Computer Science and Information Systems 21(4):1293–1320 https://doi.org/10.2298/CSIS240327039D

TRL-PROTAC: A pre-trained generator of PROTACs targeting specific proteins optimized by reinforcement learning

Yuhao Dai and Fei Zhu*

School of Computer Science and Technology, Soochow University
215006 Suzhou, Jiangsu, China
20224227033@stu.suda.edu.cn
zhufei@suda.edu.cn

Abstract. Proteolysis-targeting chimeras (PROTACs) introduce a novel paradigm in drug development, incorporating three essential components: the warhead, the E3 ligand, and the linker. The complexity of the ternary structure, particularly the intricate design of the linker, presents a significant challenge in PROTACs drug design. Here an integrated protocol for design and evaluation of PROTACs targeting specific proteins, called TRL-PROTAC is proposed. TRL-PROTAC is focused on the de novo design of complete PROTACs by effectively joining the designed ligands targeting the proteins of interest (POI) with linkers. The ligands for POIs and E3 ligases are generated by a molecular generation model for targeting proteins, and the linker design is generated by a sequence-to-sequence model consisting of a transformer variant and the policy-based reinforcement learning method which is employed to optimize the reward values for generating PROTACs. The three components are then integrated and optimized based on their pharmacokinetic (PK) and degradation (DEG) properties. The experimental results have strongly confirmed that TRL-PROTAC is superior in optimizing properties. For existing PROTACs, TRL-PROTAC improves DEG scores by 0.45 and lowers PK scores by 1.20. Moreover, TRL-PROTAC enhances binding affinity by 2.15 in PROTACs generated from scratch.

Keywords: proteolysis-targeting chimeras, transformer, reinforcement learning, drug design, protein-ligand interaction.

1. Introduction

Proteolysis-targeting chimeras (PROTACs) are a promising new class of drugs [1], [2], [3]. The PROTAC molecule consists of three parts: one ligand (warhead) binding to the POI, one ligand (E3 ligand) for recruiting an E3 ubiquitin ligase and a chemical bridge (linker) that links the two ligands. This enables the target protein and ubiquitin E3 ligase to be recognized and induced to undergo polyubiquitination, leading to target protein degradation and disease treatment [4]. Unlike traditional occupancy-driven inhibitors that require sufficient binding affinity to the druggable site of the target protein to exert their effects, PROTACs utilize event-driven mechanisms and only need to bind briefly to

* Corresponding author

the target protein to induce its ubiquitination and degradation. The advantages of event-driven drugs such as PROTACs are that they enable the efficacy of drugs even at sub-stoichiometric concentrations [5]. Moreover, an equivalent PROTAC can degrade multiple equivalent target proteins and avoid off-target effects caused by high drug concentrations. Additionally, many proteins play important roles in cancer development but lack druggable pockets, making them "undruggable" by small molecule drugs [6]. However, the emergence of PROTAC technology has made most protein targets in human cells "druggable", greatly expanding the therapeutic prospects of difficult-to-treat diseases such as metabolic syndrome. Moreover, the ability to persistently block downstream signals and delay kinase recombination reduces the likelihood of target compensation, as demonstrated in multiple biological evaluations of PROTACs [7].

The process of traditional drug development is notoriously low in success rate, given that it involves target validation, high-throughput screening, optimization of lead compounds, preclinical trials, and clinical trials, all of which are expensive and time-consuming [8]. However, the development of artificial intelligence (AI) has had a significant impact on the pharmaceutical industry, reducing the time and cost of drug development and shortening the development cycle [9], [10], [11]. AI has emerged as an important tool in computational-aided drug design [12], [13], [14], [15], [16]. Currently, AI-assisted drug design methods can be primarily classified into three categories: structure-based methods, ligand-based methods [17], and hybrid methods [18]. Among the structure-based methods, those focusing on protein structure and the dynamic properties of proteins are emerging as a promising direction for development [19]. Amongst, the representative AlphaDrug model was proposed by Qian et al. [20] for the de novo generation of molecules targeting specific target proteins. The Monte Carlo tree search (MCTS) algorithm was used to search for candidate drugs in all possible molecular spaces. This approach enabled candidate drugs to correctly dock with protein targets in an exclusive manner. Another representative model was the reinforcement learning (RL) model proposed by Olivecrona et al. [21]. This model utilized Recurrent Neural Networks (RNNs) [22] in combination with the reinforcement learning method for de novo drug design and introduced an augmented likelihood function for training.

Initially developed in different domains, machine learning (ML) techniques are increasingly being applied to drug discovery due to their potential to expedite the entire drug development process. As a result, a wide range of ML methods, including Artificial Neural Networks (ANNs) [23], RNNs [22], Long Short-Term Memory (LSTM) networks [24], and Transformer models [25], are being applied in drug design, particularly in addressing the challenges of predicting compound bioactivity. In contrast, deep learning in PROTACs is still in its infancy. PROTACs are distinct from conventional small-molecule drugs in that they consist of three components. When considering the generation of PROTACs, not only the effectiveness of the generated individual small molecules should be considered, but also the pharmacological properties of the overall molecule, particularly the linker generation [26]. Due to the complex and dynamic nature of the ternary structure, linker design remains a significant challenge.

Previous studies have shown that representing PROTACs as SMILES strings is feasible for linker generation. Yang et al. [27] transformed linker design into a sentence completion task and introduced a language model SyntaLinker to generate novel linkers. Zheng et al. [26] used a pair of warhead and E3 ligand as input and outputted the

designed PROTACs with favorable properties that are chemically feasible. To generate PROTACs with better pharmacokinetic (PK) properties, a Proformer model [26] was developed through pre-training and fine-tuning. This model was then integrated into a memory-augmented reinforcement learning framework with an experience reward function, which facilitated the generation of optimized PROTACs. Li et al. [8] proposed the DeepPROTACs model to help design effective PROTACs. Ligands and binding pockets were represented as graphs, and a graph convolutional network was applied to extract their features. As for linkers, they were fed into the model as SMILES strings and features were extracted using a bidirectional LSTM layer.

Zheng et al. [26] made remarkable contributions in the field of PROTACs, but their model only considered the PK property of generated PROTACs, where a smaller PK value indicates a higher oral availability of the molecule. However, in some cases, optimizing for a smaller PK value can lead to a smaller DEG value, which reduces the molecule's degradation efficacy, and this is not desirable. Furthermore, when more complex warheads and E3 ligands were input into their model, beyond the scope of the training and testing datasets, the model was unable to generate valid PROTAC molecules. The likely reason was that their molecular generation task involved first generating complete PROTAC molecules and then checking for the presence of warheads and E3 ligands. This task required the model to generate accurate warheads and E3 ligands as a prerequisite. However, the ultimate goal was to generate linkers, which made the generation of warheads and E3 ligands an additional burden for the model.

In this work, a rational strategy (TRL-PROTAC) targeting specific proteins for the design and evaluation of PROTACs with both better pharmacokinetic (PK) properties and higher degradation (DEG) efficacy was proposed for the first time. To address the issues raised in the preceding paragraph, we proposed a novel molecular generation task in which linkers were directly generated from the given warheads and E3 ligands using a pre-trained generator, the Lmser Transformer. Subsequently, these generated linkers were concatenated with the original components to obtain complete PROTACs. To acquire warheads and E3 ligands that possess higher binding affinity to target proteins, we employed the AlphaDrug model [20] to generate novel warheads and E3 ligands. To obtain PROTACs with desired properties, a policy-based reinforcement learning method is employed. Additionally, the probabilities predicted by the DeepPROTACs model were incorporated as DEG scores into the reward function of the reinforcement learning method along with PK scores, and a reward function was creatively proposed that simultaneously optimized both PK and DEG properties during the molecule generation process.

Objectives. The objectives of our work can be summarized as follows:

- Designing complete PROTACs targeting the POI. For this objective, we introduced a novel approach to design and optimize PROTACs targeting the POI (Section 3.1).
- Solving the model training problem for PROTACs with small amounts of data. For this objective, we employed a pre-training and fine-tuning approach on the Lmser Transformer model, which has been demonstrated to possess superior capabilities in drug design [20], aimed to address challenges stemming from the limited dataset (Section 3.2).
- Optimizing the properties of model-generated PROTACs. For this objective, we utilized a reinforcement learning approach with an augmented likelihood function to

optimize the Lmser Transformer, aiming to generate molecules with desirable properties (Section 3.3).
- Enhancing binding affinity between the warhead and the POI and between the e3 ligand and the e3 ligase. For this objective, the AlphaDrug model [20] was employed to generate warheads and E3 ligands with enhanced binding affinity to the POI (Section 2.1).

