Knowledge-aware Reinforced Language Models for Protein Directed Evolution

Yuhao Wang [*1 2] **Qiang Zhang** [*1 2] **Ming Qin** [1 2] **Xiang Zhuang** [1 2] **Xiaotong Li** [1 2] **Zhichen Gong** [2] **Zeyuan Wang** [1 2] **Yu Zhao** [3] **Jianhua Yao** [3] **Keyan Ding** [1 2] **Huajun Chen** [1 2]

Abstract

Directed evolution, a cornerstone of protein optimization, is to harness natural mutational processes to enhance protein functionality. Existing Machine Learning-assisted Directed Evolution (MLDE) methodologies typically rely on data-driven strategies and often overlook the profound domain knowledge in biochemical fields. In this paper, we introduce a novel Knowledge-aware Reinforced Language Model (KnowRLM) for MLDE. An Amino Acid Knowledge Graph (AAKG) is constructed to represent the intricate biochemical relationships among amino acids. We further propose a Protein Language Model (PLM)-based policy network that iteratively samples mutants through preferential random walks on the AAKG using a dynamic sliding window mechanism. The novel mutants are actively sampled to fine-tune a fitness predictor as the reward model, providing feedback to the knowledge-aware policy. Finally, we optimize the whole system in an active learning approach that mimics biological settings in practice. KnowRLM stands out for its ability to utilize contextual amino acid information from knowledge graphs, thus attaining advantages from both statistical patterns of protein sequences and biochemical properties of amino acids. Extensive experiments demonstrate the superior performance of KnowRLM in more efficiently identifying high-fitness mutants compared to existing methods.

[*]Equal contribution [1]Zhejiang University [2]ZJU-Hangzhou Global Scientific and Technological Innovation Center [3]Tencent AI Lab. Correspondence to: Qiang Zhang <qiang.zhang.cs@zju.edu.cn>, Keyan Ding <dingkeyan@zju.edu.cn>, Huajun Chen <huajunsir@zju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Proteins play a pivotal role in numerous biological processes, and their study is fundamental to advancing our understanding of life sciences. Deep learning, a subset of machine learning, has been extensively applied in the field of protein research, including function annotation (Xu et al., 2021), 3D structure prediction (Jumper et al., 2021), amino acid sequence generation, and protein engineering (Anishchenko et al., 2021). The incorporation of deep learning techniques has substantially improved both the accuracy and efficiency of protein research.

Directed evolution (Arnold, 1998; Smith & Petrenko, 1997; Winter et al., 1994), a commonly used technique in protein engineering, aims to refine and optimize the protein functions of interest by mimicking the process of natural selection (Packer & Liu, 2015). The essence of this technique lies in screening and optimizing protein sequences to achieve improved biological performance, such as catalytic efficiency (Toyao et al., 2019), thermal stability (Tian et al., 2010), or drug affinity (Hie et al., 2020). Conventionally, researchers rely on random or unsupervised sampling methods to generate a range of protein mutants and then select those with desired traits (Wu et al., 2019). This approach is often labor-intensive and time-consuming, with a key challenge of effective exploration of the complex and astronomical protein sequence space to identify optimal mutants.

The rapid development of machine learning technologies has provided new momentum for protein directed evolution. Machine learning's capacity to process and analyze vast quantities of biological data is particularly beneficial for unraveling complex protein sequence-function relationships. For example, researchers use clustering to find informative mutants (Qiu et al., 2021). More recently, reinforcement learning (RL) has been leveraged in the realm of directed evolution. It enables effective navigation in the protein sequence space, optimizing search strategies to discover mutations that yield the desired performance. A prime example of this application is EvoPlay (Wang et al., 2023), which draws inspirations from the AlphaZero self-play reinforcement learning framework. This method enhances search efficiency while simultaneously reducing experimental workload, showcasing RL's potential in protein design.

However, existing methods focus on algorithmic optimization to achieve specific biological functions, and rely on a data-driven trial-and-error mechanism, which may be inefficient to capture all necessary biochemical details in complex protein research tasks. For example, data-driven amino acid mutations might disregard critical aspects like chemical properties, size, and charge, all of which can profoundly influence protein stability and functionality. For protein evolution, understanding amino acid biochemical properties is crucial for predicting how proteins respond to specific mutations. Neglecting these intricate biological nuances in the learning process might fail to achieve anticipated biological activity or show instability in practical applications.

In this paper, we propose a Knowledge-aware Reinforced Language Model (**KnowRLM**) for protein directed evolution, which encodes complex biochemical rules and relationships directly into the decision process in reinforcement learning. First of all, we build an Amino Acid Knowledge Graph (AAKG), describing the critical biochemical features of amino acids and their intrinsic connections. Then we propose a knowledge-aware policy network based on protein language models to predict mutation sites and types through preferential random walks on the AAKG. In particular, we introduce a dynamic sliding window mechanism into the walk strategy, adjusting the scope of amino acid exploration adaptively. We implement the reward model as a fitness predictor of mutants to provide pesudo rewards for policy optimization. Moreover, the reward model is iteratively fine-tuned in an active learning manner to improve its predictive capability. The last reward model will be employed as the ultimate fitness predictor to sample the optimal mutants. The contributions of this paper are summarized as follows:

- We construct the first amino acid knowledge graph that maintains domain knowledge and models structured connections between amino acids.
- We introduce a knowledge-aware policy that considers the physicochemical properties of amino acids during the exploration phase in reinforcement learning.
- We demonstrate the critical role of domain knowledge in enhancing MLDE efficiency, offering new insights for protein directed evolution.

2. Related Works

Machine Learning-assisted Directed Evolution (MLDE). MLDE represents a novel strategy in protein engineering, facilitating the computational screening of all mutant sequences. This process hinges on iteratively optimizing prediction and sampling phases: the former learning from labeled data to map the fitness landscape of sequences, and the latter leveraging model predictions to guide sequence selection in experimental iterations (Hie & Yang, 2022). The ftMLDE (Wittmann et al., 2021) approach demonstrated its effectiveness by employing an enriched training set, highlighting the significance of selecting highly informative data for navigating the protein fitness landscape. The CLADE (Qiu et al., 2021) method combined unsupervised hierarchical clustering with supervised learning to efficiently explore the mutational space. CLADE2.0(Qiu & Wei, 2022) was further refined through the incorporation of multiple evolutionary scoring metrics and evolution-driven cluster sampling. The TS-DE strategy(Yu et al., 2018) utilized Thompson sampling coupled with a simplistic Bayesian linear model. AFP-DE (Qin et al., 2023) harnessed PLMs to facilitate active sampling and fine-tuning, progressively augmenting both the sampling efficacy and the model performance throughout iterative cycles.

