Protein Design by Directed Evolution Guided by Large Language Models

Thanh V. T. Tran and Truong Son Hy

***Abstract*—Directed evolution, a strategy for protein engineering, optimizes protein properties (i.e., fitness) by a rigorous and resource-intensive process of screening or selecting among a vast range of mutations. By conducting an *in silico* screening of sequence properties, machine learning-guided directed evolution (MLDE) can expedite the optimization process and alleviate the experimental workload. In this work, we propose a general MLDE framework in which we apply recent advancements of Deep Learning in protein representation learning and protein property prediction to accelerate the searching and optimization processes. In particular, we introduce an optimization pipeline that utilizes Large Language Models (LLMs) to pinpoint the mutation hotspots in the sequence and then suggest replacements to improve the overall fitness. Our experiments have shown the superior efficiency and efficacy of our proposed framework in the conditional protein generation, in comparision with other state-of-the-art baseline algorithms. We expect this work will shed a new light on not only protein engineering but also on solving combinatorial problems using data-driven methods. Our implementation is publicly available at https://github.com/HySonLab/Directed_Evolution.**

***Index Terms*—Directed evolution, protein engineering, machine learning, large language models.**

I. Introduction

PROTEIN are essential biomolecules that play a diverse range of critical roles in every living organism. They are involved in virtually every important activity that happens inside living things: digesting food, contracting muscles, moving oxygen throughout the body, attacking foreign viruses, etc. The effective resolutions to biological challenges can also serve as effective resolutions to human challenges, given the extensive utilization of proteins in various domains such as food, chemicals, consumer products, and medicinal applications. However, comprehending the complex structure and function of proteins remains an immensely challenging task, owing to the diversity of protein sequences and the three-dimensional structure they have. Therefore, computational biology and machine learning (ML) methods have emerged as indispensable tools among researchers and engineers. These methods offer the capability to process vast amounts of biological data, extract meaningful patterns, and make predictions that would be impractical through traditional experimental approaches alone [1]. Moreover, large language models have been pretrained on extensive protein sequence databases [2], [3], enabling them to capture intricate sequence-structure-function relationships. These models empower researchers to extract valuable insights from protein sequences, understand their evolution, and predict their behavior in various biological contexts.

In the field of protein engineering, fitness can be defined as performance on a desired property or function. Examples of fitness include catalytic activity for enzymes [4] and fluorescence for biomarkers [5]. Protein optimization seeks to improve protein fitness by altering the underlying sequences of amino acids. Nevertheless, the space of possible proteins is too large to search exhaustively naturally, in the laboratory, or computationally. The problem of the identification of ideal sequences is classified as NP-hard, as there currently exists no polynomial-time algorithm for efficiently exploring this particular domain [6]. Functional proteins are scarce in this vast space of sequences, and as the desired level of function increases, the number of sequences having that function decreases exponentially [7]. Consequently, the occurrence of functional sequences is rare and overshadowed by nonfunctional and mediocre sequences. In recent years, generative models have emerged as powerful tools for generating creative and diverse content, ranging from text [8], [9], images [10], [11] to speech [12] and more. Leveraging the success of generative models in text and image generation, numerous studies have been made to harness these techniques to design novel proteins with tailored properties and functions [13]–[15]. This developing paradigm promises to accelerate wet-lab experiments with data-driven models to find desirable proteins more efficiently.

Directed evolution, a laboratory methodology wherein biological entities possessing desired characteristics are generated through iterative cycles of genetic diversification and library screening or selection, has emerged as a highly valuable and extensively utilized instrument in both fundamental and practical realms of biological research [16]–[19]. This method was conceptualized using the natural process of evolution and the idea of natural selection as its primary sources of inspiration. In nature, organisms with advantageous traits are more likely to survive and reproduce, passing on their beneficial characteristics to the next generation. Similarly, directed evolution starts with a protein having some level of the desired function. Subsequently, a series of mutation and screening rounds are undertaken, wherein mutations are introduced (e.g., through site-saturation mutagenesis) to generate a collection of variant proteins. Through this iterative process, the most optimal variant is identified, and the cycle continues until a satisfactory level of improvement is achieved. Fig. 1A demonstrates the workflow of this algorithm.

Machine learning-guided directed evolution (MLDE) is a

Corresponding author: Truong Son Hy.

Truong Son Hy is with University of Alabama at Birmingham, United States (email: thy@uab.edu)

Both authors are with FPT Software AI Center, Hanoi, Vietnam (emails: ThanhTVT1@fpt.com, sonht13@fpt.com)

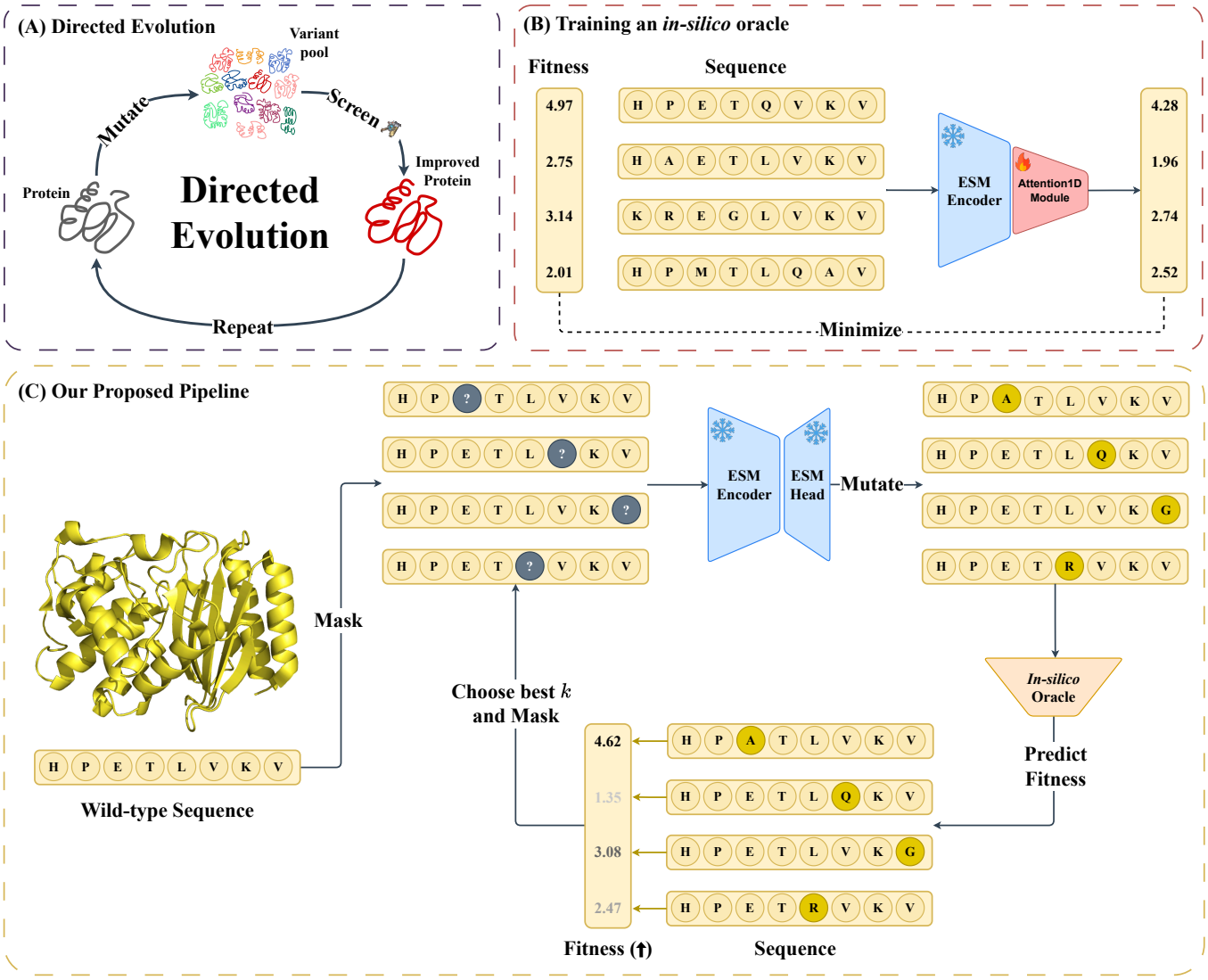

Fig. 1. **(A)** Workflow of traditional directed evolution: many variants are screened, and the best variant is selected as the parent for the next round of mutagenesis and screening. **(B)** We train the *in-silico* oracle as the "ground-truth" evaluator to predict the fitness score of each generated sequence. **(C)** Our proposed MLDE framework. The workflow begins by identifying a protein with activity for a target function. Once the starting point is identified, diversity is introduced through mutagenesis, and the fitness scores of the resulting variants are predicted by a trained *in-silico* oracle. Based on these predictions, we can select multiple variants as potential parent candidates for the subsequent iteration, balancing exploitation and exploration.

strategy for protein engineering that can be applied to a range of biological systems, reducing the burden of laboratory experiments by performing *in-silico* population selection through model-based fitness prediction. These techniques can leverage data from both improved and unimproved sequences in order to accelerate the process of evolution and broaden the range of properties that can be optimized. This is achieved by intelligently selecting new variants for screening, which enables the achievement of higher levels of fitness than what can be accomplished solely through directed evolution [20]–[22]. Employing this novel biology algorithm, we propose a deep learning-based framework for MLDE that optimizes directly on the discrete space of all possible sequences. Our pipeline can be described by Fig. 1C.