Contributions. Our contributions can be summarized as follows:

- We proposed an effective strategy for the rational design of PROTACs targeting specific proteins, rather than designing PROTACs based on existing warheads and E3 ligands pairs.
- We introduced a novel approach to design PROTACs, which involves generating linkers with attachment points first, and then connecting these linkers to warheads and E3 ligands.
- We considered the pharmacokinetic and degradation properties of PROTACs in the process of molecular generation by defining an appropriate reward function.
- We meticulously collected and provided structural data for the protein pockets targeted by PROTACs.
- We conducted extensive experiments and visually presented the results through illustrations.

The structure of the remaining sections of this paper is outlined as follows. Section 2 presents a review of related work and the underlying motivation. In Section 3, we delve into the specifics of our method. Section 4 elaborates on the experimental setups involving the use of the model, including data preparation. Section 5 outlines the experimental validation of the model and the subsequent analysis of the results. Section 6 discusses the potential applications of our method. Finally, Section 7 concludes and summarizes the findings.

2. Related work and motivation

This section encompasses related work and the underlying motivation. Subsection 2.1 introduces the AlphaDrug model used for de novo design of conventional small molecule drugs, highlighting the insights it brings to this study. This subsection also provides details about the AlphaDrug model. Subsection 2.2 presents the methodology employed by the DeepPROTACs model and its implications for this endeavor. Subsections 2.3 to 2.5 elaborate on the details of the three methods: Beam Search, Temperature Sampling and the PROTAC-RL model.

2.1. AlphaDrug

AlphaDrug [20] is an advanced deep learning model that possesses the capability to design ligands tailored to target proteins, taking into consideration the binding affinity between the protein and ligand during the molecule generation process. Proteins are represented as sequences of amino acids, and ligands are represented as SMILES strings. To facilitate the inefficiency which is due to the information transfer bottleneck from the top layer

of the encoder to various levels of decoder layers, a transformer variant called Lmser Transformer is utilized [20]. By integrating AlphaDrug into the TRL-PROTAC framework, we were empowered to initiate the design of PROTACs from scratch. The process commenced with the identification of high-affinity warheads specific to target proteins and high-affinity E3 ligands for specific E3 ligases. Subsequently, attachment points were established to link the designed ligands, resulting in the formation of warheads and E3 ligands pairs, pivotal for the subsequent generation of PROTACs.

Additionally, the AlphaDrug model was trained on the PROTACs dataset and compared with our model. The SMILES forms of warheads and E3 ligands are used as input to the Lmser Transformer, while the SMILES forms of linkers are used as the output. Furthermore, the MCTS algorithm is employed to optimize the properties of generated PROTACs. It plays a crucial role in AlphaDrug. The MCTS algorithm is pivotal in reinforcement learning and decision-making tasks, extensively applied to explore vast state spaces and determine optimal actions [28]. It finds applications in domains such as board games, game AI, and resource allocation [29].

MCTS's core idea is employing Monte Carlo simulations to estimate action values and guiding the search process through a constructed tree. In this work, we have introduced modifications to the MCTS process by incorporating the pre-trained Lmser Transformer for simulation purposes. The algorithm comprises four key steps:

Step 1. Selection: Starting from the root node, nodes are traversed using a certain strategy until an unexpanded node or a terminal state is reached. A modified version of the Predictor with Upper Confidence bounds applied to Trees (PUCT) algorithm [30] is used, which prioritizes nodes with fewer visits. The PUCT algorithm equation for selecting child nodes is:

$$\tilde{a}_{\mathrm{t}} = \underset{a \in \mathcal{A}}{\arg\max} \left(\mathrm{Q}\left(\tilde{C}_{\mathrm{t}-1}, a\right) + U\left(\tilde{C}_{\mathrm{t}-1}, a\right) \right) \tag{1}$$

where $\tilde{a}_{\mathrm{t}}$ is a child node after t selections, $\tilde{C}_{\mathrm{t}-1}$ represents the context with the pair of warhead and E3 ligand, and $Q\left(\tilde{C}_{t-1}, a\right) = W_a/N_a$ represents the expected reward for choosing symbol a in context $\tilde{C}_{\mathrm{t}-1}$, with W_a and N_a representing the total reward and visit count for the node; $U\left(\tilde{C}_{t-1},a\right) = c_{\text{puct}} P\left(a \,|\, \tilde{C}_{t-1}\right) \sqrt{N_t} / \left(1+N_t(a)\right)$, with c_{puct} controlling the degree of exploration.

Step 2. Expansion: If the chosen node has not been expanded, one or more child nodes are added, representing potential actions.

Step 3. Simulation: Expanded child nodes are simulated via Lmser Transformer, generating outcomes based on reward or scoring rules.

Step 4. Backpropagation: Simulation rewards propagate back along the path from the simulated node to the root, updating statistics such as visit counts and cumulative rewards.

MCTS has widespread applications in reinforcement learning, game strategy, and optimization problems. Its successful use in computer programs like AlphaGo underscores its potent search and decision-making capabilities.

2.2. DeepPROTACs

DeepPROTACs [8] is a deep learning model specifically designed to predict the degradability of PROTACs. In the DeepPROTACs model, ligands and ligand binding pockets are

generated and represented as graphs. These graph representations are then inputted into Graph Convolutional Networks (GCNs) to extract relevant features. On the other hand, SMILES representations of linkers are passed through a Bidirectional Long Short-Term Memory (BiLSTM) layer to generate their respective features [8]. In our research, we integrated the predicted probability from DeepPROTACs into the TRL-PROTAC model to guide the determination of PROTAC degradability. By incorporating this probability within the reward function of the policy-based reinforcement learning method, we were able to consider the overall degradability of PROTACs during the process of linker generation. This integration facilitated the generation of PROTACs with enhanced degradability, optimizing their effectiveness in targeted protein degradation.

2.3. Beam Search

Beam Search is a widely used decoding algorithm in sequence generation tasks, often applied to tasks like machine translation and text generation [31]. It aims to find the most likely sequence of tokens given a trained generative model. Unlike greedy decoding, which selects the token with the highest probability at each step, Beam Search maintains a fixed-size set of candidate sequences, known as the beam width.

The core idea of Beam Search is to explore multiple possible sequences simultaneously by considering a limited set of promising candidates. At each decoding step, the algorithm expands each candidate sequence by generating all possible next tokens and calculates their probabilities using the model. Then, it selects the top-k candidates with the highest probabilities based on a scoring function. These candidates become the new set of candidates for the next step. The scoring function combines the probability of the current sequence with the probability of the new token, which can be defined as:

$$\text{Score}(s, y) = \log P(y|s) \tag{2}$$

where s is a candidate sequence, y is a new token, and $P(y|s)$ is the probability assigned by the generative model.

The algorithm repeats this process iteratively until the sequences reach the desired length. One key parameter is the beam width, denoted as k, which determines how many candidate sequences are kept at each step. Larger k values allow for more exploration but may increase computation time. While Beam Search is efficient and can improve output quality compared to greedy decoding, it may suffer from generating repetitive sequences and missing out on diverse alternatives. Researchers have proposed variations like length normalization and diverse beam search to address these issues.

In summary, Beam Search strikes a balance between exploration and exploitation, leveraging a fixed-size set of candidate sequences to find high-probability solutions in sequence generation tasks.

2.4. Temperature Sampling

When discussing the outputs of generative models, "Temperature Sampling" [32] stands as a commonly used technique to modulate the diversity and determinism of model-generated results. It introduces a temperature parameter τ to reweight the output probability distribution, thereby influencing the generated outcomes. A higher temperature value

τ smoothens the distribution, encouraging the model to sample from lower probability options and enhancing result diversity. Conversely, a lower temperature value τ narrows the distribution, prompting the model to favor high probability options and generating relatively deterministic outcomes.

This temperature adjustment process can be expressed using the following equation:

$$\mathrm{Softmax}(z)_i = \frac{\exp\left(z_i/\tau\right)}{\sum_{j=1}^{n} \exp\left(z_j/\tau\right)} \tag{3}$$

where z_i represents the logits prior to softmax transformation, n denotes the possible number of symbols, and i indicates the index of a specific symbol. By dividing logits by the temperature parameter τ and then performing softmax normalization, a new probability distribution $\mathrm{Softmax}(z)$ is obtained, where values reflect the probabilities of selecting each symbol.