Protein Language Models (PLMs). Protein language models (Devlin et al., 2018; Vig et al., 2020) have made significant strides in bioinformatics and computational biology. These models, drawing parallels with natural language processing (NLP) techniques, parse and predict protein sequence characteristics, offering new insights into protein structure and function understanding. A Transformer model (Rao et al., 2021) was proposed to capture advanced structural properties of proteins, including the spatial proximity of amino acids in 3D structures and functional regions like binding sites. The ESM model (Verkuil et al., 2022), an advanced protein language model, was designed to understand and predict protein structure and function by capturing evolutionary relationships among protein sequences. ESM-1v (Meier et al., 2021) learns intrinsic patterns and rules within protein sequences, showcasing the application potential of PLMs in biological research. LM-GVP (Wang et al., 2022a) employs Transformer blocks alongside a graph network extracted from the three-dimensional structure of proteins. PromptProtein (Wang et al., 2022b) leverages prompt-guided multi-task pretraining to amalgamate diverse aspects of protein structure at various levels. SaProt (Su et al., 2023)introduces an innovative vocabulary that seamlessly combines residue tokens with structural tokens.

Reinforcement Learning (RL). Reinforcement learning has been successfully implemented in diverse fields to enhance existing design methodologies, as indicated in studies (Schaff et al., 2019; Yu et al., 2018). EvoPlay (Wang et al., 2023) introduced a self-play reinforcement learning framework based on AlphaZero, utilizing simulated single-point mutations and a synergy of policy-value neural networks with Monte Carlo tree search to guide the optimization of protein sequences. ChemRLformer (Ghugare et al., 2023) devised an RL-based molecular design algorithm to identify high-value small molecules within an expansive search space. TCRPPO (Chen et al., 2023) developed a Proximal Policy Optimization (PPO) based approach (Schulman et al., 2017) to tailor T-Cell Receptor (TCR) optimization

for any given peptide. However, RL methods that overlook domain knowledge may deviate during exploration, with certain amino acid substitutions seemingly beneficial for functionality but potentially detrimental to protein stability. To mitigate this, we propose the integration of expert knowledge into RL to efficiently yield high-fitness mutants.

3. Task Formulation and Preliminary

3.1. Formulation of Directed Evolution

A protein can be conceptualized as a sequence of amino acid tokens, represented as $X = [x_1, \ldots, x_N]$ within the protein space $\mathbb{X}$. Here, each x denotes one of the twenty distinct amino acids in nature, and N denotes the length of the protein sequence. The objective of directed evolution is to ascertain the optimal amino acid sequence X^* that exhibits the pinnacle of biological fitness within the expansive protein space. Mathematically, this is formulated as:

$$X^* = \arg\max_{X \in \mathbb{X}} \mathcal{F}(X), \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the sequence-to-fitness prediction function. For a target protein sequence X, the mutation space at n distinct sites (with $n < N$) proliferates to encompass 20^n unique sequences, each a potential candidate for the optimal sequence. The crux of directed evolution lies in efficiently traversing this vast mutation space to pinpoint the global optimum. This is conventionally achieved through sequential queries in biological fitness experiments. However, the substantial financial and resource implications of these experiments necessitate an economical query strategy. From a mathematical standpoint, this translates to the learning of the function $\mathcal{F}(\cdot)$ utilizing the smallest possible set of annotated data points.

3.2. Reinforcement Learning-based Directed Evolution

Preliminary of Reinforcement Learning. Reinforcement Learning tackles sequential decision-making problems by learning from interaction and feedback sequences. Generally, the problem to be solved is described by a Markov Decision Process (MDP), which is characterized by a tuple $\langle \mathbb{S}, \mathbb{A}, \mathbb{E}, r, \gamma \rangle$. Here, $\mathbb{S}$ represents the state space and $\mathbb{A}$ symbolizes the action space. The function $\mathbb{E} : \mathbb{S} \times \mathbb{A} \times \mathbb{S}' \to [0, 1]$, acts as the transition function. The function $r : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$, is the reward function, and γ denotes the discount rate. The goal of RL is to develop a solution, denoted as a policy $\pi_\theta(a_t|s_t)$, that effectively maps states to actions, i.e., making the right decision in each state. The optimization objective is:

$$\theta^* = \arg\max_{\theta} E_{\tau \sim p(\tau|\theta)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right], \tag{2}$$

where θ^* denotes the optimal paramatrisation of the policy, and $s_t \in \mathbb{S}, a_t \in \mathbb{A}$. The probability distribution over trajectories, denoted as $p(\tau|\theta)$, is induced by the policy π_θ and the transition function $\mathbb{E}$.

Directed Evolution as Markov Decision Process. We model the directed evolution process of a protein sequence as a sequence of mutation decisions.

State Space Each state $s_t \in S$ is a mutant sequence of the protein. A protein sequence is delineated as an ordered array of its constituent amino acids, which can be represented as $s_t = [x_1^t, x_2^t, \ldots, x_N^t]$ at step t (where $t = 0, \ldots, T$). The initial state s_0 in this context is defined as the wild-type protein. A state s_t is terminal (denoted as s_T) if it reaches the maximum step limit T. In this formulation, the state s can be exchangeable with the protein sequence X in Eq.(1).

Action Space The action consists of two parts, i.e., $a_t = (\hat{p}_t, \hat{x}_t)$, where $\hat{p}_t$ is the position to mutate and $\hat{x}_t$ is an amino acid candidate. The policy first needs to decide where to mutate, i.e., $\hat{p}_t$, then decide what to mutate, i.e., $\hat{x}_t$. $E(s_{t+1}|s_t, a_t)$ denotes the probability of transitioning to the next state s_{t+1} at timestep $t + 1$ from state s_t with action a_t. In our paper, the transition refers to mutation from one protein sequence to another.

Reward Function The reward function quantifies the fitness score for each instance of protein mutation. More specifically, the reward value $r(s_t, a_t)$ is derived following the execution of action a_t (representing a specific mutation) in state s_t, leading to the new state s_{t+1}. Here, we let $r(s_t, a_t) = \mathcal{F}(s_{t+1})$, where the function $\mathcal{F}$ maps the extent of this change to the corresponding reward value. This mapping function plays a pivotal role in quantitatively assessing the impact of each mutational action within our reinforcement learning framework.