Our contributions can be summarized as follows:

- We propose a novel LLM-guided mutation prediction paradigm framework for *in-silico* directed evolution,
- We introduce a masking strategy that aids the identification of promising mutations,

II. Related Work

A. Directed Evolution

Directed Evolution is a classical paradigm for protein sequence design that has achieved several successes. Under this framework, ML algorithms play a crucial role in improving the sample efficiency of evolutionary search. [23] utilizes a fast Markov chain Monte Carlo sampler that uses gradients to propose promising mutations. [24] formulates the local search as a proximal optimization problem to derive an objective for searching the steepest improvement. [25] combines hierarchical unsupervised clustering sampling and supervised learning to guide protein engineering. This work is later improved by [26]. [27] provides an approach to target informative sequences

without the initial global search. Nevertheless, these methods have not yet incorporated large language models, which have proven efficient in multiple related tasks. Our proposal describes the combination of a protein language model yielding a highly favorable outcome.

B. Protein Property Prediction

One of the primary objectives of bioinformatics is the prediction of protein function, which can be applied to a vast array of biological issues, including the identification of drug targets and the comprehension of disease mechanisms. Numerous computational approaches have been developed to autonomously forecast protein function, with specific investigations concentrating on site- or domain-specific predictions [28], [29]. Traditional machine learning classifiers, such as support vector machines, random forests, and logistic regression, have been extensively used for protein function prediction [30], [31]. In recent years, deep learning has led to unprecedented improvements in the performance of methods tackling a broad spectrum of problems, including predicting protein function [32]–[37]. These developments facilitate our work as we employ attention, a well-known mechanism in deep learning, to acquire a comprehensive understanding of the protein sequence and anticipate its properties.

C. Protein Generation

Significant advancements have been made in the development of methodologies for the generation of functional protein sequences with desired features in recent times. Conventional approaches that utilize multiple sequence alignments of homologous proteins, such as ancestral sequence reconstruction [38], have exhibited efficacy in producing functional proteins, but with certain limitations in their applicability. To access a broader sequence space, different ML techniques have been employed. Some techniques, including reinforcement learning [39], [40], Bayesian optimization [41]–[43], importance sampling [44], and adaptive evolution method [45] have been used and achieved great results in their defined tasks. Modern algorithms such as language models provide a powerful architecture to learn from large sets of amino acid sequences across families for the purpose of generating diverse, realistic proteins [46]–[48]. Other generative techniques like variational auto-encoder [49], [50], generative adversarial networks [14], [51], and diffusion [52] also show promising outcomes in numerous protein engineering tasks. Yet, by applying MLDE to generating sequences, we find that this algorithm demonstrates strong performance, especially in comparison to other state-of-the-art (SOTA) methods.

III. Background

A. Notation and Problem Formulation

Through the process of directed evolution, protein fitness is enhanced by emulating natural selection through mutagenesis. From a mathematical standpoint, directed evolution can be conceptualized as an optimization problem for a black-box algorithm, seeking the optimal sequence $x^\star$

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x), \tag{1}$$

where $\mathcal{X}$ is the sequence mutation space, and $f(x)$ is an unknown sequence-to-fitness function for sequence x in $\mathcal{X}$. During the optimization, the algorithm choose the best mutated sequence from an intermediate generated population $\mathbf{x}$ (the size of the population $|\mathbf{x}|$ is n) so that the returned fitness $y^\star = f(x^\star)$ is the best fitness among values $\mathbf{y}$.

B. Transformer

Transformer [53] is a formidable force in the modern deep learning stack. Originally, Transformer follows an encoder-decoder scheme and employs a multi-headed self-attention mechanism. The fundamental concept underlying this mechanism is for each element within the sequence to acquire information from other tokens present in the same sequence. Concretely, given a tensor of sequence features $\mathbf{X} \in \mathbb{R}^{n \times d}$, Transformer computes three matrices including query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) via three linear transformations $\mathbf{Q} = \mathbf{X}\mathbf{W}_q^\top$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k^\top$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_v^\top$. A self-attention tensor ($\mathbf{H}$) can be computed as follows:

$$\mathbf{H} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_0}}\mathbf{V}\right) \tag{2}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are learnable parameters in $\mathbb{R}^{d_0 \times d}$, resulting in $\mathbf{H} \in \mathbb{R}^{n \times d_0}$. Moreover, each $\mathbf{H}$ in Eq. 2 denotes an attention head. The output of multiple heads $(\mathbf{H}_1, \ldots, \mathbf{H}_h)$ are then concatenated together to form a final tensor $\mathbf{H} = \text{Concat}(\mathbf{H}_1, \ldots, \mathbf{H}_h)$, where h is the number of attention heads. Finally, the new representation $\mathbf{X}'$ can be computed by feeding $\mathbf{H}$ into a feed-forward neural network (FFN).

C. ESM-2: Protein Language Model

Inspired by [54], ESM-2 [55] adopts the encoder-only Transformer architecture style with small modifications. The original Transformer uses absolute sinusoidal positional encoding to inform the model about token positions. These positional encodings are added to the input embeddings at the bottom of the encoder stack. Nevertheless, this approach demonstrates limited generalizability beyond the specific context window on which it is trained. The implementation of Rotary Positional Embedding (RoPE) [56] in ESM-2 enables the model to extrapolate beyond its trained context window with no additional computational effort. The ESM-2 model is capable of generating latent representations for individual amino acids inside a protein sequence. This is achieved through pre-training on a vast dataset consisting of millions of protein sequences including billions of amino acids. Specifically, given an input protein, 15% of amino acids are masked and ESM-2 is tasked with predicting these missing positions. Although this training objective only directly involves predicting missing amino acids, achieving a high degree of success requires the model to learn complex internal representations of its input. Consequently, the model is able to

produce a concise representation of the full protein sequence, without relying on three-dimensional information. This rich and meaningful representations of ESM-2 has aided numerous studies, including protein functional prediction [57], [58], protein structure prediction [55], protein-protein interaction prediction [59], protein multimodal representation [60], and protein design [61], [62].

IV. Method

This section describes the overall pipeline of our framework. Initially, after dividing the sequence population into multiple parts, we apply different masking strategies for each portion to retrieve a partially-masked population. These masked variants are then fed into ESM-2 to predict the tokens being masked. After that, the newly generated population is measured for fitness by a fine-tuned Attention1D. These steps are repeatedly iterated until they reach a pre-defined iteration number. The full workflow of our method is illustrated in Fig. 1C.

A. Selecting Mutation Positions

We follow the scheme of the text-based masked language model [54], which masks some of the tokens from the input and predicts the original vocabulary id of the masked. In the context of protein generation, we assume that the new prediction tokens can boost the fitness values of the protein sequence. In this study, we conducted two different masking strategies: **(1)** random masking and **(2)** importance masking. In the first scheme, we randomly replace token(s) in the sequence by the $\langle\texttt{MASK}\rangle$ tokens. For the latter, we follow the soft-masking strategy described in [63]. Concretely, when we split a protein sequence into multiple substrings of length k called *k-mers*, it creates a distribution based on the frequency of each k-mer. Therefore, when masking and recovering the sequences, k-mer with various importance should be treated differently. Following [63], we consider the following aspects to measure and define the importance of k-mers in one sequence:

1) Relevancy: We adopt the TF-IDF weights [64], $w_{\text{TF-IDF}}$, of the k-mer as one way to measure the relevance of k-mer w in one sequence x:

$$w_{\text{TF-IDF}}(w,x) = \frac{f_{w,x}}{\sum_{w'\in\mathbf{x}} f_{w',x}} \log \frac{N}{1+|\{x\in\mathcal{X} : w\in x\}|}, \tag{3}$$

where $f_{w,x}$ is the number of times that k-mer w occurs in sequence $\mathbf{x}$, N is the number of sequences in the corpus, and $\mathcal{X}$ is the set of sentences, and $|\{x\in\mathcal{X} : w\in x\}|$ is the number of sequences where the k-mer w appears. A higher weight for k-mer w in sequence x in TF-IDF means that the k-mer might be more important in the sequence.

2) Entropy: We also consider measuring the amount of information with entropy H [65] in the k-mer w to reflect the importance of that k-mer:

$$H(w) = -p(w)\log(p(w)), \tag{4}$$

where $p(w) = \frac{f_w}{\sum_{j=1}^{V} f_j}$ represents the probability of k-mer w and f is the number of appearances of k-mer in the corpus. A k-mer with lower entropy indicates that it might contain less information and thus be less important compared to the one with higher entropy.

In practice, we combine these two measures (with normalization) to decide the importance I of the k-mer w in one sequence x by:

$$I(w) = \frac{w_{\text{TF-IDF}}(w,x)}{\sum_{w'\in x} w_{\text{TF-IDF}}(w',x)} + \frac{H(w)}{\sum_{w'\in d} H(w')}. \tag{5}$$

Based on the introduced importance I of the k-mers in a sequence, we decide to replace k-mer(s) with the lowest importance value(s) by $\langle\texttt{MASK}\rangle$ token(s). By doing this, models could replace the least important k-mers with better ones for better generation quality.

