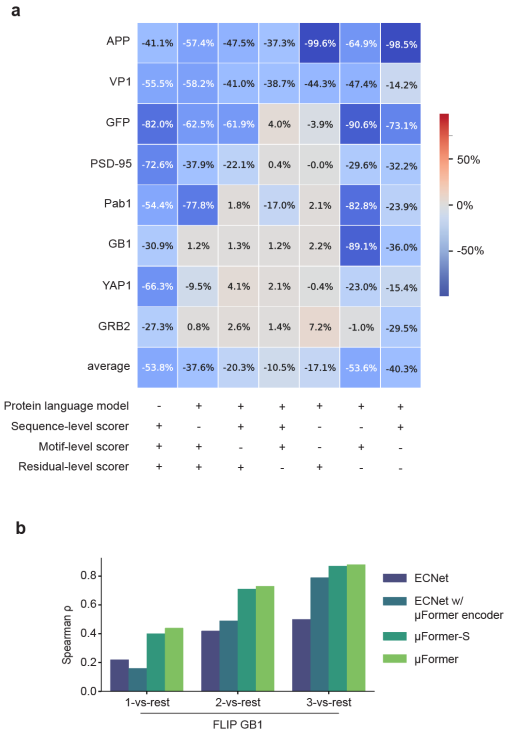
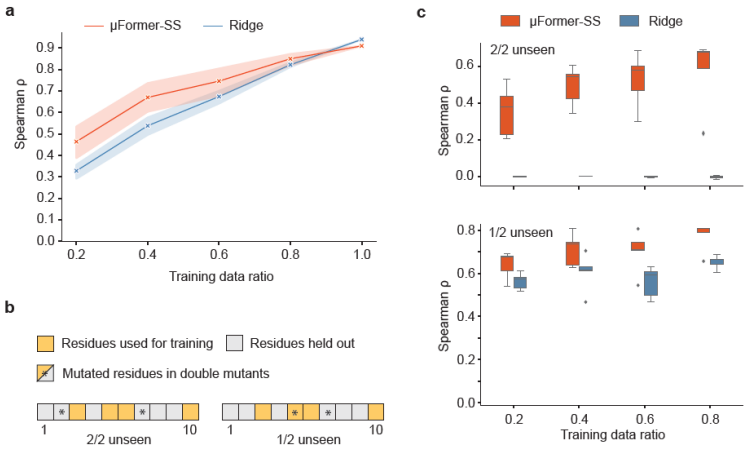


Fig. 6: Design high-functioning sequences with μProtein. a) Percentage of TEM-1 high-order mutants that provide a growth advantage to *E. coli* on cefotaxime compared to the wild type, shown for both RL-designed and randomly generated sequences b) Fold change in growth of TEM-1 mutants compared to wild-type *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) Distribution of mutations in extended-spectrum β-lactamases (ESBLs) (inner ring) compared to RL-designed mutants (outer ring). e) 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. f) Illustration depicting nucleotide mutations at position T261 in RL-designed mutants (left) and natural mutants (ESBL strains, right). 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.



Extended Data Figure 1. Ablation and analysis on μFormer components. **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 (Methods) is a variation with a model size similar to ECNet. 1-vs-rest: a train-test split where single-point mutants are used for training, and multi-point mutants are reserved for testing. 2-vs-rest: a train-test split where single- and double-point mutants are used for training, and all higher-order mutants are reserved for testing. 3-vs-rest: a train-test split where single-, double-, and triple-point mutants are used for training, and all higher-order mutants are reserved for testing. See Supplementary Notes for details.



Extended Data Figure 2. Analysis of μProtein. a) Performance of μFormer and Ridge 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). Center: mean. Error bands: standard deviation. Five experiments are performed for each setting with random selection on training data. b) 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. c) Performance of μFormer and Ridge on different splits of GB1 double mutants. Training data split criteria are the same as in a). Center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers. Five experiments are performed for each setting with random selection on training data.

Extended Data Table 1: Size of training (single-point mutants) and test (multi-point mutants) datasets in single-to-multi setting

Dataset	No. of training	No. of test
GB1	1,045	535,917
TEM-1	5,468	12,374
Pab1	1,187	36,521
PSD-95	1,280	5,696
GRB2	1,034	62,332
APP	468	14,015
GFP	1,084	50,630
YAP1	362	9,713
VP1	532	41,796