The RL strategy aims to maximize the expected accumulated scalar rewards over the course of the entire mutation process. This sequential approach allows for a comprehensive evaluation of the mutation process, focusing on the cumulative impact of successive actions rather than isolated end results. Considering Eq.(1) and Eq.(2), it is discernible that the objective of identifying an optimal solution in reinforcement learning aligns congruently with the aim of ascertaining the optimal protein sequence in directed evolution. This alignment permits the conceptualization of the objective function $\mathcal{F}(X)$ in directed evolution as analogous to the reward function $r(s_t, a_t)$ in the context of reinforcement learning. Within this framework, each potential solution X is analogous to executing an action a_t in a given state s_t. Consequently, the endeavor to discover the optimal solution in directed evolution is akin to the pursuit of an optimal policy in reinforcement learning, thereby facilitating the

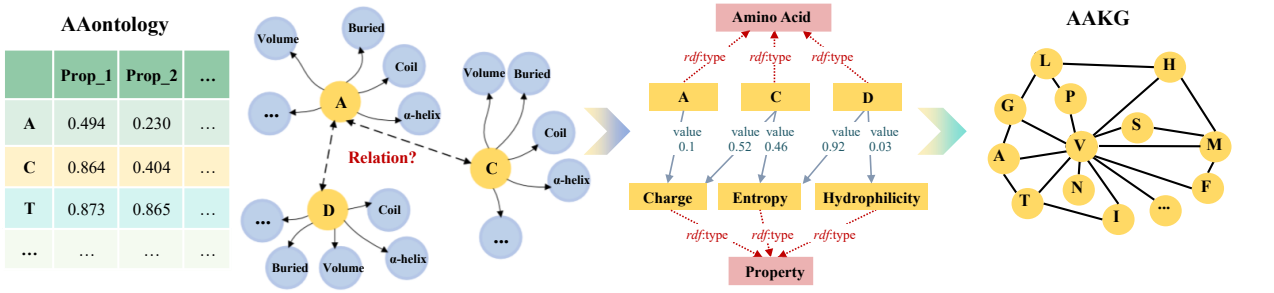


Figure 1. Illustration of the amino acid knowledge graph (AAKG). We construct AAKG by leveraging AAontology (Breimann et al., 2023), which encompasses a multitude of properties, enabling us to infer intricate relationships among amino acids.

establishment of the following equivalence relation:

$$\min \ell(\mathcal{F}(X)) \simeq \arg\max_{\theta} E_{\tau \sim p(\tau|\theta)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right], \tag{3}$$

where $\ell(\cdot)$ is the loss function to optimize the Eq.(1).

To ensure comprehensive understanding and facilitate ease of reference throughout this paper, we include a detailed table of symbols and their definitions in Appendix A.

4. Methodology

This section introduces KnowRLM for directed evolution, encompassing the following processes. First, we construct a knowledge graph of amino acids based on their properties, capturing intricate and interconnected relationships among amino acids. Building upon this, we then propose a Knowledge-Aware Policy to predict mutated sites and types through preferential random walks on the AAKG. Last, the reward model (i.e., a fitness predictor of mutants) provides feedback to KAP. We optimize KnowRLM in an active learning manner, with the identified mutants being annotated by an oracle and used to finetune the fitness predictor.

4.1. Amino Acid Knowledge Graph

Domain knowledge is crucial for protein analysis. Existing knowledge sources either do not contain amino acid-level information, such as ProteinKG25 (Zhang et al., 2022), or lack structured relations, such as AAontology (Breimann et al., 2023). To fill this gap, we construct an amino acid-centric knowledge graph (AAKG). Figure 1 illustrates the construction process of AAKG.

Specifically, based on AAontology, we identify various properties for each amino acid to construct AAKG, comprising two levels: instance and class, colored yellow and red respectively. At the class level, we have delineated amino acid classes. To forge intra-amino acid connections, we elected to also model the property at the class level. At the instance level, the 20 amino acids are instantiated as entities of the amino acid class, while various physicochemical properties of amino acids, such as polarity and volume, are instantiated as entities of the property class. Different amino acid entities can establish indirect relations through property entities. Entities are assigned to their respective classes through *rdf*:type, indicated by red dotted arrows. Furthermore, as illustrated by the blue arrows, we establish inter-entity relationships through object properties, signifying the specific numerical values pertaining to the amino acids' properties.

With properties as intermediates, we establish connections between amino acid entities. To do so, we measure the physicochemical similarity between amino acid entities based on the absolute average difference in all properties, and use the ranking of similarity as their edge weight $d(\cdot)$ in AAKG:

$$d(x, x') = \sum_{z=1}^{Z} |K_z(x) - K_z(x')|, \tag{4}$$

where Z represents the number of amino acid properties, K represents the value of a property. The values of Z and $K_z(\cdot)$ are determined according to AAOntology. This provides a detailed and structured representation of amino acid properties in the knowledge graph, depicting intricate connections among amino acids.

4.2. Knowledge-Aware Policy

The knowledge-aware policy aims to sample the optimal mutants with the highest fitness, which is achieved by predicting mutation sites and mutated amino acid types using PLMs and AAKG, as shown in Figure 2.

Mutation Site Prediction. Given a wild-type protein sequence, similar to EvoPlay, at each timestep t, we conduct single-site mutation. We begin by employing a PLM followed by a multi-layer perception (MLP) to predict the most likely mutation site of the n candidate sites:

$$\hat{p}_t = \text{MLP}(\mathcal{P}(s_t)), \tag{5}$$

where $\mathcal{P}$ denotes the PLM for protein embedding. Note that these predictions are not traditional category labels but

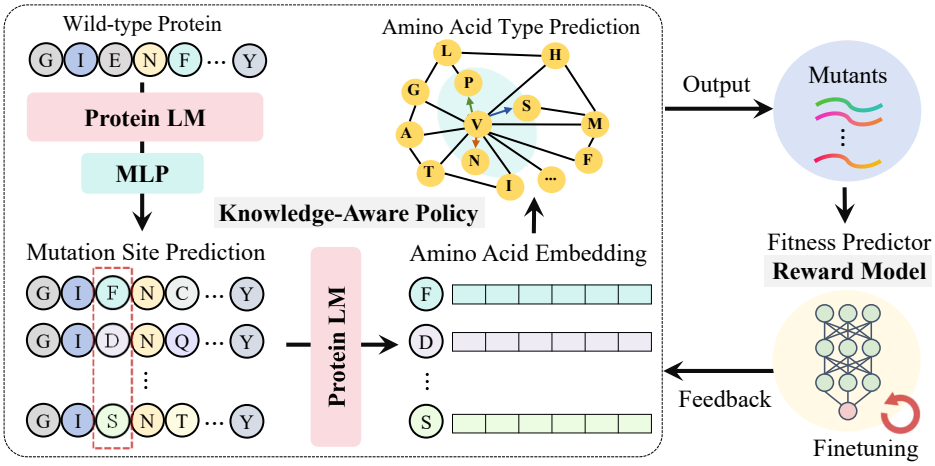


Figure 2. The proposed KnowRLM method consists of two main components: a knowledge-aware policy network and a reward model. Starting from a wild-type protein sequence, we first employ a protein language model (PLM) followed by a multi-layer perception (MLP) to predict the mutation site. The predicted site is then masked and suggested with other amino acids. Subsequently, the suggested amino acids are fed into the PLM to obtain their embeddings, which are utilized for performing preferential random walks on the AAKG to sample potentially optimal mutants. Finally, these mutants are evaluated using a reward model to calculate their fitness values and provide feedback to the knowledge-aware policy network. The above procedure is iteratively conducted in an active learning manner.

rather form a part of the policy. Once the mutation site is identified, the original amino acid at the position $\hat{p}_t$ is masked, leading to a temporary state:

$$s_{t+1}^{\text{MASK}} = [x_1, \ldots, \texttt{[MASK]}_{\hat{p}_t}, \ldots, x_N].$$

Amino Acid Type Prediction. After determining the mutation site $\hat{p}_t$, we then consider the appropriate amino acid mutant. The process of amino acid mutation is conceptualized as navigating from one amino acid node to another on the AAKG.