B. Machine Learning-guided Improvement

Algorithm 1 Machine Learning-guided Directed Evolution

Input: Wild-type x^{wt}, fitness y^{wt}, population n, beam m, iterations N, random ratio α, k-mer size k

Output: population sequence $\mathbf{x}_{\text{top}}$, population fitness $\mathbf{y}_{\text{top}}$

$\mathbf{x}_{\text{prev}} \leftarrow$ Duplicate x^{wt} for n times
$\mathbf{y}_{\text{prev}} \leftarrow$ Duplicate y^{wt} for n times
for $i = 1, \ldots, N$ **do**
  $\mathbf{x}_{\text{cur}} \leftarrow$ Duplicate sequences in $\mathbf{X}_{\text{prev}}$ for m times
  $L \leftarrow \text{len}(\mathbf{x}_{\text{cur}})$
  // Masking
  $\mathbf{x}_{\text{cur}} \leftarrow \textsc{Shuffle}(\mathbf{x}_{\text{cur}})$
  $\mathbf{x}_1 \leftarrow \textsc{RandomMask}(\mathbf{x}_{\text{cur}}[:\text{int}(\alpha L)], k)$
  $\mathbf{x}_2 \leftarrow \textsc{ImportanceMask}(\mathbf{x}_{\text{cur}}[\text{int}(\alpha L):], k)$
  $\mathbf{x}_{\text{mask}} \leftarrow \textsc{Concat}(\mathbf{x}_1, \mathbf{x}_2)$
  // Predicting mutation and extracting features
  $\mathbf{X}_{\text{repr}}, \mathbf{R}_{\text{feat}} \leftarrow \text{ESM2}(\mathbf{x}_{\text{mask}})$
  // Replacing $\langle\texttt{MASK}\rangle$ tokens with predicted ones
  $\mathbf{x}_{\text{mut}} \leftarrow \textsc{Argmax}(\textsc{Softmax}(\mathbf{X}_{\text{repr}}))$
  // Predicting fitness
  $\mathbf{y}_{\text{mut}} \leftarrow \textsc{Attention1D}(\mathbf{R}_{\text{feat}})$
  // Choosing best variants
  $\mathbf{x}_{\text{comb}} \leftarrow \textsc{Concat}(\mathbf{x}_{\text{mut}}, \mathbf{x}_{\text{prev}})$
  $\mathbf{y}_{\text{comb}} \leftarrow \textsc{Concat}(\mathbf{y}_{\text{mut}}, \mathbf{y}_{\text{prev}})$
  $\mathbf{x}_{\text{top}}, \mathbf{y}_{\text{top}} \leftarrow \textsc{TopK}(\mathbf{x}_{\text{comb}}, \mathbf{y}_{\text{comb}}, n)$
  $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}_{\text{top}}$
  $\mathbf{y}_{\text{prev}} \leftarrow \mathbf{y}_{\text{top}}$
end for

After the generation of a population of masked sequences, these sequences are subsequently inputted into the ESM-2 model. Following this, the neural network generates predictions for the masked tokens and produces a latent representation of these variations. In the final stage, we evaluate the sequences that have been newly proposed by measuring their fitness values. In practice, validating these generated sequences experimentally is costly and time-intensive. We follow prior works [24], [44] in leveraging a trained evaluator model (i.e., oracle) as a proxy for wet-lab measurements. This oracle operates on the input representation $\mathbf{X}' \in \mathbb{R}^{B\times L\times d_0}$ derived

TABLE I
DETAILED INFORMATION AND STATISTICS OF THE EIGHT PROTEIN DATASETS.

Dataset	Organism	Protein	Optimization Target	Length	Size	Percentiles 0.25	0.50	0.75
avGFP [66]	Aequorea victoria	GFP	Brightness	237	51,715	1.428	3.287	3.161
AAV [67]	Homo sapiens	VP1	AAV viabilities	28	42,330	−3.964	−0.840	1.321
TEM [68]	Escherichia coli	TEM-1 β-Lactamase	Thermodynamic stability	286	5,199	0.049	0.444	0.934
E4B [69]	Mus musculus	UBE4B	Ubiquitin ligase activity	102	91,032	−1.830	−0.984	−0.093
AMIE [70]	Escherichia coli	Amidase	Hydrolysis activity	341	6,417	−1.228	−0.666	−0.263
LGK [71]	Lipomyces starkeyi	Levoglucosan kinase	Levoglucosan utilization	439	7,633	−0.871	−0.562	−0.394
Pab1 [72]	Saccharomyces cerevisiae	Poly(A)-binding	mRNA binding	75	36,389	−0.116	−0.022	0.036
UBE2I [73]	Homo sapiens	UBE2I	Growth rescue rate	159	3,022	0.068	0.492	0.766

from a pre-trained ESM model after processing a protein sequence. The architecture of this module is as follows:

$$\mathbf{X}_2 = \mathrm{ReLU}(\mathrm{Linear}_1(X')) \quad \in \mathbb{R}^{B\times L\times d_{\text{hid}}} \tag{6}$$

$$\mathbf{X}_3 = \mathrm{ReLU}(\mathrm{Linear}_2(X_2)) \quad \in \mathbb{R}^{B\times L\times d_{\text{hid}}} \tag{7}$$

$$\mathbf{X}_4 = \mathrm{Attention1D}(\mathbf{X}_3) \quad \in \mathbb{R}^{B\times d_{\text{hid}}} \tag{8}$$

$$\mathbf{X}_5 = \mathrm{ReLU}(\mathrm{Linear}_3(X_4)) \quad \in \mathbb{R}^{B\times d_{\text{hid}}} \tag{9}$$

$$\mathbf{X}'' = \mathrm{ReLU}(\mathrm{Linear}_4(X_5)) \quad \in \mathbb{R}^{B\times 1} \tag{10}$$

where B is the number of sequences, L is the input length, and d_{hid} is the hidden dimension of the module. As illustrated in the Figure 1B, this model is subsequently trained to comprehend the fitness landscape of the specified protein sequence. The values are thereafter arranged in descending order to generate the most optimal sequences that will proceed to the subsequent iteration of the pipeline.

In summary, our pipeline can be described by Algorithm 1. It is important to note that while selecting the best variants in each iteration, we combine the present "effective" population with the population from the preceding iteration. This procedure guarantees that in the event of mutations resulting in a reduction in fitness values, the pipeline will opt for the variations from the preceding round rather than the inferior ones.

V. EXPERIMENTS

In this section, we conduct extensive experiments to validate the effectiveness of our proposed pipeline on protein sequence design task.

A. Datasets

Following [24] and [44], we evaluate our method on the following eight protein engineering benchmark datasets: (1) Green Fluorescent Protein (**avGFP**), (2) Adeno-Associated Viruses (**AAV**), (3) TEM-1 β-Lactamase (**TEM**), (4) Ubiquitination Factor Ube4b (**E4B**), (5) Aliphatic Amide Hydrolase (**AMIE**), (6) Levoglucosan Kinase (**LGK**), (7) Poly(A)-binding Protein (**Pab1**), (8) SUMO E2 Conjugase (**UBE2I**). Table I provides comprehensive details of the eight datasets utilized in this study, including the protein name, its organism, sequence length, dataset size, optimization target, and percentile distribution. Each dataset corresponds to different optimization targets, which we simplify as a singular term *"fitness"* when reporting the benchmark results in this paper. The detailed data descriptions are provided in the Supplementary Material.

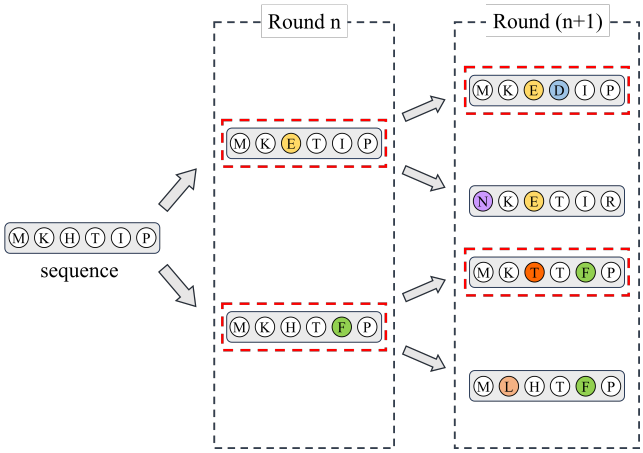

Fig. 2. Illustration of beam search adaptation in directed evolution. In this figure, the population number is 2, and the beam size is 2, which makes the *effective* population size 4 (i.e., sequences in the third generation before dropping inferior ones). The sequences covered in red boxes are chosen to move to the next iteration.

B. Implementation Details

Starting with a population of wild-type protein sequences, we split them into two separate groups. Each group is then subjected to a specific masking strategy, resulting in a population of masked sequences. These sequences are then fed to the pretrained 35-million-parameter version of the ESM-2 model, which generates the representation of the sequences and introduces mutations to sequences by replacing the ⟨MASK⟩ with the amino acid that the model confidently predicted. This representation is then fed into the oracle described in Section IV-B to predict the fitness value of sequences. Subsequently, the population is arranged in a descending order to select the optimal n variations according to their respective fitness values. These variants are the prospective candidates that will undergo the same pipeline in the subsequent iteration. Following several rounds, we would observe enhanced protein sequences exhibiting improved fitness ratings.