2.5. PROTAC-RL

The PROTAC-RL model [26] is a deep learning model designed to generate complete PROTACs based on existing warheads and E3 ligands. This model utilizes reinforcement learning to optimize the PK property of PROTACs. The model begins by one-hot encoding the source sequence (comprising warheads and E3 ligands sequences) and subsequently embedding it into a latent representation denoted as $\mathcal{H}$. The decoder leverages $\mathcal{H}$ to generate a probability distribution for the tokens, which are sampled iteratively. Sampling continues until the model encounters the ending token "</s>", ultimately forming the output sequence (the complete PROTACs sequence) $Y = (y_1, \ldots, y_m)$. The model's objective is to minimize the cross-entropy loss $\mathcal{L}$, computed between the target sequence $M_t = (t_1, \ldots, t_k)$ and the generated sequence Y:

$$\mathcal{L}(Y, M) = -\sum_{i=1}^{k} y_i \log t_i \tag{4}$$

The PROTAC-RL model optimizes the generation of PROTACs attributes through a Markov decision process. In this process, the action $A = (a_1, a_2, \ldots, a_m)$ represents the sampling of tokens from a dictionary. The state $S = (s_1, s_2, \ldots, s_m)$ is a hidden state, representing intermediate molecules sampled in previous steps. The policy π refers to the probability distribution of tokens learned by the model from the data. The reward $R(Y)$ is defined using the following equation [26]:

$$R(Y) = \begin{cases} \max\left(0, 1 - \frac{1}{\alpha}|f(Y) - \text{ Target }|\right), & \text{if valid;} \\ 0, & \text{if invalid.} \end{cases} \tag{5}$$

where α is a scalar coefficient, $f(Y)$ is a function for calculating attributes, and Target represents the custom target attribute value.

3. Methodology

In this section, we present the concrete methodology employed. Subsection 3.1 demonstrates the overarching design approach for complete PROTACs. Subsection 3.2 details

the training of ProLinker using pre-training and fine-tuning methods, along with the process of designing from warheads and E3 ligands to linkers and merging the three components. Subsection 3.3 elucidates the training of the policy network using policy-based reinforcement learning.

3.1. Architecture of TRL-PROTAC

An integrated protocol for design and evaluation of PROTACs (Figure. 1) is proposed to generate PROTACs with ideal properties (PK and DEG). As presented in Figure. 1, we first utilized the AlphaDrug model to generate warheads and E3 ligands with high binding affinity to the target proteins. To generate valid linkers, a novel end-to-end generation task was proposed for the rational design of linkers from warheads and E3 ligands. In pursuit of this goal, we initially pre-trained the Lmser Transformer, a transformer variant from the AlphaDrug model, on a large dataset of quasi-PROTAC molecules. Subsequently, the model was fine-tuned on a dataset of actual PROTAC molecules. The quasi-PROTAC dataset and the actual PROTACs dataset are two different datasets. The actual PROTACs dataset contains the PROTACs molecule, which is a ternary complex consisting of the warhead, the linker and the e3 ligand. Whereas, the quasi-PROTAC dataset is a small molecule dataset, where each data entry consists of only one small molecule. The purpose of the model being trained on the quasi-PROTAC dataset first is to allow the model to learn the syntactic information of the SMILES sequences. As a result, the trained model named ProLinker was able to generate linkers. Following that, the reinforcement learning method was utilized for step-by-step molecule generation to generate PROTACs with ideal properties.

Our model took warhead, POI pocket, E3 ligand and E3 pocket as input. In addition, the ligands are encoded by the model in the form of SMILES strings, while the pockets are encoded using mol2 files. All SMILES strings have already been tokenized to construct a vocabulary. ProLinker was applied for sampling and predicting the next token based on intermediate molecules. The reinforcement learning method generated complete molecules by performing sampling and subsequently calculated the reward score by RDKit [33] of each molecule. Formally,

$$\mathrm{PK}(X) = \mathrm{AB-MPS} = |\mathrm{LogP}(X) - 3| + \mathrm{NAR} + \mathrm{NRotB} \tag{6}$$

$$\mathrm{Reward}(X) = \mathrm{DEG} * \frac{1}{|\mathrm{LogP}(X) - 3| + \mathrm{NAR} + \mathrm{NRotB}} * 10 \tag{7}$$

where X is a PROTAC represented in SMILES format, LogP is commonly used to indicate the lipophilicity of a compound, NAR represents the number of aromatic rings, NRotB denotes the number of rotatable bonds, and DEG is the probability of degradation predicted by the DeepPROTACs model.

During the step-by-step generation process, the algorithm selects the optimal token for the current step by taking into consideration the feedback score obtained from the previous steps. According to the research conducted by DeGoey et al. [34], a notable negative correlation (correlation coefficient $r = -0.41$) exists between AB-MPS and oral availability. It was observed that a lower AB-MPS score is indicative of a higher oral availability of the drug. Apart from that, the DEG score predicted by DeepPROTACs represented the

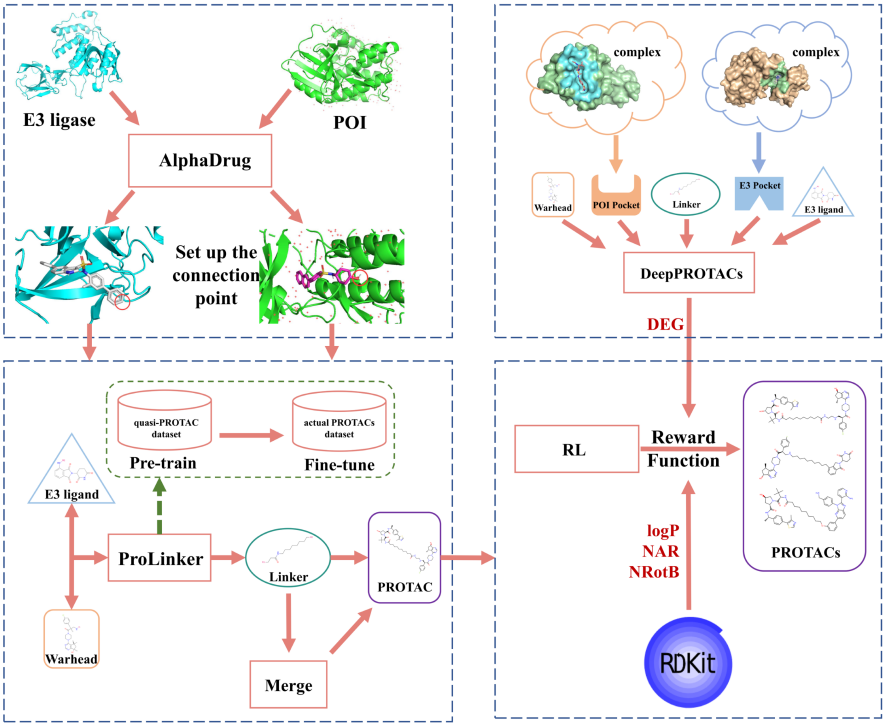

Fig. 1. Flowchart of TRL-PROTAC which is a rational strategy for design and evaluation of PROTACs targeting specific proteins. It encompasses three distinct components: the generation of warheads and E3 ligands, the generation of linkers, and the optimization of PROTAC properties

degradation efficacy of given PROTACs. A higher DEG score indicates a greater probability of degradation. The final reward function is derived from the integration of the two aforementioned scores. Due to the fact that the resulting final score obtained by dividing the DEG value by the PK value is relatively small, it has been decided to amplify the final score by a factor of 10 and utilize it as the reward value.

3.2. ProLinker

ProLinker is a crucial step in the TRL-PROTAC workflow. Building upon the validated efficacy of the Lmser Transformer architecture in the AlphaDrug model [20], ProLinker adopted the same framework for its training process. Given the inherent scarcity of available datasets specific to PROTACs, our approach involved an initial pre-training phase on a substantial collection of quasi-PROTAC molecules, followed by a fine-tuning stage utilizing actual PROTACs. Detailed information about important parameters of ProLinker is obtained in Section 4.3. Detailed procedures regarding pre-training and fine-tuning can be found in Algorithm 1 and Algorithm 2.