Specifically, to align the statistical regularities in the PLM with the physicochemical properties in the AAKG, we utilize the position-sensitive amino acid embeddings from the PLM as node embeddings in the AAKG. That is, $\texttt{[MASK]}_{\hat{p}_t}$ is substituted with each of the 20 amino acids, denoted as x^i with $i \in \{1, \cdots, 20\}$, converting s_{t+1}^{MASK} into s_{t+1}^i:

$$s_{t+1}^i = \left[x_1, \ldots, x^i, \ldots, x_N\right], \tag{6}$$

which is then fed into the PLM model to generate the embedding of amino acid x^i at position $\hat{p}_t$: $h_{x^i} = \mathcal{P}(s_{t+1}^i)[\hat{p}_t]$.

Identifying the mutant type of amino acids is achieved by navigating from one amino acid node to another in AAKG. The conventional random walk algorithm capable of path finding neglects the prior knowledge information (Pearson, 1905). Hence, we introduce a preferential random walk strategy. To measure the transition probability from one node to an adjacent one, we employ the cosine similarity between the embeddings of the two nodes. Our strategy is to find the new replacement within the neighbourhood of the amino acid x^i specified by the AAKG. Mathematically, the preferential random walk can be expressed as:

$$\hat{x}_t = \underset{\{x^j | d(x^i, x^j) < \mu\}}{\arg\max} \cos(h_{x^i}, h_{x^j}), \tag{7}$$

where $d(\cdot)$ is defined as Eq.(4), μ represents a hyperparameter threshold. By integrating the PLM with AAKG in this manner, we can empower the policy network to make well-informed action $a_t = (\hat{p}_t, \hat{x}_t)$, leading to the state

$$s_{t+1} = [x_1, \ldots, \hat{x}_t, \ldots, x_N] \tag{8}$$

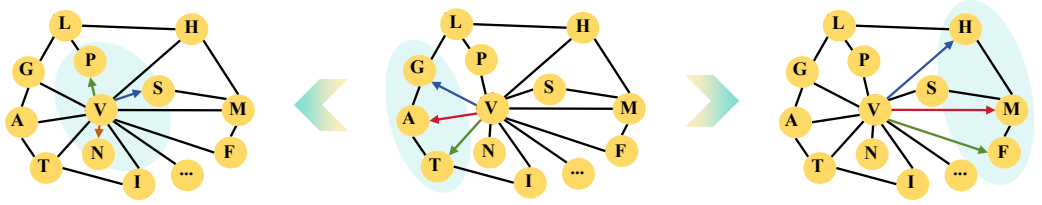
where $\hat{x}_t$ locates in the $\hat{p}_t$ position of the protein sequence.

Dynamic Sliding Window Mechanism. In the context of mutation processes, there is a tendency to mutate towards amino acids with similar properties, which may lead to convergence to the local optima. To counteract this and encourage RL exploration, we introduce a dynamic sliding window mechanism in the preferential random walk strategy, as shown in Figure 3. This algorithm serves as a nuanced complement to the preferential random walk, facilitating a broader investigation of the protein space. To achieve this, Eq. (7) can be rewritten as:

$$\hat{x}_t = \underset{d(x^i, x^j) \in [w_l, w_r]}{\arg\max} \cos(h_{x^i}, h_{x^j}), \tag{9}$$

where $[w_l, w_r]$ represents the starting position of the sliding window. The window position is dynamically adjusted based on the fitness outcomes of mutations. If a mutation results in an increase in fitness,

$$w_l = \max(w_1, w_l - \delta),\ w_r = \max(w_2, w_r - \delta), \tag{10}$$



(a) Window near the origin (b) Initial Window (c) Window far from the origin

Figure 3. Dynamic sliding window mechanism in our preferential random walk strategy. The initial window is located as shown in (b), and the node **V** is the origin of amino acids. When mutations lead to an increase in fitness, the window will slide toward the origin, as shown in (a), enabling the exploration of amino acids with similar properties. However, if consecutive mutations fail to increase fitness, the window will slide away from the origin, as shown in (c). This allows the policy network to explore amino acids with significantly different properties, thereby enhancing the potential to discover globally optimal solutions.

where δ is a small increment, w_1, w_2 are the lower bounds of the left and right endpoints respectively. If multiple consecutive mutations result in negligible changes,

$$w_l = \min(w_3, w_l + \delta),\ w_r = \min(w_4, w_r + \delta), \quad (11)$$

where w_3, w_4 are the upper bounds of the left and right endpoints respectively. The window size can be adjusted adaptively. This sliding window mechanism is pivotal to balance exploration and exploitation in reinforcement learning. It dynamically adjusts the scope of amino acid exploration based on the evolving fitness landscape, ensuring that the model does not prematurely converge on suboptimal solutions and continues to explore diverse potential mutations.

4.3. Optimize Policy Function with Pseudo Reward

The policy optimization process involves iteratively adjusting the policy network parameters to maximize accumulated rewards $\sum_{t=0}^{T} r_t$. It is worth noting that the reward function r_t is implemented by a fitness predictor, which provides a pseudo-assessment of the fitness of mutants. This process plays a crucial role in aligning the model's output with the specific goals of the directed evolution task, ensuring that each successive iteration yields a protein sequence more aligned with the desired characteristics. In our approach, the policy function includes a mutation site prediction module (composed of the trainable PLM $\mathcal{P}$ and MLP) and a mutation type prediction module (implemented via preferential random walk and is parameter-free), as presented in Section 4.2. The policy function is then optimized by maximizing the expected return, as in Eq. (2), through the policy gradient algorithm:

$$\nabla_\theta J(\theta) = E_{\tau\sim p(\tau|\theta)}\left[\sum_{t=0}^{T}\left(\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'}\right)\nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right]. \quad (12)$$

4.4. Finetune Reward Function with Annotated Mutants

The reward function r_t (i.e., the fitness predictor $\mathcal{F}$) is fine-tuned within an active learning framework. This approach iteratively samples protein sequences through the policy network, followed by annotating the fitness of these samples by an oracle. Each round of sampling and annotation contributes to the accumulating dataset used for training the fitness predictor. Hence, it is characterized by a continuous cycle of sampling, annotating, and training, allowing the model to progressively refine its predictive capabilities. The optimization of the predictor $\mathcal{F}$ involves adjusting its parameters to minimize the regression loss function in Eq.(3),

$$\ell = \frac{1}{D}\sum_{d=1}^{D}\left|\mathcal{F}(X_d;\eta) - y(X_d)\right|. \quad (13)$$

Herein, $y(X_d)$ is the annotated fitness value of the protein X_d with in the training dataset, with $d \in \{1, \cdots, D\}$.