In our settings, we set the population size to 128 following [40]. Inspired by the beam search algorithm, which has proven its efficiency in the field of natural language processing [74], [75], the model proposes 4 separate mutated sequences (i.e., a beam size of 4), instead of 1 mutation in traditional directed evolution, to each variant in the population, making the *effective* population size to 512. This adaptation

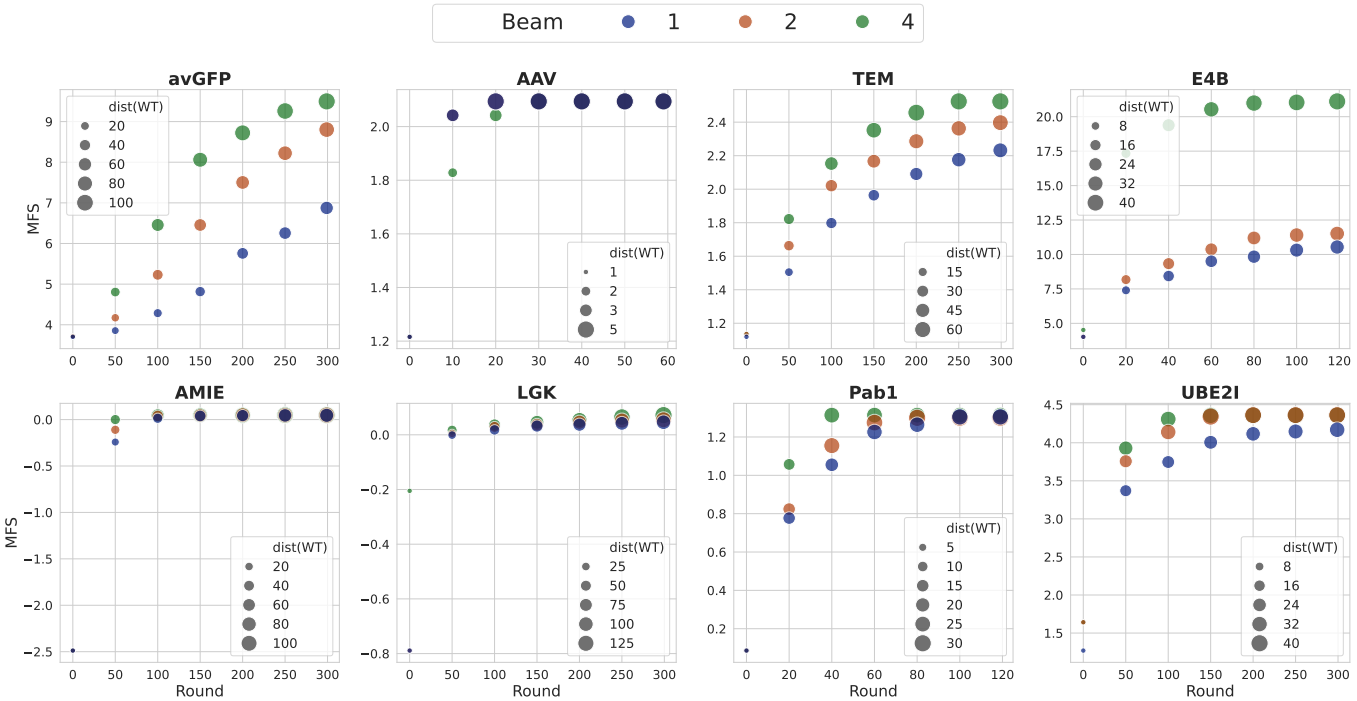

Fig. 3. Effects of different beam sizes (k-mer size is fixed to 1) on eight tasks. The x-axis is the number of rounds, and the y-axis is the fitness score. Size of dots represents the edit distance from those particular sequences to wild-type one.

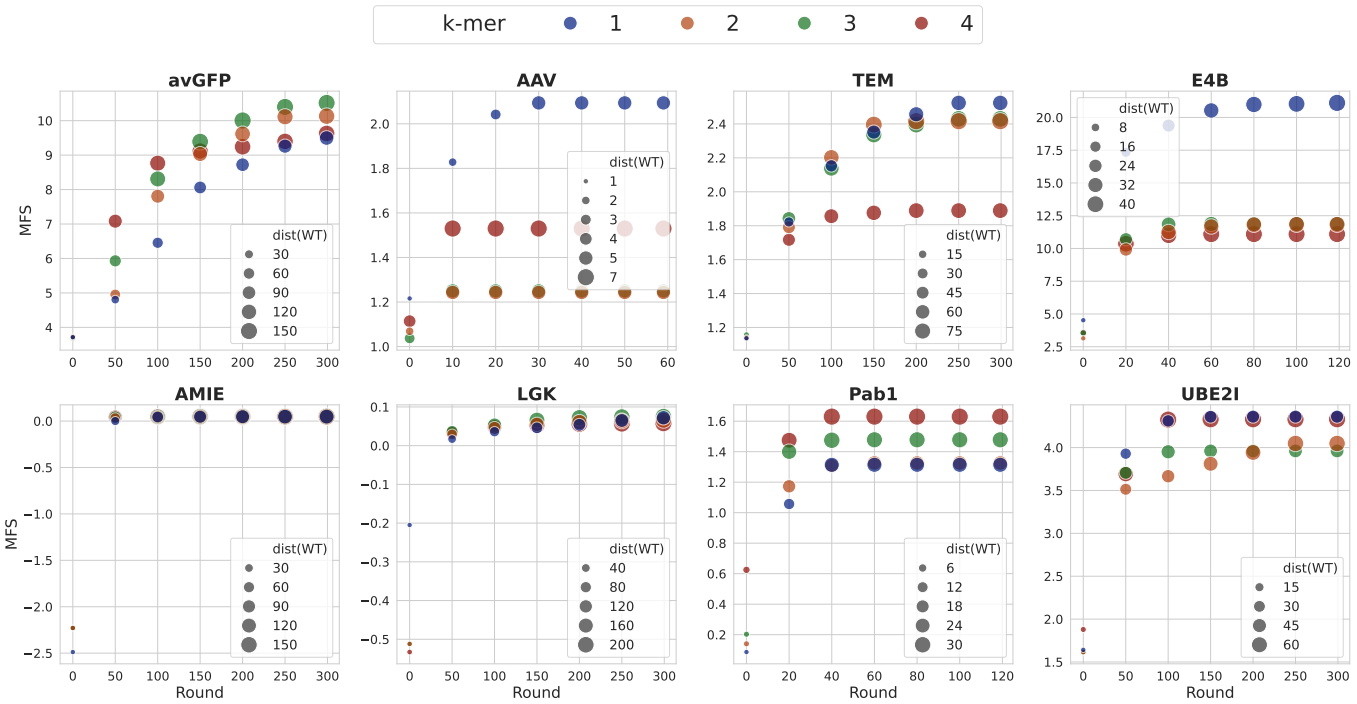

Fig. 4. Effects of different k-mer sizes to mask (beam size is fixed to 4) on eight tasks. The x-axis is the number of rounds, and the y-axis is the fitness score. Size of dots represents the edit distance from those particular sequences to wild-type one.

TABLE II
MAXIMUM FITNESS SCORES (MFS) OF ALL METHODS ACROSS EIGHT DATASETS. HIGHER VALUES INDICATE BETTER FUNCTIONAL PROPERTIES IN THE DATASET. **BOLD RESULTS** INDICATE THE BEST VALUE AMONG ALL METHODS ASSESSED USING THE SAME ORACLES.

Methods	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
AdaLead	3.23 ± 0.78	-1.55 ± 0.37	0.25 ± 0.23	-0.37 ± 0.14	-1.48 ± 0.27	-0.05 ± 0.04	0.38 ± 0.39	2.77 ± 0.21	0.41
DyNA PPO	5.33 ± 0.47	-2.82 ± 0.17	0.57 ± 0.05	-0.58 ± 0.05	-2.79 ± 0.00	-0.06 ± 0.01	0.18 ± 0.01	2.63 ± 0.06	0.31
CMA-ES	5.13 ± 0.24	-3.27 ± 0.16	0.59 ± 0.05	-0.66 ± 0.10	-2.79 ± 0.00	-0.09 ± 0.03	0.25 ± 0.18	2.53 ± 0.10	0.22
DbAS	5.14 ± 0.22	-2.89 ± 0.16	0.49 ± 0.02	-0.43 ± 0.34	-2.09 ± 0.20	-0.03 ± 0.03	0.23 ± 0.09	2.74 ± 0.18	0.40
CbAS	5.19 ± 0.34	-2.80 ± 0.38	0.48 ± 0.03	-0.66 ± 0.03	-1.78 ± 0.69	-0.06 ± 0.03	0.28 ± 0.22	2.69 ± 0.05	0.42
COMs	3.54 ± 0.93	-3.53 ± 0.63	0.47 ± 0.07	-0.86 ± 0.02	-20.18 ± 1.66	-0.09 ± 0.02	0.16 ± 0.01	2.09 ± 0.07	-2.31
PEX	3.80 ± 0.08	2.38 ± 0.70	0.25 ± 0.13	4.32 ± 1.90	-0.36 ± 0.09	0.01 ± 0.00	1.33 ± 0.21	3.58 ± 0.74	1.91
GFN-AL	5.03 ± 0.13	-4.44 ± 0.13	0.65 ± 0.10	-0.83 ± 0.85	-37.36 ± 1.92	-5.74 ± 0.11	**1.40 ± 0.07**	3.85 ± 0.32	-4.68
GGS	3.37 ± 0.00	**2.44 ± 0.25**	1.12 ± 0.03	-1.15 ± 0.05	-3.36 ± 0.51	-0.97 ± 0.73	0.06 ± 0.01	4.10 ± 0.00	0.70
MLDE	**9.50 ± 0.37**	2.09 ± 0.00	**2.54 ± 0.06**	**11.56 ± 0.14**	**0.05 ± 0.00**	**0.07 ± 0.01**	1.32 ± 0.06	**4.41 ± 0.12**	**3.94**
− w/o RM	6.09 ± 0.05	2.04 ± 0.00	2.06 ± 0.00	10.41 ± 0.07	0.02 ± 0.00	0.01 ± 0.00	0.94 ± 0.03	4.23 ± 0.03	3.23
− w/o IM	9.05 ± 0.47	2.10 ± 0.12	2.35 ± 0.14	10.91 ± 0.08	0.04 ± 0.00	0.05 ± 0.01	1.12 ± 0.05	**4.56 ± 0.13**	3.77

TABLE III
DIVERSITY OF ALL METHODS ACROSS EIGHT DATASETS. THIS TABLE PROVIDES INSIGHT INTO THE EXPLORATION AND EXPLOITATION TRADE-OFF AMONG METHODS.