Algorithm 1 Pre-training with Lmser Transformer

Require: Pre-training data: quasi-PROTAC molecules D_{quasi}, learning rate: lr
Ensure: Pre-trained model parameters: $\theta_{\text{pre-trained}}$
Initialize Lmser Transformer model with random weights: $model \leftarrow$ Lmser_Transformer()
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{pre-trained}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    for $batch \in D_{quasi}$ **do**
        $input\_batch, target\_batch \leftarrow$ prepare_batch($batch$)
        $out \leftarrow model(input\_batch)$
        $loss \leftarrow$ Calculate the nll loss of out and $target\_batch$
        Backward propagation of the $loss$
        Update parameters of the $optimizer$
    end for
end for
$\theta_{\text{pre-trained}} \leftarrow$ parameters of the $model$

Algorithm 2 Fine-tuning with Lmser Transformer

Require: Fine-tuning data: actual PROTACs D_{actual}, pre-trained model: $\theta_{\text{pre-trained}}$, learning rate: lr
Ensure: Fine-tuned model parameters: $\theta_{\text{fine-tuned}}$
Initialize Lmser Transformer model with $\theta_{\text{pre-trained}}$: $model \leftarrow$ Lmser_Transformer($\theta_{\text{pre-trained}}$)
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{fine-tuned}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    for $batch \in D_{actual}$ **do**
        $input\_batch, target\_batch \leftarrow$ prepare_batch($batch$)
        $out \leftarrow model(input\_batch)$
        $loss \leftarrow$ Calculate the nll loss of out and $target\_batch$
        Backward propagation of the $loss$
        Update parameters of the $optimizer$
    end for
end for
$\theta_{\text{fine-tuned}} \leftarrow$ parameters of the $model$

Algorithm 1 is designed to initialize a Lmser Transformer model by training it on a dataset of quasi-PROTAC molecules. The objective is to equip the model with meaningful representations of molecular structures. The training loop consists of iterating over a predefined number of epochs, with each epoch comprising batches of quasi-PROTAC molecules. For each batch, the "input_batch" refers to the SMILES sequence of warhead and e3 ligand with attachment points. And the "target_batch" refers to the SMILES sequence of the linker with attachment points on both sides. Then the model predicts molecular structures. The negative log-likelihood (nll) loss is then calculated to quantify the disparity between the model's predictions and the actual targets. Through backpropagation, the model's parameters are adjusted to minimize this loss. This iterative optimization process is facilitated by the Adam optimizer. Algorithm 2 is designed for training on actual PROTAC molecules. This process involves refining a pre-trained Lmser Transformer model using a dataset of actual PROTAC molecules (D_{actual}). The pre-trained model ($\theta_{\text{pre-trained}}$), previously obtained through Algorithm 1, serves as the starting point for fine-tuning.

It is worth noting that we took a completely new approach to generate PROTACs. We generated linkers sequences with attachment points from the sequences of warheads and E3 ligands pairs. These generated linkers were then connected to warheads and E3 ligands using the attachment points to obtain complete PROTACs sequences. This differed from the conventional method of directly generating complete PROTACs sequences from the pairs of warheads and E3 ligands sequences. The input format for warheads and E3 ligands pairs was structured as "*N1CCN(c2ccc(Nc3ncc4cc(C(=O)N(C)C)n(C5CCCC5)c4n3)nc2)CC1.*Nc1cccc2c1C(=O)N(C1CCC(=O)NC1=O)C2=O", where "*" denoted the attachment point and "." represented the boundary between the warhead and the E3 ligand. The input sequence is initially tokenized into individual tokens from the vocabulary, resulting in a sequence of tokens. Subsequently, the token sequence is mapped to numerical vectors based on the positions of different tokens in the vocabulary. Vectors were then input into the model for encoding by the encoder. The model received information from the encoder for processing and outputted a probability distribution over different tokens in the vocabulary through the decoder. Afterward, a sequence was generated by sampling based on this probability distribution. The output format for linkers was represented as "[1*]CCOCCOCC[2*]", where "[1*]" denoted the attachment point between the linker and the warhead, and "[2*]" denoted the attachment point between the linker and the E3 ligand. By adopting this generation approach, the model was still capable of generating valid linkers and assembling them into functional PROTACs, even in cases where the structures of warheads and E3 ligands were complex.

We trained the Lmser Transformer model on prepared PROTACs datasets with the objective of minimizing the following loss function:

$$J(\Theta) = -\sum_{(S,m)\in\mathcal{D}}\sum_{\tau=1}^{t}\sum_{a\in\mathcal{V}} y_a \ln P\left(a \mid C_\tau(S,m)\right) \tag{8}$$

where Θ denotes the parameters of the Lmser Transformer network, $\mathcal{D}$ is the training dataset, $\mathcal{V}$ is the vocabulary, S is the input sequence, $m = (a_1, a_2, \ldots, a_t)$ is the target sequence, $C_\tau = \{S, (a_1, a_2, \ldots, a_\tau)\}, \tau = 1, \ldots, t$, represents a sequence of contexts, $P(a \mid C_\tau)$ is the probability of the next token $a_{\tau+1}$, and y_a is a binary label indicating whether a is the next token.

3.3. Optimizing policy network with reinforcement learning

A policy-based reinforcement learning method is employed to optimize the policy network. The policy-based approach constitutes a significant methodology within reinforcement learning [35], [36]. Policy-based approaches enable the explicit learning of an optimal stochastic policy [37]. Figure. 2 provides a comprehensive depiction of the procedural steps involved in training the agent through the utilization of reinforcement learning. This method shares the same concept as that of [21].

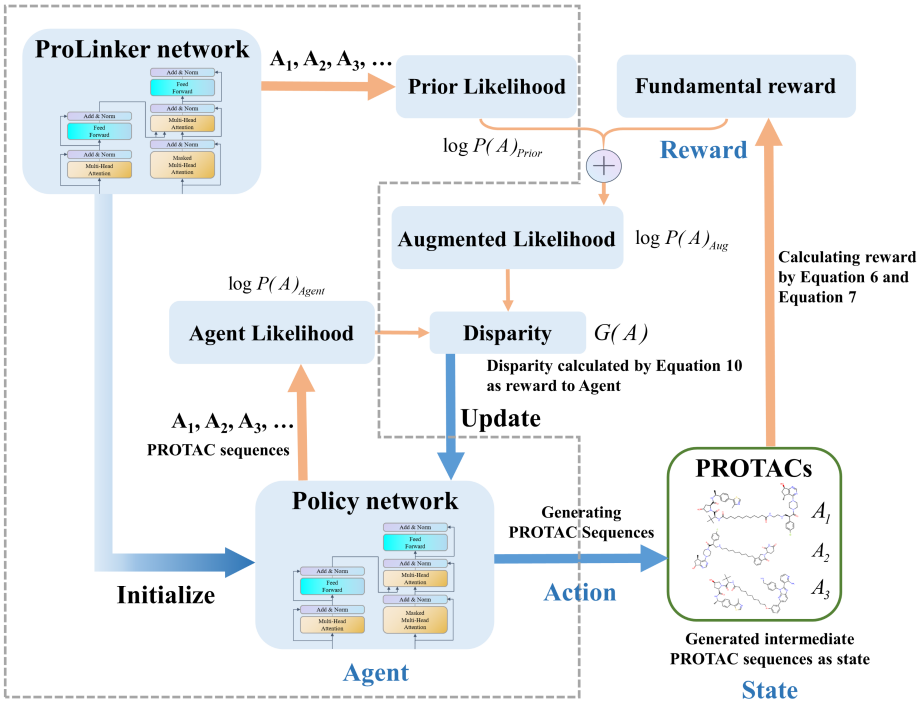

Fig. 2. The illustration of constructing the policy-based reinforcement learning model. Continuous actions are taken by the policy network until a complete PROTAC sequence is generated. Subsequently, the fundamental reward is calculated based on Formula 6 and Formula 7. The ProLinker network computes the prior likelihood for the generated PROTAC sequence. The augmented likelihood is obtained by combining the fundamental reward with the prior likelihood. The disparity between the augmented likelihood and the agent likelihood is then used as the final reward to update the agent

The reinforcement learning training process usually entails equipping an agent with the task of choosing an action, denoted as $a \in A(s)$, based on a specific state, $s \in S$, where S encompasses all possible states [38]. Here, $A(s)$ stands for the set of actions available in those states. The agent's policy, denoted as $\pi(a \mid s)$, indicates the probability

of selecting action a in state s, and $r(a \mid s)$ represents the resulting outcome or reward following a specific action. In this work, the reinforcement learning modeling is outlined as follows.

Reinforcement learning modeling.

- **Agent:** The policy network responsible for generating PROTAC sequences.
- **State** $s \in S$: The intermediate molecule in the process of PROTAC generation.
- **Action** $a \in A(s)$: Selection of the next token in the vocabulary based on the intermediate molecule.
- **Reward** $G(A)$: Accumulated reward obtained by calculating molecular properties after generating a complete molecule.

We intend to utilize the Lmser Transformer architecture as the agent to proficiently generate PROTACs represented in the form of SMILES strings, while ensuring the incorporation of desired attributes. This task is framed within the context of a partially observable Markov decision process, where the agent's objective is to determine the subsequent token a in the vocabulary, based on the given intermediate sequence s. An episodic task refers to a task with a well-defined endpoint after undergoing step T [39]. This task starts sampling from the "&" symbol and concludes upon sampling "$" making it an episodic task. In this task, the action, denoted as a, represents the selection of a token from the vocabulary $\mathcal{V}$ to be added to the generated sequence. The state s refers to the currently generated intermediate sequence. The policy $\pi(a \mid s)$ signifies the probability distribution for choosing the next token in the intermediate sequence. $r(a \mid s)$ represents the reward value obtained after selecting the next token. In the molecular generation task, as the properties of a molecule are only meaningful once the entire molecule is generated, we calculate the cumulative reward $G(A)$ after generating the complete molecule $A = a_1, a_2, ..., a_T$. The product of action probabilities $P(A) = \prod_{t=1}^{T} \pi\left(a_t \mid s_t\right)$ signifies the likelihood of the generated sequence by the model.