The specific details of KnowRLM are included in Appendix B. Within the framework of active learning, we elucidate the process of optimizing the model and exploring variants through iterative cycles of reinforcement learning.

5. Experimental Results

5.1. Experimental Settings

Datasets. Our study utilized two widely recognized public datasets, GB1 (Wu et al., 2019) and PhoQ (Podgornaia & Laub, 2015), to assess the effectiveness of the proposed KnowRLM method. The GB1 dataset represents the domain B1 of protein G, a critical component in numerous biological processes. This dataset is renowned for its complexity and is extensively used to benchmark MLDE methods. It comprises a comprehensive array of 149,361 annotated mutants, derived from a possible 160,000 combinations, concentrated around four critical mutation sites: V39, D40, G41, and V54. This dataset's intricate fitness landscape, with numerous local optima and a majority of mutants displaying fitness values below the wild-type GB1, provides a rigorous testing

Table 1. Performance comparison across varying sample sizes on the GB1 dataset.

Sample size	192			288			384		
Evaluation metric	max	mean	NDCG	max	mean	NDCG	max	mean	NDCG
MLDE	0.650	0.183	0.767	0.680	0.217	0.794	0.684	0.203	0.789
ftMLDE(EVmutation)	0.725	0.233	0.791	0.770	0.280	0.814	0.935	0.414	0.833
ftMLDE(Transformer)	0.761	0.239	0.792	0.814	0.298	0.819	0.932	0.416	0.813
CLADE	0.785	0.309	0.801	0.802	0.303	0.803	0.835	0.458	0.857
CLADE2.0	0.886	0.376	0.808	0.903	0.419	0.858	0.935	0.491	0.879
EvoPlay	0.837	0.433	0.826	0.834	0.457	0.839	0.840	0.460	0.847
KnowRLM	**0.931**	**0.494**	**0.851**	**0.972**	**0.534**	**0.862**	**0.972**	**0.562**	**0.884**

Table 2. Performance comparison across varying sample sizes on the PhoQ dataset.

Sample size	192			288			384		
Evaluation metric	max	mean	NDCG	max	mean	NDCG	max	mean	NDCG
MLDE	0.309	0.069	0.753	0.364	0.087	0.775	0.361	0.095	0.791
ftMLDE(EVmutation)	0.297	0.072	0.754	0.431	0.114	0.807	0.436	0.115	0.804
ftMLDE(Transformer)	0.346	0.073	0.756	0.414	0.108	0.802	0.422	0.117	0.815
CLADE	0.319	0.070	0.759	0.441	0.089	0.762	0.467	0.089	0.777
CLADE2.0	0.345	0.118	0.781	0.383	0.125	0.779	0.399	0.148	0.814
EvoPlay	0.443	0.119	0.782	0.444	0.135	0.786	0.474	0.143	0.804
KnowRLM	**0.486**	**0.129**	**0.821**	**0.532**	**0.152**	**0.816**	**0.658**	**0.157**	**0.819**

ground for MLDE methodologies. The fitness values in this dataset primarily gauge the binding efficacy of various GB1 mutants to the antibody IgG-Fc. Complementary to GB1, the PhoQ dataset focuses on a different protein, featuring 140,517 annotated data points out of 160,000 potential mutants across four mutation sites: A284, V285, S288, and T289. The fitness value in this dataset is indicative of the phosphatase or kinase activity of various PhoQ mutants. This dataset allows for the exploration of another dimension of protein functionality, diversifying the scope of our study. Due to the unsuccessful expression of certain protein mutants in the biological experiments used to construct these datasets, it suggests that these mutants may not adhere to the fundamental principles of proteins and might not exist in the physical world. Consequently, some mutants lack fitness values. To address this issue, we implemented a strategy that imposes a higher penalty on mutants lacking fitness values, thereby discouraging such mutations.

Evaluation Metrics. The evaluation of the MLDE methodologies in this study employs a multifaceted approach to ensure a comprehensive assessment of the model performance. Normalized Discounted Cumulative Gain (NDCG) (Wang et al., 2013) serves as a principal metric due to its relevance in ranking-related problems, which is analogous to selecting high-fitness mutants among a pool of candidates. NDCG evaluates the correlation between the predicted and actual fitness values of mutants. Beyond NDCG, evaluating the model's effectiveness involves analyzing both the mean and maximum fitness values of a combined set, which includes both the samples generated during the sampling process and the top-ranking mutants identified in the prediction phase. These metrics collectively offer a comprehensive view of the model's capabilities, encompassing not only its ability to identify the highest fitness mutants (maximum value) but also the overall fitness level across the entire set of mutants considered (mean value).

We performed an extensive comparative analysis of our approach relative to five sophisticated baseline methodologies, including MLDE(Wu et al., 2019), ftMLDE(Wittmann et al., 2021), CLADE(Qiu et al., 2021), CLADE2.0(Qiu & Wei, 2022) and EvoPlay(Wang et al., 2023). Specific baseline configurations and implementation details are provided in the Appendix C. Our code is available at https://github.com/HICAI-ZJU/KnowRLM.

5.2. Main Results

Performance Comparison. To evaluate KnowRLM, we conducted a comparative analysis with several state-of-the-art (SOTA) baselines on the GB1 and PhoQ datasets. The performance of all methods is presented in Table 1 and Table 2. Our findings, as detailed in two tables, show that

Table 3. Results of ablation study on the AAKG

			max	mean	NDCG
GB1	192	Know	**0.931**	**0.494**	**0.851**
		No-Know	0.768	0.403	0.816
	288	Know	**0.972**	**0.534**	**0.862**
		No-Know	0.887	0.508	0.818
	384	Know	**0.972**	**0.562**	**0.884**
		No-Know	0.930	0.504	0.816
PhoQ	192	Know	**0.486**	**0.129**	**0.821**
		No-Know	0.447	0.126	0.793
	288	Know	**0.532**	**0.152**	**0.816**
		No-Know	0.447	0.138	0.814
	384	Know	**0.658**	**0.157**	**0.819**
		No-Know	0.446	0.143	0.819

MLDE, which often yields sequences with low fitness due to its reliance on random sampling, performed the least effectively among the methods tested. Conversely, methods like ftMLDE (EVmutation), ftMLDE (Transformer), and CLADE, CLADE2.0, EvoPlay, which respectively use prior information, hierarchical clustering, evolution score and Monte Carlo Tree in their models, mitigated this issue to varying degrees. Simultaneously, we delved into the impact of Sampling Rounds. Our observations revealed that KnowRLM requires only one or two rounds to surpass most other methods, significantly reducing the cost and time associated with annotating mutant varieties. Sample sizes of 196, 288, and 384 were indicative of one, two, and three rounds of reinforcement learning, respectively. In these settings, KnowRLM consistently outperformed the comparative methods. Additionally, it was observed that on relatively simpler datasets, such as GB1, the efficacy of KnowRLM did not significantly diminish despite a reduction in the number of training samples. This not only highlights its effectiveness but also presents a promising solution for scenarios characterized by a scarcity of biological experimental data.