Methods	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
AdaLead	9.81 ± 3.23	6.90 ± 0.52	8.07 ± 4.11	6.90 ± 1.56	7.43 ± 0.37	7.44 ± 3.96	7.44 ± 4.13	7.42 ± 0.58	7.68
DyNA PPO	204.38 ± 0.14	114.23 ± 0.13	157.96 ± 0.78	140.62 ± 0.15	171.12 ± 0.22	205.10 ± 1.11	185.16 ± 0.09	179.21 ± 0.11	169.72
CMA-ES	173.37 ± 4.23	20.40 ± 0.58	210.14 ± 1.76	73.65 ± 0.22	248.20 ± 2.97	319.89 ± 54.05	54.05 ± 1.64	116.78 ± 1.34	152.06
DbAS	205.72 ± 0.06	25.14 ± 0.03	247.72 ± 0.04	89.60 ± 0.05	294.52 ± 0.06	378.22 ± 0.06	66.42 ± 0.06	138.73 ± 0.07	180.69
CbAS	205.71 ± 0.05	25.17 ± 0.02	247.63 ± 0.03	89.63 ± 0.01	294.50 ± 0.08	378.31 ± 0.11	66.43 ± 0.05	138.79 ± 0.08	180.77
COMs	70.32 ± 9.97	45.76 ± 0.39	73.19 ± 6.49	68.40 ± 0.50	100.73 ± 1.70	126.62 ± 10.61	115.24 ± 0.38	109.65 ± 2.69	88.74
PEX	7.05 ± 0.97	4.78 ± 0.51	6.37 ± 0.90	6.20 ± 0.94	8.00 ± 0.93	7.84 ± 1.16	7.01 ± 1.95	8.27 ± 1.73	6.94
GFN-AL	6.25 ± 0.25	0.50 ± 0.02	1.23 ± 0.04	29.06 ± 0.44	46.58 ± 0.22	112.75 ± 2.51	27.13 ± 0.65	2.35 ± 0.07	28.23
GGS	4.63 ± 0.54	12.71 ± 0.78	7.39 ± 0.67	10.58 ± 1.57	14.69 ± 0.14	16.15 ± 0.25	16.73 ± 0.38	3.01 ± 0.17	10.74
MLDE	4.92 ± 1.48	1.90 ± 0.07	2.32 ± 0.85	2.32 ± 0.84	4.86 ± 1.60	6.83 ± 1.20	1.69 ± 2.14	1.04 ± 0.70	3.11

TABLE IV
NOVELTY OF ALL METHODS ACROSS EIGHT DATASETS. THIS TABLE PROVIDES INSIGHT INTO THE EXPLORATION AND EXPLOITATION TRADE-OFF AMONG METHODS.

Methods	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
AdaLead	13.49 ± 5.16	17.81 ± 0.48	41.41 ± 25.30	48.93 ± 2.71	41.17 ± 3.94	78.87 ± 32.10	73.53 ± 20.64	78.59 ± 12.63	49.23
DyNA PPO	201.70 ± 0.49	111.57 ± 0.10	156.78 ± 1.00	139.23 ± 0.27	170.37 ± 0.32	205.82 ± 0.94	185.69 ± 0.15	179.80 ± 0.27	168.87
CMA-ES	202.11 ± 0.43	20.77 ± 0.31	247.97 ± 1.08	85.57 ± 1.44	232.00 ± 9.96	375.44 ± 12.31	63.95 ± 1.30	138.07 ± 0.83	170.74
DbAS	201.80 ± 0.19	21.26 ± 0.15	247.49 ± 0.37	86.29 ± 0.12	293.44 ± 0.22	382.31 ± 0.37	64.93 ± 0.17	138.80 ± 0.19	179.53
CbAS	201.79 ± 0.31	21.31 ± 0.08	247.44 ± 0.35	86.13 ± 0.17	293.47 ± 0.20	382.27 ± 0.07	65.06 ± 0.20	138.91 ± 0.16	179.55
COMs	184.18 ± 0.35	98.83 ± 0.53	111.93 ± 1.66	101.93 ± 0.32	123.59 ± 1.95	150.66 ± 1.10	134.30 ± 1.97	129.80 ± 0.81	129.40
PEX	4.32 ± 0.69	1.91 ± 0.23	4.58 ± 0.89	5.12 ± 0.45	4.13 ± 0.66	16.35 ± 0.96	4.32 ± 1.40	5.40 ± 1.61	5.77
GFN-AL	220.63 ± 8.95	24.95 ± 1.21	255.94 ± 5.06	90.63 ± 1.03	326.26 ± 4.97	413.95 ± 2.08	64.05 ± 0.51	145.46 ± 1.92	192.73
GGS	3.55 ± 0.27	2.06 ± 0.15	4.03 ± 0.36	8.08 ± 0.33	9.99 ± 0.01	20.99 ± 0.01	8.38 ± 0.10	2.86 ± 0.16	7.49
MLDE	103.35 ± 2.11	2.58 ± 0.02	70.16 ± 2.45	28.30 ± 2.63	122.83 ± 8.07	156.66 ± 5.76	23.46 ± 1.26	41.02 ± 4.68	68.55

is illustrated in Fig. 2. We configure the random ratio to 0.6 and the k-mer size to 1 for the remaining hyperparameters. It is important to acknowledge that the outcomes generated from these configurations are not guaranteed to be optimal. The examination of the influence of each hyperparameter is conducted in Section V-F.

To ensure unbiased evaluation and avoid circular use of oracles, which can potentially cause information leakage, we employ two separate oracles for each fitness dataset: (1) the optimization oracle that guides the model optimization and (2) the evaluation oracle that assesses the performance of every methods. Following [24], we construct both oracles by freezing the ESM-based encoders and training an Attention1D module stacked atop to predict fitness scores. We use the pre-trained 33-layer ESM-2 for the former and the trained oracle provided by [24] for the latter. Details on these oracles are provided in the Supplementary Material. For each algorithm, we run all the experiments five times and report the average scores with their standard deviation.

C. Baseline Algorithms

We assess the performance of our method in comparison to several representative baselines: (1) **AdaLead** [76] is an advanced implementation of model-guided evolution. (2) **DyNA PPO** [39] applies proximal policy optimization to search sequences on a learned landscape model. (3) **CMA-ES** [77] is a famous evolutionary search algorithm. (4) **DbAS** [78] is a probabilistic modeling framework employing an adaptive

sampling algorithm. (5) **CbAS** [79] is an improvement over DbAS that conditions on desired properties. (6) **COMS** [80] is conservative objective models for offline MBO. (7) **PEX** [24] is a model-guided sequence design algorithm using proximal exploration. (8) **GFN-AL** [40] applies GFlowNet to design biological sequences. (9) **GGS** [81] is a graph-based smoothing method to optimize protein sequences. To ensure precise evaluation, we re-execute and re-evaluate baseline methods using the same oracle. For implementations from (1) to (5), we utilized the open-source implementation provided by [76]. As for other baseline methods, we adapted and modified the codes released by their respective authors. Details about the hyperparameter selection for these baselines are discussed in the Supplementary Material.

D. Evaluation Metrics

We employ three metrics from [40] to assess our sequence design pipeline: (1) **MFS**: Maximum fitness score, (2) **Diversity**, and (3) **Novelty**. Additionally, we consider two additional metrics for robustness: (4) **AFS**: Average fitness score, and (5) **dist(WT)**: the distance between the best-designed sequence and wild-type one. Detailed descriptions and mathematical formulations of these metrics can be found in the Supplementary Material. It's important to note that while greater diversity, novelty, and dist(WT) provide insights into the exploration and exploitation trade-offs, they do not necessarily indicate superior performance.

E. Experimental Results

1) Comparison with other baselines: As described in Table II, our approach exhibits superior fitness scores in 7 out of 8 protein families when evaluated against methods utilizing the same oracle. Notably, it is observed that AAV and Pab1 contain sequences with limited lengths, up to 28 and 76 amino acids, respectively. However, for datasets featuring longer sequences, leveraging the capabilities of protein language models trained on millions of protein sequences notably enhances MLDE's optimization capabilities. This is particularly evident on the avGFP and E4B benchmarks, where our method achieves statistically significant outperformance over the baselines. These findings suggest that our approach adeptly harnesses the potential of protein language models to introduce novel mutations into existing proteins.

Additionally, Tables III and IV outline the diversity and novelty metrics for all methods. It is crucial to emphasize that while these results offer insights into the exploration and exploitation trade-off, they may not necessarily reflect the efficacy of an algorithm, as it largely depends on the specific objectives of each method. Some methods aim to optimize the fitness score while maximizing diversity and novelty [44], while others may prioritize minimizing novelty [24], and some may solely focus on optimizing the fitness score alone [81]. Moreover, a random algorithm can achieve maximum diversity and novelty scores. This is evident in the case of GFN-AL and COMs in the AMIE dataset, where it attains high novelty and diversity score, yet achieves the lowest MFS and AFS among all methods.

2) Average performance: Another crucial factor determining the efficacy of an optimization algorithm is its ability to consistently produce high-quality samples within the population. Table V illustrates this aspect through the computation of the AFS, showcasing our method's prowess as it consistently outperforms all evaluated methods across 8 benchmarks. As a result, our method achieves the top performance, highlighting the highest average score of 3.669, surpassing the runner-up, PEX, by a significant margin of 5.6-fold.

3) Mutations to best design: Table VI presents the number of mutations required to transform the wild-type sequence into the best-designed sequence within our final proposed population, along with the corresponding percentages relative to the actual length of the protein. Overall, our method modifies the wild-type sequences within the range of 17.9% to 45.4%, with a predominant occurrence around 31% for the majority of tasks. These figures align with natural protein diversity, as proteins within the same family exhibit variability. For instance, [82] demonstrated the existence of multiple GFP proteins in nature, with cgreGFP exhibiting the highest fluorescence despite only sharing 41% sequence similarity with the avGFP utilized in our study.