The agent is characterized by a structural configuration akin to that of the ProLinker network, and it is initialized using ProLinker network parameters that have undergone meticulous fine-tuning. The agent policy π is securely tethered to the pre-existing ProLinker policy, skillfully amalgamating the rewards acquired from the attribute values of the sampled PROTACs. This leads to a novel augmented likelihood $\log P(A)_{\text{Aug}}$. Formally,

$$\log P(A)_{\text{Aug}} = \log P(A)_{\text{Prior}} + \sigma Reward(A) \tag{9}$$

where $P(A)_{\text{Prior}}$ represents the prior likelihood calculated by the ProLinker model after step T, σ is a scalar coefficient, and $Reward(A)$ represents the property value of the generated molecule and is calculated using equation 7. The objective of the agent is to acquire a policy that effectively mitigates the disparity between the agent likelihood $\log P(A)_{\text{Agent}}$ and the augmented likelihood $\log P(A)_{\text{Aug}}$. This disparity will be represented as the reward $G(A)$. Formally,

$$G(A) = -\left[\log P(A)_{\text{Agent}} - \log P(A)_{\text{Aug}}\right]^2 \tag{10}$$

The current objective of the agent is to optimize the reward value $G(A)$, which is attained by minimizing the loss function $J(\theta) = -G$. The detailed specifics of the policy-based optimization algorithm can be found in Algorithm 3.

Algorithm 3 is devised for optimizing the policy network through reinforcement learning. It utilizes the fine-tuned model ($\theta_{\text{fine-tuned}}$) to initialize both the prior network ($Prior$) and the policy network ($Agent$). The parameters of the fine-tuned model are retained as frozen prior model parameters (θ_{prior}). The optimization process involves sampling a batch of molecules using the agent, computing likelihoods for both the agent and prior networks, calculating scores, and then updating the parameters based on the loss function. The aim is to obtain updated agent model parameters (θ_{agent}) over a specified number of epochs.

Algorithm 3 Optimizing policy network with reinforcement learning

Require: Fine-tuned model: $\theta_{\text{fine-tuned}}$, batch size: $batch\_size$, learning rate: lr
Ensure: Updated agent model parameters: θ_{agent}, Frozen prior model parameters: θ_{prior}
Initialize prior network ($Prior$) with $\theta_{\text{fine-tuned}}$: $Prior \leftarrow$ Lmser_Transformer($\theta_{\text{fine-tuned}}$)
Initialize policy network ($Agent$) with $\theta_{\text{fine-tuned}}$: $Agent \leftarrow$ Lmser_Transformer($\theta_{\text{fine-tuned}}$)
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{fine-tuned}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    Initialize the set of molecules $\mathbf{D_s}$
    for $i \leftarrow 1$ to $batch\_size$ **do**
        $a_0 \leftarrow$ the start symbol
        for $t \leftarrow 1$ to T **do**
            Select action $a_t \leftarrow \arg\max_a \pi_{agent}(a_{t-1}|s_{t-1})$
        end for
        $A_i \leftarrow a_0 a_1 \dots a_T$
        Add A_i to $\mathbf{D_s}$
    end for
    $agent\_likelihood \leftarrow$ Compute the likelihood of $\mathbf{D_s}$ on $Agent$
    $prior\_likelihood \leftarrow$ Compute the likelihood of $\mathbf{D_s}$ on $Prior$
    $score \leftarrow$ calculate scores of $\mathbf{D_s}$
    $augmented\_likelihood \leftarrow prior\_likelihood + \sigma score$
    $loss \leftarrow (augmented\_likelihood - agent\_likelihood)^2$
    Backward propagation of the $loss$
    Update parameters of the $optimizer$
end for
$\theta_{\text{agent}} \leftarrow$ parameters of the $Agent$

4. Experimental settings

This section encompasses pertinent details of the experimental setup. Subsection 4.1 illustrates the data preparation process for the subsequent experiments. Subsection 4.2 provides a detailed overview of the comparative methods employed. Subsection 4.3 presents the hyperparameter settings for pre-training and fine-tuning. Subsection 4.4 outlines the hyperparameters for the reinforcement learning method used in training the policy network.

4.1. Data preparation

4.1.1 Data for training ProLinker

The datasets used for training and validating ProLinker were curated by Zheng et al. [26] The pre-training dataset consisted of a collection of quasi-PROTAC small molecules carefully selected from the ZINC dataset. These molecules were chosen based on their similarity in chemical space to PROTACs. The fine-tuning data, on the other hand, comprised actual PROTACs obtained from the PROTAC-DB database.

These datasets consist of input data (src) and target data (tgt). The input data comprises pairs of warheads and E3 ligands, while the target data consists of complete PROTACs. To facilitate the proposed novel task, we have retained the original input data while modifying the target data from complete PROTACs molecules to individual linkers. Finally, a total of 204,840 quasi-PROTAC small molecules and 6,556 actual PROTACs were acquired. Following the methodology of previous investigations, these datasets were partitioned into training and validation sets in a ratio of 8:1.

4.1.2 Data for generating PROTACs based on existing warheads and E3 ligands

To validate the effectiveness and superiority of the ProLinker and reinforcement learning method combination, we first employed them to generate PROTACs on the existing dataset. The preprocessing of the dataset is depicted in Figure. 3. Due to the fact that the ProLinker model has already been trained on the training and validation sets, only the test set (consisting of a total of 165 pairs of Warhead and E3 Ligand molecules) was selected for linker generation. By using the structure search tool provided by the PROTAC-DB database [40], the corresponding information for the PROTACs in the database (a total of 162 molecules) can be retrieved. The PROTAC-DB database provides information such as the ID, name, target name, and E3 Ligase name for the molecules. Additionally, detailed information for both the Warhead and E3 Ligand molecules can be viewed, and links to external databases such as the ChEMBL database [41] are provided. In the ChEMBL database, information about the three-character representation of the ligand molecule in the PDBeChem database [42], [43], [44] can be obtained. Using the Uniprot [45] ID for the target protein and the three-character representation of the ligand molecule, the receptor protein-ligand small molecule complex structure can be retrieved from the RCSB PDB database [46]. A total of 54 pairs of warheads and E3 ligands have been obtained, all of which have well-defined complex structures with the target proteins. Detailed information about these 54 pairs of warheads and E3 ligands is obtained in Supplementary Table S1.

In the subsequent step, using the ProDy [47] package in Python, the protein pocket structure within 5 Å of the ligand molecule was obtained based on the name of the complex structure and the three-character representation of the ligand molecule. Subsequently, the protein pocket structure was converted from PDB format to MOL2 format using the Open Babel (obabel) tool [48]. Finally, the warheads, E3 ligands, and corresponding pockets were input into the model for PROTACs generation.

4.1.3 Data for de novo design of PROTACs targeting specific proteins

Subsequently, the ProDy package [47] was utilized to select individual chains from complex structures and separate the protein and ligand. The protein structure was saved in PDB format, while the ligand structure was saved in SDF format. A total of 28 unique

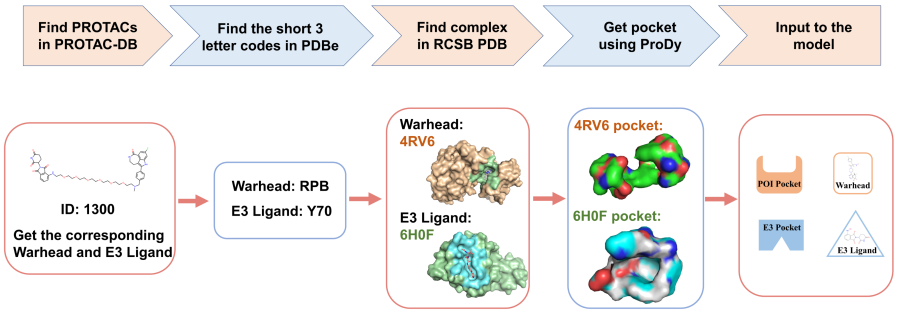

Fig. 3. The comprehensive workflow for data processing.

protein-ligand pairs were obtained. These two types of structures were then input into the AlphaDrug model for the generation of ligands with higher binding affinity. The generated molecules with the highest binding affinity will be selected as the ligands for the corresponding target proteins and E3 ligases for linker generation. The generated linkers will then be further processed and combined to obtain the final PROTACs.

4.2. Comparative methods

We compared our method with other methods, including Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model.