Sampling Result Analysis. The ftMLDE model posits that the abundance of high-fitness mutants within the training dataset can augment predictive performance. In the GB1 and PhoQ datasets, 92% of mutants exhibit fitness values below 1% of the global maximum, highlighting the difficulty and importance of efficiently identifying high-fitness mutants during sampling. This influences the global landscape prediction for GB1 and PhoQ significantly. However, our findings suggest that, in addition to high-fitness mutants, mutants exhibiting a diverse range of fitness are also crucial. These diverse mutants play a pivotal role in enhancing the understanding of the global fitness landscape. Existing methods rely heavily on the final prediction stage, often failing to identify high-fitness and diverse mutants

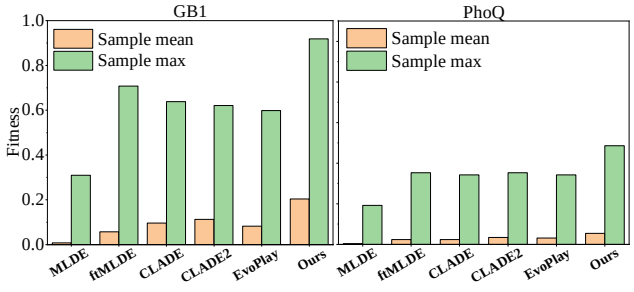


Figure 4. Evaluating the performance of 384 mutants sampled from the GB1 and PhoQ datasets.

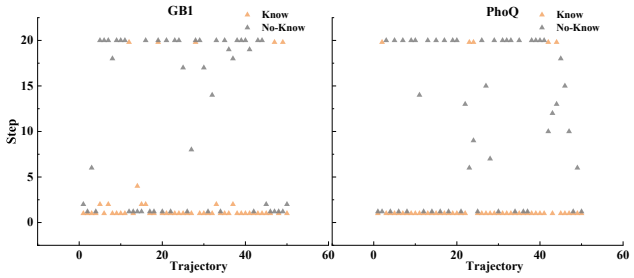


Figure 5. Analysis of the mutation process. The vertical axis represents the number of steps per trajectory, while the horizontal axis indicates the number of trajectories. A policy with AAKG can complete trajectories in fewer steps, i.e., it can more efficiently identify the variants with higher fitness than the wide-type protein.

during sampling, leading to extensive and time-consuming biological experiments. KnowRLM addresses this issue effectively. As shown in Figure 4, by comparing the mean and maximum fitness values of 384 samples, we demonstrate KnowRLM's superior sampling performance on both datasets. Particularly in the GB1 dataset, KnowRLM demonstrated superior performance compared to the reinforcement learning method Evoplay, exhibiting 1.53 and 2.46 times the maximum and average fitness values, respectively. Furthermore, it also showed notable improvement in the context of the PhoQ dataset. Meanwhile, as illustrated in Figure 5, we have visualized the advantages of knowledge-enhanced strategies in identifying variants. Compared to strategies without knowledge enhancement, these approaches require fewer actions to discover variants with high adaptability. Most often, they succeed in finding such variants within one or two actions. Additionally, the maximum number of actions required is significantly lower in comparison.

5.3. Ablation Study

We conducted an ablation experiment to investigate the impact of the knowledge graph module on the effectiveness of the reinforcement learning algorithm. To assess the significance of AAKG in the RL framework, we executed our approach without the utilization of AAKG in the policy section. Instead, the PLM was employed to fill in masked mutation sites and determine the mutated amino acids. The

Table 4. Ablation experiments on dynamic sliding windows on the GB1 dataset

		max	mean	NDCG
192	Non dynamic	0.831	0.472	0.826
	Original	0.931	0.494	**0.851**
	$w_1 \uparrow w_2 \uparrow w_3\ w_4$	**0.958**	0.470	0.813
	$w_1\ w_2\ w_3 \downarrow w_4 \downarrow$	0.951	0.496	0.815
	$w_1 \uparrow w_2 \downarrow w_3 \downarrow w_4 \downarrow$	0.945	**0.500**	0.820
288	Non dynamic	0.862	0.492	0.859
	Original	**0.972**	0.534	**0.862**
	$w_1 \uparrow w_2 \uparrow w_3\ w_4$	0.958	0.510	0.828
	$w_1\ w_2\ w_3 \downarrow w_4 \downarrow$	0.963	0.544	0.857
	$w_1 \uparrow w_2 \downarrow w_3 \downarrow w_4 \downarrow$	0.971	**0.553**	0.860
384	Non dynamic	0.862	0.513	0.826
	Original	0.972	0.551	0.878
	$w_1 \uparrow w_2 \uparrow w_3\ w_4$	0.971	0.543	0.853
	$w_1\ w_2\ w_3 \downarrow w_4 \downarrow$	0.965	0.559	0.869
	$w_1 \uparrow w_2 \downarrow w_3 \downarrow w_4 \downarrow$	**0.973**	**0.581**	**0.892**

findings, as presented in Table 3, demonstrate that across various numbers of reinforcement learning iterations, the integration of a knowledge graph invariably leads to an improvement in performance. This effect is especially pronounced under scenarios characterized by a constrained number of reinforcement learning cycles and a shortage of annotated protein data. This finding underscores the critical role of the knowledge graph in enhancing the policy network to effectively navigate the protein mutation space.

We also conducted ablation experiments on the dynamic window strategy to investigate the impact of this module on our algorithm. As shown in Table 4, when the sliding window mechanism was disabled, a significant decline in model performance was observed. This finding strongly supports the critical role of the sliding window strategy in enhancing the effectiveness of the model. Furthermore, we explored the effects of modifying the dynamic window parameters (w_1, w_2, w_3 and w_4) through additional tests, focusing on the GB1 dataset. Our experiments indicate that changing these parameters within a considered reasonable range does not substantially affect performance. However, it is noteworthy that extreme modifications to these parameters can deviate the results from our initial objectives, leading to a decline in performance. This underscores the importance of maintaining a balanced configuration of dynamic window parameters to preserve model efficiency and accuracy. Specific window parameters are detailed in Appendix C.

6. Conclusion and Future Work

This study demonstrates the significant potential of integrating knowledge graphs with reinforcement learning for protein directed evolution. Our proposed KnowRLM showcases an improved capability to identify high-fitness mutants efficiently, reducing the number of required experimental rounds and associated costs. Looking ahead, we anticipate that more sophisticated amino acid knowledge graphs can be proposed, potentially incorporating more dynamic and expansive biological data. Additionally, we can use a richer knowledge graph to assist in processing more complex mutations and explore its applicability in diverse protein engineering scenarios, aiming to broaden the horizons of computational biology and biotechnological innovation.