F. Analysis

1) Effect of masking strategy: We conduct an ablation study to demonstrate how each masking strategy contributes to the performance. In particular, we compare MLDE with its variants, including (1) without random masking (w/o RM): we do not use random masking for MLDE and (2) without importance masking (w/o IM): we only rely on random masking to select positions to mutate. As shown in the lower section of Table II, removing either component leads to a decrease in algorithm performance. This observation highlights the effectiveness of employing both strategies, affirming the efficacy of our proposed masking approach.

2) Effect of beam size: The impact of varying beam sizes is depicted in Fig. 3. The graphic illustrates that increasing the beam size leads to accelerated convergence of the pipeline and results in sequences with higher fitness values throughout each iteration. This further underscores the effectiveness of incorporating the beam search technique.

3) Effect of k-mer size: The efficacy of selecting the k-mer size for sequence masking is illustrated in Fig. 4. It is evident that the influence of the k-mer size varies across datasets. For instance, while the distinction among settings is evident in the AAV dataset, with a k-mer size of 1 being the optimal setting, no significant difference is observed when we examine LGK and AMIE datasets. Therefore, it is apparent that the impact of the k-mer size fluctuates and requires careful adjustment in order to achieve optimal outcomes.

4) Validation of generated sequences: To comprehensively evaluate the generated samples, we assess the folding ability of proteins by using ESMFold [55]. There are three main reasons for us to decide utilizing ESMFold: (1) It generates **state-of-the-art three-dimensional structure**

TABLE V
AVERAGE FITNESS SCORES (AFS) OF ALL METHODS ACROSS EIGHT DATASETS. HIGHER VALUES INDICATE BETTER FUNCTIONAL PROPERTIES IN THE DATASET. **BOLD RESULTS** INDICATE THE BEST VALUE AMONG ALL METHODS ASSESSED USING THE SAME ORACLES.

Methods	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
AdaLead	2.94 ± 0.63	-1.88 ± 0.33	0.23 ± 0.19	-0.47 ± 0.09	-2.11 ± 0.26	-0.07 ± 0.05	0.33 ± 0.30	2.56 ± 0.18	0.19
DyNA PPO	2.42 ± 0.09	-4.31 ± 0.11	0.21 ± 0.03	-1.06 ± 0.04	-40.91 ± 0.83	-2.71 ± 1.27	0.12 ± 0.00	1.85 ± 0.03	-5.55
CMA-ES	2.61 ± 0.33	-5.95 ± 0.14	0.19 ± 0.03	-1.33 ± 0.07	-38.68 ± 1.21	-9.41 ± 0.69	0.11 ± 0.01	1.93 ± 0.06	-6.32
DbAS	2.54 ± 0.04	-4.70 ± 0.08	0.13 ± 0.01	-1.10 ± 0.02	-41.52 ± 0.33	-5.08 ± 0.52	0.12 ± 0.00	1.90 ± 0.02	-5.97
CbAS	2.54 ± 0.07	-4.65 ± 0.07	0.13 ± 0.01	-1.11 ± 0.02	-41.24 ± 0.38	-5.72 ± 0.21	0.12 ± 0.00	1.92 ± 0.03	-6.00
COMs	1.75 ± 0.04	-6.07 ± 0.03	0.01 ± 0.01	-1.38 ± 0.02	-35.92 ± 0.43	-12.55 ± 0.38	0.09 ± 0.00	1.35 ± 0.03	-6.59
PEX	3.14 ± 0.14	-1.10 ± 0.45	0.13 ± 0.02	1.80 ± 1.38	-0.91 ± 0.11	-0.19 ± 0.05	0.79 ± 0.14	1.53 ± 0.19	0.65
GFN-AL	3.83 ± 0.30	-4.57 ± 0.21	0.30 ± 0.02	-1.59 ± 0.11	-37.36 ± 1.49	-6.57 ± 0.34	1.17 ± 0.05	3.49 ± 0.11	-3.52
GGS	3.53 ± 0.03	-3.09 ± 0.19	0.28 ± 0.05	-1.52 ± 0.07	-8.32 ± 0.49	-3.12 ± 0.14	0.01 ± 0.00	1.74 ± 0.22	-1.31
MLDE	**9.33** ± 0.40	**1.99** ± 0.02	**2.04** ± 0.06	**10.55** ± 0.13	**0.03** ± 0.00	**0.04** ± 0.01	**1.27** ± 0.06	**4.01** ± 0.12	**3.67**

TABLE VI
EDIT DISTANCE BETWEEN THE WILD-TYPE SEQUENCE AND THE BEST-DESIGNED SEQUENCE.

Task	dist(WT)	Task	dist(WT)
avGFP	107.60 ± 2.94 (45.4%)	AMIE	124.40 ± 9.18 (36.5%)
AAV	5.00 ± 0.00 (17.9%)	LGK	149.40 ± 6.15 (34.0%)
TEM	70.80 ± 2.64 (24.8%)	Pab1	26.00 ± 1.27 (34.7%)
E4B	31.40 ± 2.42 (30.8%)	UBE2I	42.00 ± 4.69 (26.4%)

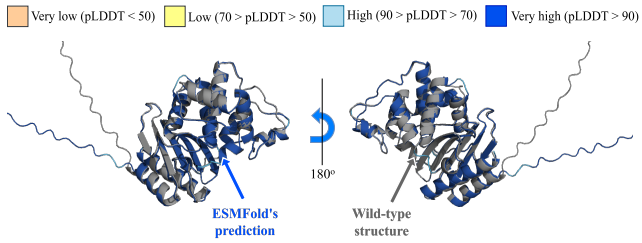

Fig. 5. 3D visualization of best-designed variant of TEM-1 β-Lactamase protein (colored), aligned with its wild-type structure (gray). The color of predicted structure represents ESMFold's local prediction confidence (pLDDT) per amino acid location.

predictions directly from the primary protein sequence. (2) It operates without reliance on multiple-sequence alignment (MSA), a crucial but computational-burden component in the architecture of many established folding models [83], [84], resulting in a 60x speedup for short sequences and a **10x speedup on average** compared to these models. (3) It predicts structures with confidence and does not depend on any templates, **mitigating the potential for producing an "unfolded" structure** for a sequence highly similar to a template. Figure 5 illustrates the tertiary structure of TEM's best variant, aligned with its wild-type structure. The root mean square deviation (RMSD) between the two structures is 0.324 $\AA$. Additionally, the figure demonstrates the high confidence of our predicted structure, validating our model's capability to design a real TEM-1 β-Lactamase protein. Structures of other proteins are presented in the Supplementary Material. We also explore the application of ESM-1v [85] in the zero-shot validation of generated sequences, which is introduced in the Supplementary Material.

Our implementation is publicly available at https://github.com/HySonLab/Directed_Evolution.

VI. CONCLUDING REMARKS

This paper introduces a Machine Learning-Driven Evolution (MLDE) framework. Incorporating two distinct masking strategies, namely random masking and importance masking, our pipeline utilizes pretrained models with minimal training, resulting in protein sequences exhibiting higher fitness. Experimental results across six protein sequence design tasks demonstrate that our method surpasses several robust baselines across nearly all metrics. In the future work, we aim to apply our Directed Evolution framework to the task of ligand-binding protein redesign, antibody design, etc. We hope our framework will enable scientists and researchers to efficiently optimize proteins with certain given properties.

Limitations: Despite the positive outcomes of our approach, the protein sequences we created have not been verified through *in-vitro* experimentation, which introduces a level of uncertainty associated with the oracle model. The results presented in our research are obtained by averaging data from five separate runs with different seeds, which can accommodate some variance on these metrics. Additionally, all fitness data utilized in our study are derived from wet-lab experiments, ensuring that the fitness values within the training datasets are realistic. Although there is currently no perfect metric available, we believe that new methods should be encouraged in the field of computational biology.

REFERENCES

[1] C. Hsu, H. Nisonoff, C. Fannjiang, and J. Listgarten, "Learning protein fitness models from evolutionary and assay-labeled data," *Nature Biotechnology*, vol. 40, no. 7, pp. 1114–1122, Jul 2022. [Online]. Available: https://doi.org/10.1038/s41587-021-01146-5

[2] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, p. e2016239118, 2021. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2016239118

[3] R. Verkuil, O. Kabeli, Y. Du, B. I. Wicky, L. F. Milles, J. Dauparas, D. Baker, S. Ovchinnikov, T. Sercu, and A. Rives, "Language models generalize beyond natural proteins," *bioRxiv*, pp. 2022–12, 2022.

[4] D. W. Anderson, F. Baier, G. Yang, and N. Tokuriki, "The adaptive landscape of a metallo-enzyme is shaped by environment-dependent epistasis," *Nature Communications*, vol. 12, no. 1, p. 3867, 2021.

[5] S. J. Remington, "Green fluorescent protein: a perspective," *Protein science*, vol. 20, no. 9, pp. 1509–1519, 2011.

[6] N. A. Pierce and E. Winfree, "Protein Design is NP-hard," *Protein Engineering, Design and Selection*, vol. 15, no. 10, pp. 779–782, 10 2002. [Online]. Available: https://doi.org/10.1093/protein/15.10.779

[7] J. M. SMITH, "Natural selection and the concept of a protein space," *Nature*, vol. 225, no. 5232, pp. 563–564, Feb. 1970. [Online]. Available: https://doi.org/10.1038/225563a0

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[11] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[12] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar *et al.*, "Voicebox: Text-guided multilingual universal speech generation at scale," *arXiv preprint arXiv:2306.15687*, 2023.