- Beam Search is an algorithm commonly employed in tasks involving the generation of sequences, like language translation or text generation. It operates by concurrently exploring several potential solutions, maintaining a restricted set of the most promising options (referred to as the "beam"), with the goal of identifying the most probable sequence based on a scoring mechanism.
- Temperature Sampling is a method commonly employed in probabilistic models, particularly in generative models like language models. It entails manipulating the temperature parameter during the sampling process, which in turn influences the diversity of generated outputs: higher values amplify randomness, while lower values induce a greater degree of predictability in the generated results.
- The AlphaDrug model is a deep learning model designed for creating custom ligands that are specific to target proteins, considering the binding affinity between proteins and ligands during molecule generation. It consists of a transformer variant known as the Lmser Transformer and the MCTS algorithm. The Lmser Transformer is used to address the inefficiency caused by information transfer bottlenecks in the encoder-decoder architecture. And MCTS is a sophisticated algorithmic strategy frequently applied for making decisions in intricate domains with multiple choices. MCTS systematically constructs a decision tree, intelligently exploring and exploiting possibilities, resulting in improved decision-making especially in situations marked by ambiguity and extensive options [30].

- PROTAC-RL is built upon the SyntaLinker [27] language model, enabling the generation of complete PROTACs sequences from existing warheads and E3 ligands sequences. The model consists of a pre-trained SyntaLinker model for learning the representation of PROTAC molecules and the reinforcement learning approach to optimize the PK property of generated PROTACs.

For specific descriptions of these methods, refer to Section 2.1 and Sections 2.3 to 2.5.

4.3. Hyperparameters in pre-training and fine-tuning

As the pre-training and fine-tuning employed the same model architecture, it was essential to maintain consistency in hyperparameters, as outlined in Table 1. Here, "fragsVoc" represents the vocabulary for input warheads and E3 ligands, while "LinkerVoc" represents the vocabulary for output linkers. "fragsMaxLen" and "LinkerMaxLen" correspondingly indicate the maximum lengths of input and output sequences. "fragsPaddingIdx" and "LinkerPaddingIdx" signify the indices of the start characters in input and output sequences. "frags_voc_len" and "Linker_voc_len" denote the vocabulary sizes. The following are common parameters used for training the Transformer model.

4.4. Hyperparameters in optimizing policy network

The hyperparameters for the policy-based optimization approach are detailed in Table 2. Here, "batch_size" indicates the number of samples used for training the policy network in each iteration. "epoch" and "lr" respectively denote the training epochs and learning rate for the policy network. "σ" represents the constant in Equation 9. "σ" is set to 2000 to balance the difference in magnitude between the prior likelihood computed by the ProLinker model and the reward scores.

5. Results and analysis

In the initial phase, we commenced with the pre-training of the ProLinker model on the quasi-PROTAC dataset, which was subsequently fine-tuned on the actual PROTACs dataset. Afterwards, utilizing the ProLinker model in conjunction with the reinforcement learning method, we generated 54 sets of PROTACs derived from corresponding pairs of warheads and E3 ligands (a detailed description of the data can be found in Section 4.1). Each set contains PROTACs with different PK values, DEG values, and reward values. We visualized the distribution of the scores of the generated PROTACs (Section 5.1), compared them with the original PROTACs (Section 5.2), and compared them with PROTACs generated by Beam Search, Temperature Sampling, AlphaDrug and PROTAC-RL (Section 5.3). Lastly, we embarked on de novo designs of PROTACs targeting the 33 specific proteins (after removing duplicates). And we selected the top 5 reward scores of PROTACs generated by AlphaDrug and TRL-PROTAC for comparison (Section 5.4). Moreover, we employed visualizations to enhance the analysis and evaluation of the designed PROTACs.

Table 1. Important hyperparameters of training ProLinker

hyperparameters	pre-training/fine-tuning
fragsVoc	"#", "$", "&", "(", ")", "*", "-", ".", "/", "1", "2", "3", "4", "5", "6", "7", "=", "B", "Br", "C", "Cl", "F", "I", "N", "O", "P", "S", "[B-]", "[C+]", "[C-]", "[C@@H]", "[C@@]", "[C@H]", "[C@]", "[C]", "[N+]", "[N-]", "[N@+]", "[N@@+]", "[O-]", "[O]", "[P+]", "[P@@]", "[P@]", "[S+]", "[S@@]", "[S@]", "[Si]", "[Sn]", "[c-]", "[n+]", "[nH]", "[s+]", "\\", "^", "c", "n", "o", "p", "s"
LinkerVoc	"#", "$", "&", "(", ")", "-", ".", "/", "1", "2", "3", "4", "5", "6", "7", ":", "=", "B", "Br", "C", "Cl", "F", "I", "N", "O", "P", "S", "[1*]", "[2*]", "[B-]", "[B@-]", "[BH-]", "[C+]", "[C-]", "[C@@H]", "[C@@]", "[C@H]", "[C@]", "[C]", "[IH2]", "[N+]", "[N-]", "[N@@+]", "[NH3+]", "[O-]", "[O]", "[P+]", "[P@@]", "[P@]", "[S+]", "[S@@]", "[S@]", "[Si]", "[Sn]", "[c-]", "[n+]", "[nH]", "[s+]", "\\", "^", "c", "n", "o", "s"
fragsMaxLen	202
LinkerMaxLen	91
fragsPaddingIdx	54
LinkerPaddingIdx	59
frags_voc_len	60
Linker_voc_len	64
batch_size	32
epoch	500
lr	0.0001
d_model	96
dim_feedforward	256
num_layers	4
nhead	4

Table 2. Important hyperparameters of training policy network

hyperparameters	optimizing policy network
batch_size	32
epoch	1000
lr	0.005
σ	2000

5.1. Horizontal comparison of all generated PROTACs

The ProLinker model is initially pre-trained on a quasi-PROTAC dataset containing a large number of small molecules to learn the chemical language of molecules represented by SMILES sequences. Subsequently, the ProLinker model undergoes fine-tuning on an actual PROTAC dataset to learn the specific chemical language of PROTACs. The detailed data of pre-training and fine-tuning processes is obtained in Supplementary Tables S2 and S3. The fine-tuned ProLinker is further optimized using a policy-based reinforcement learning approach to generate PROTACs with specific properties. We conducted 54 sets of experiments using existing warheads and E3 ligands to validate the effectiveness and superiority of the ProLinker model combined with the reinforcement learning method.

To provide a more intuitive representation of the distribution of scores for all generated PROTACs across multiple dimensions, we utilized the CMplot package [49] in R to generate the Manhattan plot for multiple scores. The Manhattan plot was originally designed to visualize the summary statistics results of genome-wide association studies (GWAS), especially p-values [50]. It can display the different p-values of multiple tracks at different positions of the genome. Here, we used three different scores as three different tracks and drew a circular scatter plot with PROTAC IDs as genomic positions. This plot enables us to visualize the distribution of the generated PROTAC scores from multiple dimensions, and also allows for the horizontal comparison between different PROTAC IDs.

As shown in Figure. 4, the outermost circle of the figure represents different IDs of PROTACs in PROTAC-DB, while the three trajectories from top to bottom represent the scores of PK, DEG, and reward score, respectively. This figure intuitively shows the score distribution of the model during the molecule generation process in three dimensions. It can be observed that most of the samples show long bar-shaped distributions for the three scores, indicating that the scoring of the model is not single-valued, but distributed from low to high. This further confirms that the reinforcement learning method does guide the generation of molecules with desired properties. Many molecules in the figure have PK values reaching around 15.75, and a considerable portion of molecules have DEG values above 0.5. The figure also allows for visual comparison between different molecules, showing that different molecules can be optimized to varying degrees. For example, the molecule with ID 939 has a reward value that is only presented as two dots. Because its DEG value is relatively low, and the space for optimizing this molecule is relatively small. On the other hand, the molecule with ID 1399 has a wide range of PK value changes, and can even reach a minimum of around 15, while its DEG value also changes significantly, ranging from 0 to 1. This indicates that the space for optimizing this molecule is quite large. Detailed information about these scores of these generated PROTACs is obtained in Supplementary Table S4.