Acknowledgements

This work is supported by National Natural Science Foundation of China (62302433, 62301480 U23A20496), Hangzhou West Lake Pearl Project Leading Innovative Youth Team Project (TD2023017), Zhejiang Provincial "Jianbing" "Lingyan" Research and Development Program of China (2024C01135), Zhejiang Provincial Natural Science Foundation of China (LQ24F020007), CCF-Tencent Rhino-Bird Fund (RAGR20230122) and New Generation AI Development Plan for 2030 of China (2023ZD0120802).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Anishchenko, I., Pellock, S. J., Chidyausiku, T. M., Ramelot, T. A., Ovchinnikov, S., Hao, J., Bafna, K., Norn, C., Kang, A., Bera, A. K., et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

Arnold, F. H. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

Breimann, S., Kamp, F., Steiner, H., and Frishman, D. Aaontology: An ontology of amino acid scales for interpretable machine learning. *bioRxiv*, pp. 2023–08, 2023.

Chen, Z., Min, M. R., Guo, H., Cheng, C., Clancy, T., and Ning, X. T-cell receptor optimization with reinforcement learning and mutation polices for precision immunotherapy. In *International Conference on Research in Computational Molecular Biology*, pp. 174–191. Springer, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Finn, R. D., Clements, J., and Eddy, S. R. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37, 2011.

Ghugare, R., Miret, S., Hugessen, A., Phielipp, M., and Berseth, G. Searching for high-value molecules using reinforcement learning and transformers. *arXiv preprint arXiv:2310.02902*, 2023.

Hie, B., Bryson, B. D., and Berger, B. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell systems*, 11(5):461–477, 2020.

Hie, B. L. and Yang, K. K. Adaptive machine learning for protein engineering. *Current opinion in structural biology*, 72:145–152, 2022.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., Sander, C., and Marks, D. S. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Packer, M. S. and Liu, D. R. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7): 379–394, 2015.

Pearson, K. The problem of the random walk. *Nature*, 72 (1865):294–294, 1905.

Podgornaia, A. I. and Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science*, 347 (6222):673–677, 2015.

Qin, M., Ding, K., Wu, B., Li, Z., Yang, H., Wang, Z., Ye, H., Yu, H., Chen, H., and Zhang, Q. Active finetuning protein language model: A budget-friendly method for directed evolution. In *ECAI 2023*, pp. 1914–1921. IOS Press, 2023.

Qiu, Y. and Wei, G.-W. Clade 2.0: evolution-driven cluster learning-assisted directed evolution. *Journal of Chemical Information and Modeling*, 62(19):4629–4641, 2022.

Qiu, Y., Hu, J., and Wei, G.-W. Clade: Cluster learning-assisted directed evolution. *Nature Computational Science*, 2021.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1):12348–12355, 2021.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Schaff, C., Yunis, D., Chakrabarti, A., and Walter, M. R. Jointly learning to construct and control agents using deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9798–9805. IEEE, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Smith, G. P. and Petrenko, V. A. Phage display. *Chemical reviews*, 97(2):391–410, 1997.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Tian, J., Wu, N., Chu, X., and Fan, Y. Predicting changes in protein thermostability brought about by single-or multi-site mutations. *BMC bioinformatics*, 11(1):1–9, 2010.

Toyao, T., Maeno, Z., Takakusagi, S., Kamachi, T., Takigawa, I., and Shimizu, K.-i. Machine learning for catalysis informatics: recent applications and prospects. *Acs Catalysis*, 10(3):2260–2297, 2019.

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

Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R., and Rajani, N. F. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

Wang, Y., Wang, L., Li, Y., He, D., and Liu, T.-Y. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pp. 25–54. PMLR, 2013.

Wang, Y., Tang, H., Huang, L., Pan, L., Yang, L., Yang, H., Mu, F., and Yang, M. Self-play reinforcement learning guides protein engineering. *Nature Machine Intelligence*, 5(8):845–860, 2023.

Wang, Z., Combs, S. A., Brand, R., Calvo, M. R., Xu, P., Price, G., Golovach, N., Salawu, E. O., Wise, C. J., Ponnapalli, S. P., et al. Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction. *Scientific reports*, 12(1):6832, 2022a.

Wang, Z., Zhang, Q., Shuang-Wei, H., Yu, H., Jin, X., Gong, Z., and Chen, H. Multi-level protein structure pre-training via prompt learning. In *The Eleventh International Conference on Learning Representations*, 2022b.

Winter, G., Griffiths, A. D., Hawkins, R. E., and Hoogenboom, H. R. Making antibodies by phage display technology. *Annual review of immunology*, 12(1):433–455, 1994.

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

Wu, Z., Kan, S. J., Lewis, R. D., Wittmann, B. J., and Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019.

Xu, M., Wang, H., Ni, B., Guo, H., and Tang, J. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pp. 11548–11558. PMLR, 2021.

Yu, W., Liu, C. K., and Turk, G. Policy transfer with strategy optimization. *arXiv preprint arXiv:1810.05751*, 2018.

Zhang, N., Bi, Z., Liang, X., Cheng, S., Hong, H., Deng, S., Lian, J., Zhang, Q., and Chen, H. Ontoprotein: Protein pretraining with gene ontology embedding. *arXiv preprint arXiv:2201.11147*, 2022.

A. Definitions of symbols

To ease reading and facilitate understanding of our KnowRLM, in Table 5, we summarize the symbols and notations employed throughout the paper.

Table 5. Definitions of symbols used in this paper.

Symbol	Description
i, j	the index number, $i, j \in [1, 20]$
X	a protein sequence
N	the number of amino acids in a protein sequence
x^i, x^j	the i-th and j-th amino acid in nature
y	the ground-truth fitness value of a protein sequence
D	the total number of protein samples in the training set
$\mathcal{F}(\cdot)$	the sequence-to-fitness prediction function.
Z	the number of amino acid properties
K	the value of a property
$\mathcal{P}$	a PLM model that generates the protein embedding
$\mathbb{X}$	the protein space
$\mathbb{S}$	the state space
$\mathbb{A}$	the action space
$\mathbb{E}$	the probability of transitioning to the next state
s_t	state at time t in $\mathbb{S}$
a_t	action at time t in $\mathbb{A}$
τ	the trajectory of actions
π_θ	the policy that maps states to actions, also be written $\pi_\theta(a_t\|s_t)$, parameterized by θ
$p(\tau\|\theta)$	the probability distribution over trajectories
$r(s_t, a_t)$	the reward value following the execution of action a_t in state s_t
$d(x, x')$	the edge weight between the amino acid x and x'
$\ell(\cdot)$	the collective name for the loss function
$J(\cdot)$	the optimization objective function of the policy function

B. Algorithm definition

We provide a pseudo code of KnowRLM as follows so that the readers can easily understand the whole learning procedure.