[13] Z. Wu, K. K. Yang, M. J. Liszka, A. Lee, A. Batzilla, D. Wernick, D. P. Weiner, and F. H. Arnold, "Signal peptides generated by attention-based neural networks," *ACS Synthetic Biology*, vol. 9, no. 8, pp. 2154–2161, 2020, pMID: 32649182. [Online]. Available: https://doi.org/10.1021/acssynbio.0c00219

[14] D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa, O. Savolainen, R. Meskys, M. K. M. Engqvist, and A. Zelezniak, "Expanding functional protein sequence spaces using generative adversarial networks," *Nature Machine Intelligence*, vol. 3, no. 4, pp. 324–333, Apr 2021. [Online]. Available: https://doi.org/10.1038/s42256-021-00310-5

[15] A. Hawkins-Hooker, F. Depardieu, S. Baur, G. Couairon, A. Chen, and D. Bikard, "Generating functional protein variants with variational autoencoders," *PLOS Computational Biology*, vol. 17, no. 2, p. e1008736, Feb. 2021. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1008736

[16] O. Kuchner and F. H. Arnold, "Directed evolution of enzyme catalysts," *Trends in biotechnology*, vol. 15, no. 12, pp. 523–530, 1997.

[17] F. H. Arnold, "Design by directed evolution," *Accounts of chemical research*, vol. 31, no. 3, pp. 125–131, 1998.

[18] C. A. Tracewell and F. H. Arnold, "Directed enzyme evolution: climbing fitness peaks one amino acid at a time," *Current opinion in chemical biology*, vol. 13, no. 1, pp. 3–9, 2009.

[19] P. Romero and F. Arnold, "Exploring protein fitness landscapes by directed evolution," *Nature reviews. Molecular cell biology*, vol. 10, pp. 866–76, 12 2009.

[20] P. A. Romero, A. Krause, and F. H. Arnold, "Navigating the protein fitness landscape with gaussian processes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 3, pp. E193–E201, 2013. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1215251110

[21] C. N. Bedbrook, K. K. Yang, J. E. Robinson, E. D. Mackey, V. Gradinaru, and F. H. Arnold, "Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics," *Nature Methods*, vol. 16, no. 11, pp. 1176–1184, Oct. 2019. [Online]. Available: https://doi.org/10.1038/s41592-019-0583-8

[22] Z. Wu, S. B. J. Kan, R. D. Lewis, B. J. Wittmann, and F. H. Arnold, "Machine learning-assisted directed protein evolution with combinatorial libraries," *Proceedings of the National Academy of Sciences*, vol. 116, no. 18, pp. 8852–8858, 2019. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1901979116

[23] P. Emami, A. Perreault, J. Law, D. Biagioni, and P. S. John, "Plug & play directed evolution of proteins with gradient-based discrete mcmc," *Machine Learning: Science and Technology*, vol. 4, no. 2, p. 025014, apr 2023. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/accacd

[24] Z. Ren, J. Li, F. Ding, Y. Zhou, J. Ma, and J. Peng, "Proximal exploration for model-guided protein sequence design," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 18 520–18 536. [Online]. Available: https://proceedings.mlr.press/v162/ren22a.html

[25] Y. Qiu, J. Hu, and G.-W. Wei, "Cluster learning-assisted directed evolution," *Nature Computational Science*, vol. 1, no. 12, pp. 809–818, Dec 2021. [Online]. Available: https://doi.org/10.1038/s43588-021-00168-y

[26] Y. Qiu and G.-W. Wei, "Clade 2.0: Evolution-driven cluster learning-assisted directed evolution," *Journal of Chemical Information and Modeling*, vol. 62, no. 19, pp. 4629–4641, 2022, pMID: 36154171. [Online]. Available: https://doi.org/10.1021/acs.jcim.2c01046

[27] B. J. Wittmann, Y. Yue, and F. H. Arnold, "Informed training set design enables efficient machine learning-assisted directed protein evolution," *Cell Systems*, vol. 12, no. 11, pp. 1026–1045.e7, Nov. 2021. [Online]. Available: https://doi.org/10.1016/j.cels.2021.07.008

[28] D. C. E. Koo and R. Bonneau, "Towards region-specific propagation of protein functions," *Bioinformatics*, vol. 35, no. 10, pp. 1737–1744, 10 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty834

[29] W. Torng and R. B. Altman, "High precision protein functional site detection using 3D convolutional neural networks," *Bioinformatics*, vol. 35, no. 9, pp. 1503–1512, 09 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty813

[30] Y. Guan, C. L. Myers, D. C. Hess, Z. Barutcuoglu, A. A. Caudy, and O. G. Troyanskaya, "Predicting gene function in a hierarchical context with an ensemble of classifiers," *Genome Biology*, vol. 9, no. Suppl 1, p. S3, 2008. [Online]. Available: https://doi.org/10.1186/gb-2008-9-s1-s3

[31] M. N. Wass, G. Barton, and M. J. E. Sternberg, "CombFunc: predicting protein function using heterogeneous data sources," *Nucleic Acids Research*, vol. 40, no. W1, pp. W466–W470, 05 2012. [Online]. Available: https://doi.org/10.1093/nar/gks489

[32] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 10 2017. [Online]. Available: https://doi.org/10.1093/bioinformatics/btx624

[33] M. Kulmanov and R. Hoehndorf, "DeepGOPlus: improved protein function prediction from sequence," *Bioinformatics*, vol. 36, no. 2, pp. 422–429, 07 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz595

[34] V. Gligorijević, P. D. Renfrew, T. Kosciolek, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, R. J. Xavier, R. Knight, K. Cho, and R. Bonneau, "Structure-based protein function prediction using graph convolutional networks," *Nature Communications*, vol. 12, no. 1, May 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-23303-9

[35] Z. Wang, S. A. Combs, R. Brand, M. R. Calvo, P. Xu, G. Price, N. Golovach, E. O. Salawu, C. J. Wise, S. P. Ponnapalli, and P. M. Clark, "Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction," *Scientific Reports*, vol. 12, no. 1, p. 6832, Apr 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-10775-y

[36] K. Ngo and T. S. Hy, "Target-aware variational auto-encoders for ligand generation with multi-modal protein modeling," in *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. [Online]. Available: https://openreview.net/forum?id=4k926QVVM4

[37] N. K. Ngo, T. S. Hy, and R. Kondor, "Multiresolution graph transformers and wavelet positional encoding for learning long-range and hierarchical structures," *The Journal of Chemical Physics*, vol. 159, no. 3, p. 034109, 07 2023. [Online]. Available: https://doi.org/10.1063/5.0152833

[38] Y. Gumulya, J.-M. Baek, S.-J. Wun, R. E. S. Thomson, K. L. Harris, D. J. B. Hunter, J. B. Y. H. Behrendorff, J. Kulig, S. Zheng, X. Wu, B. Wu, J. E. Stok, J. J. D. Voss, G. Schenk, U. Jurva, S. Andersson, E. M. Isin, M. Bodén, L. Guddat, and E. M. J. Gillam, "Engineering highly functional thermostable proteins using ancestral sequence reconstruction," *Nature Catalysis*, vol. 1, no. 11, pp. 878–888, Oct. 2018. [Online]. Available: https://doi.org/10.1038/s41929-018-0159-5

[39] C. Angermueller, D. Dohan, D. Belanger, R. Deshpande, K. Murphy, and L. Colwell, "Model-based reinforcement learning for biological sequence design," in *International conference on learning representations*, 2019.

[40] M. Jain, E. Bengio, A. Hernandez-Garcia, J. Rector-Brooks, B. F. P. Dossou, C. A. Ekbote, J. Fu, T. Zhang, M. Kilgour, D. Zhang, L. Simine, P. Das, and Y. Bengio, "Biological sequence design with GFlowNets," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 9786–9801. [Online]. Available: https://proceedings.mlr.press/v162/jain22a.html

[41] D. Belanger, S. Vora, Z. Mariet, R. Deshpande, D. Dohan, C. Angermueller, K. Murphy, O. Chapelle, and L. Colwell, "Biological sequences design using batched bayesian optimization," 2019.

[42] H. Moss, D. Leslie, D. Beck, J. González, and P. Rayson, "Boss: Bayesian optimization over string spaces," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 476–15 486. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/b19aa25ff58940d974234b48391b9549-Paper.pdf

[43] K. Terayama, M. Sumita, R. Tamura, and K. Tsuda, "Black-box optimization for automated discovery," *Accounts of Chemical Research*, vol. 54, no. 6, pp. 1334–1346, 2021, pMID: 33635621. [Online]. Available: https://doi.org/10.1021/acs.accounts.0c00713

[44] Z. Song and L. Li, "Importance weighted expectation-maximization for protein sequence design," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 32 349–32 364. [Online]. Available: https://proceedings.mlr.press/v202/song23g.html

[45] K. Swersky, Y. Rubanova, D. Dohan, and K. Murphy, "Amortized bayesian optimization over discrete spaces," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, ser. Proceedings of Machine Learning Research, J. Peters and D. Sontag, Eds., vol. 124. PMLR, 03–06 Aug 2020, pp. 769–778. [Online]. Available: https://proceedings.mlr.press/v124/swersky20a.html

[46] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost, "ProtTrans: Toward understanding the language of life through self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022. [Online]. Available: https://doi.org/10.1109/tpami.2021.3095381

[47] N. Ferruz, S. Schmidt, and B. Höcker, "Protgpt2 is a deep unsupervised language model for protein design," *Nature Communications*, vol. 13, no. 1, p. 4348, Jul 2022. [Online]. Available: https://doi.org/10.1038/s41467-022-32007-7

[48] E. Nijkamp, J. Ruffolo, E. N. Weinstein, N. Naik, and A. Madani, "Progen2: exploring the boundaries of protein language models," *arXiv preprint arXiv:2206.13517*, 2022.