5.2. Comparison with original PROTACs

In order to elucidate the substantial enhancement of our model-generated PROTACs over the original PROTACs, the original PROTACs were utilized to compute the DEG values, PK values, and reward values. As there exists only one initial PROTAC for each pair of warhead and E3 ligand, the top 3 PROTACs generated by our model ranked by reward

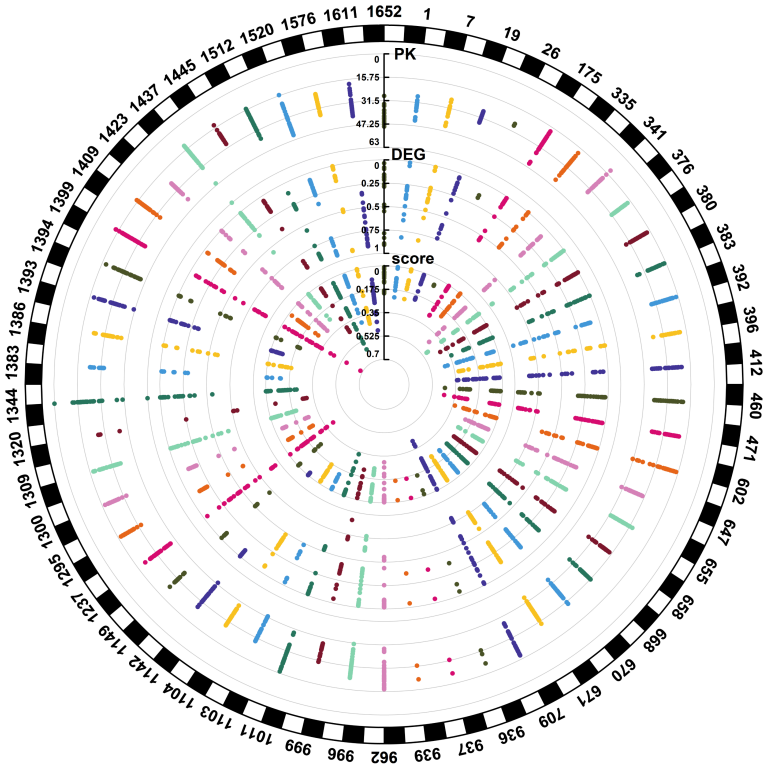

Fig. 4. The three scores, namely reward score, DEG score, and PK score, are visually represented on three distinct tracks in the Manhattan plot

values were averaged and contrasted with the original PROTACs. To present the comparison more effectively, a dumbbell plot using the ggplot2 package [51] in R was illustrated as depicted in Figure. 5. In the figure, blue indicates the original PROTACs, while green represents the PROTACs generated by our model. It is visually evident from the figure that the DEG values of most PROTACs generated by our model are significantly higher than those of the original PROTACs. At the same time, a considerable proportion of PROTACs generated by our model have lower PK values than the original PROTACs. Based on the final reward values, almost all PROTACs generated by our model have higher reward values than the original PROTACs. Our model relies on the reward value to generate PROTACs, and overall, it has successfully optimized the reward value for generating PROTACs. The DEG and PK values of these PROTACs generated by our model are relatively optimal. Therefore, the PROTACs generated by our model provide a comprehensive improvement over the original PROTACs. Detailed information about these scores of original PROTACs is obtained in Supplementary Table S5.

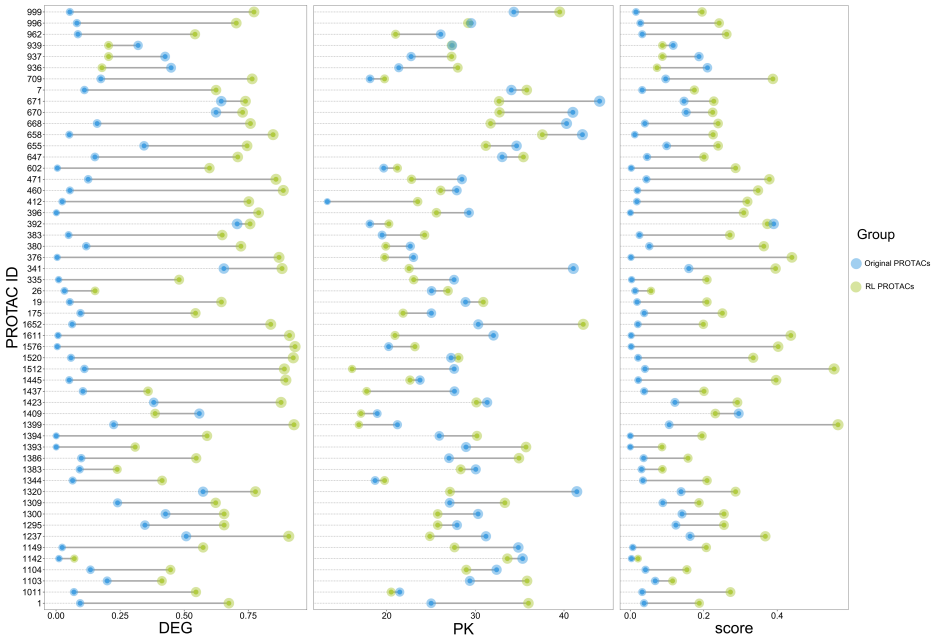

Fig. 5. Comparison between original PROTACs and PROTACs generated by TRL-PROTAC. The blue dots in the figure indicate the original PROTACs, and the green dots indicate the PROTACs generated by our model. The vertical axis indicates the different PROTAC IDs, and the horizontal axis demonstrates the three scores of the PROTACs

5.3. Comparison with other methods

To further demonstrate the superiority of our model (TRL-PROTAC), we used the classic Beam Search algorithm (beam_size=20, n_best=5), Temperature Sampling, the AlphaDrug model and the PROTAC-RL model to generate PROTACs. Here, the top 5 best PROTACs were obtained and compared with the top 5 PROTACs with the best reward scores generated by our model, as shown in Figure. 6. The figure shows that a considerable number of the PROTACs generated by TRL-PROTAC (red) have better PK values compared to those generated by the Beam Search algorithm (blue), temperature sampling (deep blue), the AlphaDrug model (orange) and the PROTAC-RL model(green). Moreover, TRL-PROTAC has a significant advantage in terms of DEG scores and reward scores for almost all PROTACs. Therefore, it can be concluded that the combination of the ProLinker model and reinforcement learning method is quite successful, as it can comprehensively consider PK and DEG scores to optimize the generated PROTACs. Detailed information about these scores of PROTACs is obtained in Supplementary Tables S6, S7 and S8.

To further perform statistical analysis, the scores of different PROTAC IDs for the same model were aggregated and subjected to significance tests (Wilcoxon test), as shown in Figure 7. The results of the significance tests are indicated by statistical significance markers in the figure. A single asterisk "*" denotes a p-value less than 0.05, two aster-

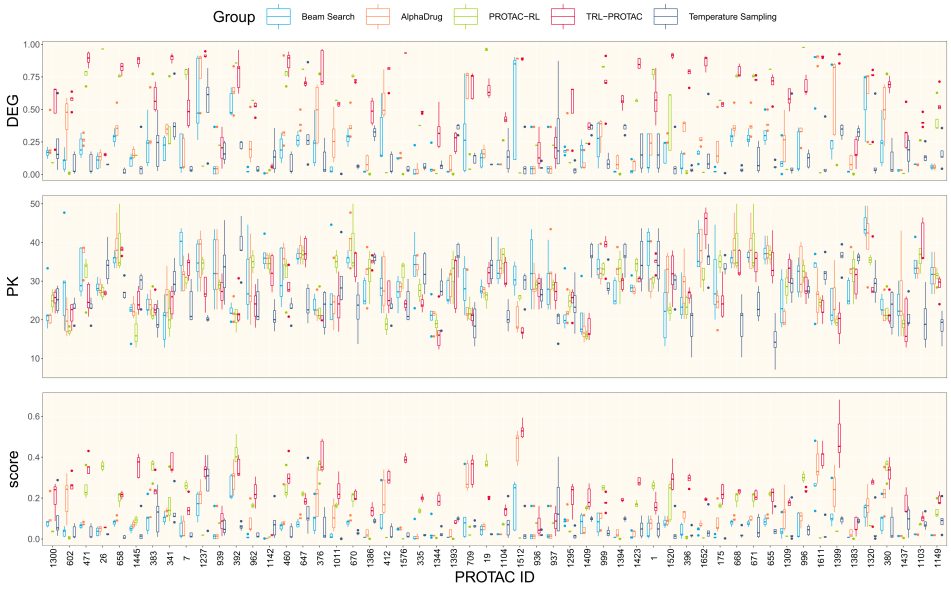

Fig. 6. Comparison between PROTACs generated by Beam Search, Temperature Sampling, AlphaDrug, PROTAC-RL and TRL-PROTAC. The horizontal axis of the figure represents the different PROTAC IDs and the vertical axis represents the three scores of PROTACs. The different coloured boxes indicate the PROTACs generated by different models, where the red boxes are the scores of the PROTACs generated by our model

isks "**" indicate a p-value less than 0.01, three asterisks "***" signify a p-value less than 0.001, four asterisks "****" indicate a p-value less than 0.0001, and "ns" represents a p-value greater than 0.05. From the figure, it can be observed that DEG scores of TRL-PROTAC, in the significance tests against other models, are marked with four asterisks, indicating an extremely significant difference between the DEG scores of our model and those of other models. Furthermore, considering the data distribution in the boxplot, TRL-PROTAC achieves higher DEG scores compared to the other models, indicating a significant superiority of TRL-PROTAC in terms of DEG scores. Regarding PK scores, the significance tests reveal that TRL-PROTAC exhibits a significant difference when compared to Beam Search and AlphaDrug, but no significant difference is observed in comparison to PROTAC-RL and Temperature Sampling. Looking at the data distribution in the boxplot, PK scores of TRL-PROTAC are slightly lower than those of Beam Search and AlphaDrug (lower PK scores are better), while they are similar to the distributions of PROTAC-RL and Temperature Sampling. This could be attributed to the PROTAC-RL model's emphasis on optimizing PK scores, while the Temperature Sampling method enhances sample diversity, making it easier to obtain molecules with better PK scores. However, TRL-PROTAC stands out because it can optimize both PK and DEG scores. Therefore, the final reward score is significantly better than that of other models.