Algorithm 1 KnowRLM for Directed Evolution

Input: AAKG, Initial policy network and reward model;
Output: The fitness predictor $\mathcal{F}$, Sampled mutant set $\mathcal{Q}$;
Initialize $\mathcal{Q} = \emptyset$, the rounds of active learning M, the maximum steps of optimizing policy T;
for $m = 1, \ldots, M$ **do**
  for $t = 1, \ldots T$ **do**
    Predict mutation sites by $\hat{p}_t = \text{MLP}(\mathcal{P}(s_t))$
    Predict mutated amino acid type by $\hat{x}_t = \underset{d(x^i,x^j)\in[w_l,w_r]}{\arg\max} \cos(h_{x^i}, h_{x^j})$
    Optimize the policy network by $\nabla_\theta J(\theta) = E_{\tau\sim p(\tau|\theta)}\left[\sum_{t=0}^{T}\left(\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'}\right)\nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right]$
  end
  Collect and annotate the high-reward mutants $\mathcal{Q}_m$
  $\mathcal{Q} = \mathcal{Q} \cup \mathcal{Q}_m$
  Finetune the reward model with $\mathcal{Q}$ by $\ell = \frac{1}{D}\sum_{d=1}^{D}|\mathcal{F}(X_d;\eta) - y(X_d)|, \quad D = |\mathcal{Q}|$
end
$\mathcal{F}$ = Final reward model
return $\mathcal{F}$, $\mathcal{Q}$

C. Experimental setup

Baseline. We conducted a comprehensive comparison of our method against four advanced baseline methodologies.

- **MLDE (Wu et al., 2019)**: This method employs a stochastic sampling method from the entire combinatorial space, followed by the use of these samples to train a machine learning ensemble for fitness prediction, illustrating the effectiveness of random sampling in capturing mutant diversity.
- **ftMLDE (Wittmann et al., 2021)**: Our comparison includes ftMLDE's unique sampling strategies EVmutation and MSA-transformer. This method is a blend of various sampling and encoding methods, offering a multifaceted approach to mutant selection and fitness prediction.
- **CLADE (Qiu et al., 2021)**: This method utilizes hierarchical clustering for sampling, targeting high-fitness mutants. It then employs supervised learning to train a fitness predictor, integrating clustering algorithms with precision learning.
- **CLADE2.0 (Qiu & Wei, 2022)**: An advanced version of CLADE, it incorporates a composite scoring function using profile HMM (Finn et al., 2011), EVmutation (Hopf et al., 2017), DeepSequence VAE (Riesselman et al., 2018), and a pretrained PLM, enriching the sampling phase with a multi-faceted scoring system.
- **EvoPlay (Wang et al., 2023)**: EvoPlay employs a unique combination of reinforcement learning techniques, including the Monte Carlo Tree Search (MCTS) and a policy-value neural network. This method is specifically adapted for mutating single-site residues in protein sequences, thereby optimizing their properties through an iterative, strategic process similar to game playing.

Implementation Details. We employed the open-source protein pre-trained model ESM-2 (Verkuil et al., 2022) as the PLM in the policy network. The reward function employs an ensemble machine learning framework (Wittmann et al., 2021), which includes a diverse array of robust learning methodologies. These include XGBoost, 1D Convolutional Networks, and Long Short-Term Memory (LSTM) models, thereby ensuring a multifaceted approach to the evaluation process. The synergy of these diverse methodologies enhances the precision and reliability of the fitness assessment, offering a comprehensive and nuanced understanding of the evolutionary potential of the protein mutants. Assign a positive reward value to good mutation outcomes and a negative reward value to poor ones. Notably, we impose a substantial penalty for mutations that result in illogical protein sequences, assigning a reward value of -100. Furthermore, a reward value of -1 is set for the rare instances of generating unknown protein sequences.

In accordance with the baselines, we consider amino acid alternation with $n = 4$ mutation sites. In the preparation phase of the experiment, we employed a clustering method that is consistent with the one used in the CLADE approach(Qiu et al., 2021), resulting in the sampling of 96 mutants. These samples were subsequently annotated by the oracle, a process integral to evaluating the selected samples' fitness. Following this, the 96 annotated samples served as the initial training data for the reward model, which was developed to furnish reward values for the reinforcement learning algorithm. Furthermore, the discount factor in Eq.(12) is 0.99. The entropy coefficient is set at 0.2, alongside a clipping parameter of 0.2, crucial for stabilizing the policy gradient updates. We implemented Vectorized Environments, an advanced method that aggregates multiple independent environments into a single unified environment.

Our model was developed and executed within the PyTorch framework, supplemented by the Stable-Baselines3 (Raffin et al., 2021) framework for reinforcement learning. The code is run on a Ubuntu server equipped with a single GPU (NVIDIA TESLA V100 32G), ensuring high-performance computing capabilities essential for handling the complexity of the model and the property of the data.

Ablation Experiment Supplement. Regarding the dynamic sliding window parameters, we carefully conducted experiments to ascertain the impact of varying w_1, w_2, w_3, and w_4 on our model's performance. Using the GB1 dataset as a case study, we adjusted these parameters within a predefined reasonable range (details of which are provided in Table 6).

We conducted additional experiments where we replaced AAKG with random values. As shown in the Table 7, the performance on the GB1 and PhoQ datasets both decreased. However, the introduction of noise may potentially encourage exploration in reinforcement learning, hence the mean and max results still meet the threshold. Notably, NDCG performance significantly decreased, indicating a lower understanding of the overall protein space. We also conducted the experiments that substitute AAKG with BLOSUM62. For the simpler GB1 dataset, the incorporation of domain knowledge from BLOSUM62 achieved a comparable performance to our method. However, on the more challenging PhoQ dataset, BLOSUM62 is inferior

to ours. Even when considering sample sizes of 192 and 288, we observed that both the mean and maximum performance metrics remain below those achieved under random conditions. This suggests that the generalization ability of BLOSUM62 is limited.

Table 6. Dynamic sliding window parameters

	w_1	w_2	w_3	w_4
Original	0	6	14	20
$w_1 \uparrow w_2 \uparrow w_3\ w_4$	2	8	14	20
$w_1\ w_2\ w_3 \downarrow w_4 \downarrow$	2	8	10	17
$w_1 \uparrow w_2 \downarrow w_3 \downarrow w_4 \downarrow$	4	6	8	16

Table 7. Performance comparison

Dataset	Sample size	192			288			384		
	Evaluation metric	max	mean	NDCG	max	mean	NDCG	max	mean	NDCG
GB1	Random	0.862	0.377	0.710	0.903	0.470	0.797	0.954	0.467	0.777
	BLOSUM62	0.862	**0.497**	0.845	0.904	0.515	0.851	0.972	0.542	0.841
	AAKG	**0.931**	0.494	**0.851**	**0.972**	**0.534**	**0.862**	**0.972**	**0.562**	**0.884**
PhoQ	Random	0.455	0.114	0.779	0.447	0.119	0.788	0.455	0.127	0.791
	BLOSUM62	0.316	0.097	0.763	0.337	0.097	0.759	0.486	0.132	0.807
	AAKG	**0.486**	**0.129**	**0.821**	**0.532**	**0.152**	**0.816**	**0.657**	**0.157**	**0.819**