[49] A. J. Riesselman, J. B. Ingraham, and D. S. Marks, "Deep generative models of genetic variation capture the effects of mutations," *Nature Methods*, vol. 15, no. 10, pp. 816–822, Oct 2018. [Online]. Available: https://doi.org/10.1038/s41592-018-0138-4

[50] X. Ding, Z. Zou, and C. L. Brooks III, "Deciphering protein evolution and fitness landscapes with latent space models," *Nature Communications*, vol. 10, no. 1, p. 5644, Dec 2019. [Online]. Available: https://doi.org/10.1038/s41467-019-13633-0

[51] A. Gupta and J. Zou, "Feedback gan for dna optimizes protein functions," *Nature Machine Intelligence*, vol. 1, no. 2, pp. 105–111, Feb 2019. [Online]. Available: https://doi.org/10.1038/s42256-019-0017-4

[52] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker, "De novo design of protein structure and function with rfdiffusion," *Nature*, vol. 620, no. 7976, pp. 1089–1100, Aug 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-06415-8

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[55] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, 2023. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.ade2574

[56] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," 2022.

[57] H. Derbel, Z. Zhao, and Q. Liu, "Accurate prediction of functional effect of single amino acid variants with deep learning," *Computational and Structural Biotechnology Journal*, vol. 21, p. 5776–5784, 2023. [Online]. Available: http://dx.doi.org/10.1016/j.csbj.2023.11.017

[58] W. Lin, J. Wells, Z. Wang, C. Orengo, and A. C. Martin, "Varipred: Enhancing pathogenicity prediction of missense variants using protein language models," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/03/20/2023.03.16.532942.1

[59] S. Sledzieski, M. Kshirsagar, M. Baek, B. Berger, R. Dodhia, and J. L. Ferres, "Democratizing protein language models with parameter-efficient fine-tuning," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/11/10/2023.11.09.566187

[60] V. T. D. Nguyen and T. S. Hy, "Multimodal pretraining for unsupervised protein representation learning," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/12/07/2023.11.29.569288

[61] T. Chen, P. Vure, R. Pulugurta, and P. Chatterjee, "AMP-diffusion: Integrating latent diffusion with protein language models for antimicrobial peptide generation," in *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. [Online]. Available: https://openreview.net/forum?id=145TM9VQhx

[62] T. Cohen and D. Schneidman-Duhovny, "Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings," in *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. [Online]. Available: https://openreview.net/forum?id=Enqxq6TWoZ

[63] J. Chen, A. Zhang, M. Li, A. Smola, and D. Yang, "A cheaper and better diffusion language model with soft-masked noise," 2023.

[64] D. Dessì, R. Helaoui, V. Kumar, D. R. Recupero, and D. Riboni, "TF-IDF vs word embeddings for morbidity identification in clinical notes: An initial study," in *Proceedings of the First Workshop on Smart Personal Health Interfaces co-located with 25th International Conference on Intelligent User Interfaces, SmartPhil@IUI 2020, Cagliari, Italy, March 17, 2020*, ser. CEUR Workshop Proceedings, S. Consoli, D. R. Recupero, and D. Riboni, Eds., vol. 2596. CEUR-WS.org, 2020, pp. 1–12. [Online]. Available: http://ceur-ws.org/Vol-2596/paper1.pdf

[65] C. Bentz and D. Alikaniotis, "The word entropy of natural languages," 2016.

[66] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez, N. S. Bogatyreva, P. K. Vlasov, E. S. Egorov, M. D. Logacheva, A. S. Kondrashov, D. M. Chudakov, E. V. Putintseva, I. Z. Mamedov, D. S. Tawfik, K. A. Lukyanov, and F. A. Kondrashov, "Local fitness landscape of the green fluorescent protein," *Nature*, vol. 533, no. 7603, pp. 397–401, May 2016. [Online]. Available: https://doi.org/10.1038/nature17995

[67] D. H. Bryant, A. Bashir, S. Sinai, N. K. Jain, P. J. Ogden, P. F. Riley, G. M. Church, L. J. Colwell, and E. D. Kelsic, "Deep diversification of an aav capsid protein by machine learning," *Nature Biotechnology*, vol. 39, no. 6, pp. 691–696, Jun 2021. [Online]. Available: https://doi.org/10.1038/s41587-020-00793-4

[68] E. Firnberg, J. W. Labonte, J. J. Gray, and M. Ostermeier, "A Comprehensive, High-Resolution Map of a Gene's Fitness Landscape," *Molecular Biology and Evolution*, vol. 31, no. 6, pp. 1581–1592, 02 2014. [Online]. Available: https://doi.org/10.1093/molbev/msu081

[69] L. M. Starita, J. N. Pruneda, R. S. Lo, D. M. Fowler, H. J. Kim, J. B. Hiatt, J. Shendure, P. S. Brzovic, S. Fields, and R. E. Klevit, "Activity-enhancing mutations in an e3 ubiquitin ligase identified by high-throughput mutagenesis," *Proceedings of the National Academy of Sciences*, vol. 110, no. 14, pp. E1263–E1272, 2013. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1303309110

[70] E. E. Wrenbeck, L. R. Azouz, and T. A. Whitehead, "Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded," *Nature Communications*, vol. 8, no. 1, p. 15695, Jun 2017. [Online]. Available: https://doi.org/10.1038/ncomms15695

[71] J. R. Klesmith, J.-P. Bacik, R. Michalczyk, and T. A. Whitehead, "Comprehensive sequence-flux mapping of a levoglucosan utilization pathway in e. coli," *ACS Synthetic Biology*, vol. 4, no. 11, p. 1235–1243, Sep. 2015. [Online]. Available: http://dx.doi.org/10.1021/acssynbio.5b00131

[72] D. Melamed, D. L. Young, C. E. Gamble, C. R. Miller, and S. Fields, "Deep mutational scanning of an rrm domain of the saccharomyces cerevisiae poly(a)-binding protein," *RNA*, vol. 19, no. 11, p. 1537–1551, Sep. 2013. [Online]. Available: http://dx.doi.org/10.1261/rna.040709.113

[73] J. Weile, S. Sun, A. G. Cote, J. Knapp, M. Verby, J. C. Mellor, Y. Wu, C. Pons, C. Wong, N. van Lieshout, F. Yang, M. Tasan,

G. Tan, S. Yang, D. M. Fowler, R. Nussbaum, J. D. Bloom, M. Vidal, D. E. Hill, P. Aloy, and F. P. Roth, "A framework for exhaustively mapping functional missense variants," *Molecular Systems Biology*, vol. 13, no. 12, p. 957, 2017. [Online]. Available: https://www.embopress.org/doi/abs/10.15252/msb.20177908

[74] A. Graves, "Sequence transduction with recurrent neural networks," 2012.

[75] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks." in *ISMIR*. Curitiba, 2013, pp. 335–340.

[76] S. Sinai, R. Wang, A. Whatley, S. Slocum, E. Locane, and E. Kelsic, "Adalead: A simple and robust adaptive greedy search algorithm for sequence design," *arXiv preprint*, 2020.

[77] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[78] D. H. Brookes and J. Listgarten, "Design by adaptive sampling," 2020.

[79] D. Brookes, H. Park, and J. Listgarten, "Conditioning by adaptive sampling for robust design," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 773–782. [Online]. Available: https://proceedings.mlr.press/v97/brookes19a.html

[80] B. Trabucco, A. Kumar, X. Geng, and S. Levine, "Conservative objective models for effective offline model-based optimization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 358–10 368.

[81] A. Kirjner, J. Yim, R. Samusevich, S. Bracha, T. S. Jaakkola, R. Barzilay, and I. R. Fiete, "Improving protein optimization with smoothed fitness landscapes," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=rxlF2Zv8x0

[82] L. Gonzalez Somermeyer, A. Fleiss, A. S. Mishin, N. G. Bozhanova, A. A. Igolkina, J. Meiler, M.-E. Alaball Pujol, E. V. Putintseva, K. S. Sarkisyan, and F. A. Kondrashov, "Heterogeneity of the gfp fitness landscape and data-driven protein design," *eLife*, vol. 11, p. e75842, may 2022. [Online]. Available: https://doi.org/10.7554/eLife.75842

[83] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03819-2

[84] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read, and D. Baker, "Accurate prediction of protein structures and interactions using a three-track neural network," *Science*, vol. 373, no. 6557, pp. 871–876, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.abj8754

[85] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives, "Language models enable zero-shot prediction of the effects of mutations on protein function," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 29 287–29 303. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/f51338d736f95dd42427296047067694-Paper.pdf

Thanh V. T. Tran is currently an AI resident working in the group of Dr. Truong Son Hy at the FPT Software AI Center, Hanoi, Vietnam. He earned his BSc degree in Computer Science from the University of Engineering and Technology, Vietnam National University. His research focuses on generative models for scientific applications in the direction of **AI for Science and Engineering**.

Dr. Hy Truong Son is currently a tenure-track Assistant Professor in the Department of Computer Science, University of Alabama at Birmingham, United States. He has earned his PhD in Computer Science from University of Chicago, and his BSc in Computer Science from Eötvös Loránd University. Prior to his current faculty position, he has worked as a lecturer and postdoctoral fellow in the Halıcıoğlu Data Science Institute at University of California, San Diego; and a tenure-track Assistant Professor in the Department of Mathematics and Computer Science at Indiana State University. His research focuses on graph neural networks, multiresolution matrix factorization, graph wavelets, deep generative models on graphs, group equivariant, and multiscale hierarchical models for scientific applications in the direction of **AI for Science and Engineering**.