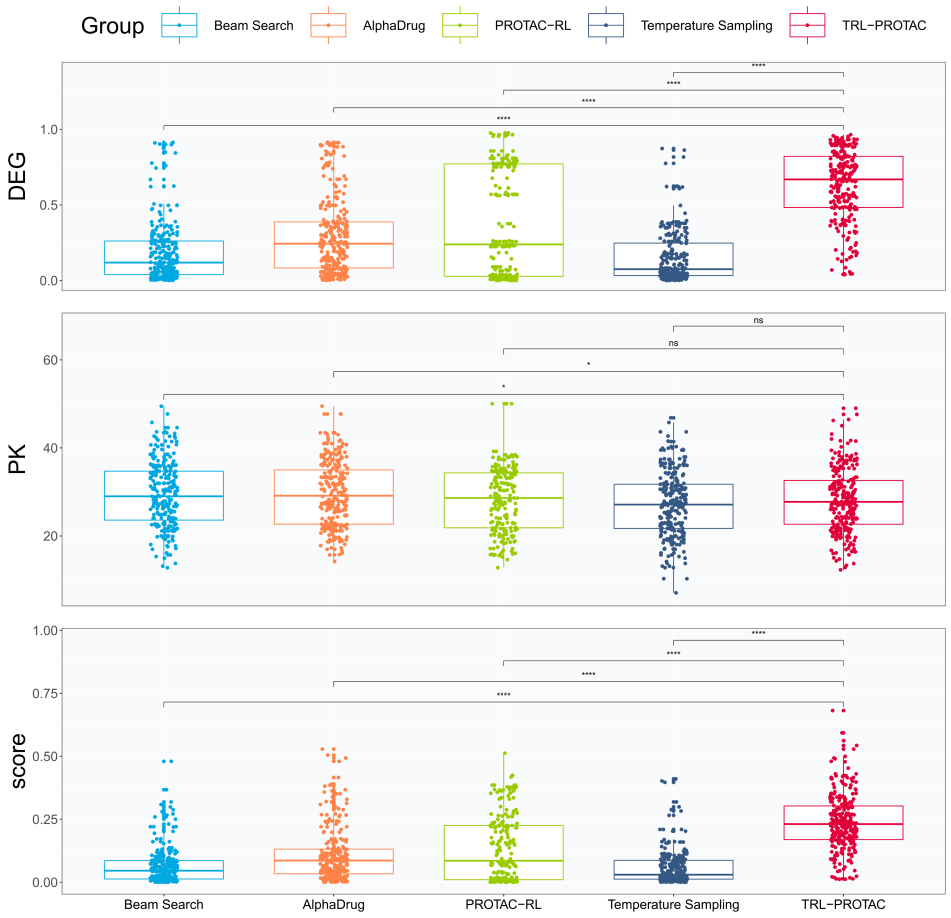

Fig. 7. Comparison between Beam Search, Temperature Sampling, AlphaDrug, PROTAC-RL and TRL-PROTAC after aggregating the scores of different PROTAC IDs. Different coloured boxes indicate PROTACs generated by different models, and the asterisks above the boxes are significance notations for significance tests between the corresponding models

5.4. Comprehensive analysis of the de novo design of PROTACs

As illustrated in Figure. 8a, through the utilization of AlphaDrug, we employed a targeted approach to design warheads and E3 ligands tailored specifically to the protein and E3 ligase, respectively. The resulting ligands demonstrated significantly enhanced binding affinities to their respective proteins. In fact, the majority of the generated ligands exhibited higher binding affinities than their original counterparts, with an impressive average increase of 2.15. Continuing from there, we proceeded to design PROTACs based on the generated ligands. As illustrated in Figure. 8b-c, the resulting PROTACs exhibited notably elevated DEG scores, with the majority exceeding the 0.5 mark. Furthermore, compared

to the MCTS algorithm, our model has demonstrated superiority across all three scores. This outcome underscores the efficacy of our model in optimizing the degradability of the generated PROTACs—an achievement that has proven elusive for other models. However, it is worth noting that these PROTACs displayed relatively high PK scores in both of these methods. This can be attributed to the fact that the ligands devised by AlphaDrug, while exhibiting enhanced binding affinities for the proteins, often showcased intricate structures such as heterocyclic motifs. Consequently, the inherent PK scores of the warheads and E3 ligands were inherently higher, thereby yielding a correspondingly elevated PK score for the generated PROTACs. Detailed information about the binding affinity scores of protein-ligand pairs is obtained in Supplementary Table S9. The scores of de novo designed PROTACs using the TRL-PROTAC model can be found in Supplementary Table S10, while the scores of de novo designed PROTACs using the AlphaDrug model are available in Supplementary Table S11.

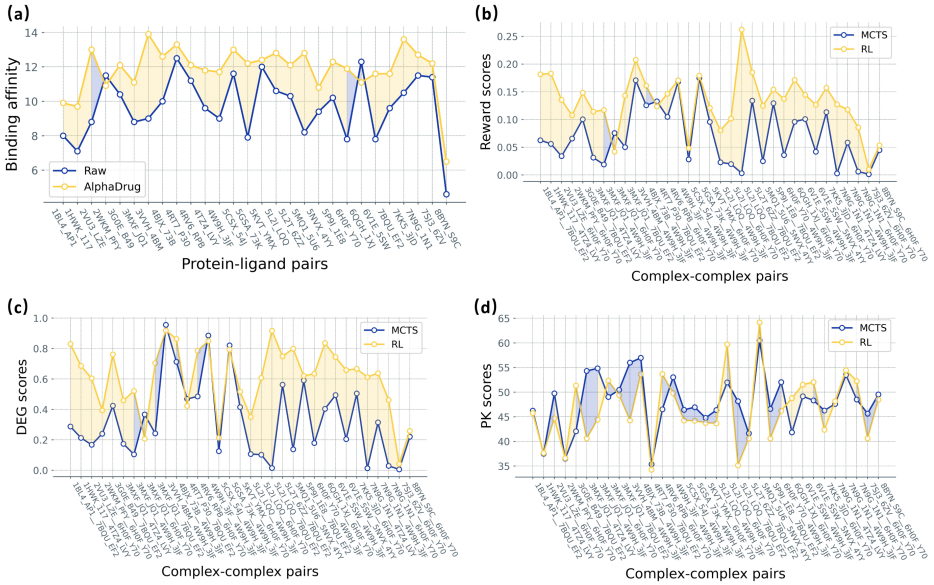

Fig. 8. Results of de novo designs of PROTACs. (a) Comparison between the binding affinity of the generated ligands and the original ligands. (b-d) The distribution of three scores (reward score, DEG score, and PK score) for PROTACs generated by AlphaDrug and TRL-PROTAC

6. Potential applications

In this section, we will discuss the potential applications of the TRL-PROTAC model. Firstly, when designing PROTACs starting from known warheads and E3 ligands, the TRL-PROTAC model demonstrates superior performance compared to other methods

(Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model), yielding more favorable PK and DEG properties. Therefore, the TRL-PROTAC model holds significant promise in discovering PROTACs drugs with enhanced attributes. Secondly, in comparison to the original PROTACs, PROTACs generated by the TRL-PROTAC model also exhibit superior characteristics, highlighting the model's potential in optimizing and replacing existing PROTACs medications. Finally, when designing PROTACs targeting specific proteins (considering binding affinity), the TRL-PROTAC model outperforms the MCTS method, showcasing its superiority in molecular de novo design.

7. Conclusion

Previous studies on generating PROTACs really did a good job, but they only considered a single property, such as PK, and then used a classification model to calculate the degradation and filter the generated PROTACs. This approach is both time-consuming and unable to generate valid PROTACs for complex inputs. We have innovatively proposed a de novo design strategy for PROTACs. Our model not only coordinates the PK property and the DEG property of PROTACs during the generation process but also avoids the issue of greedy algorithms not considering long-term benefits. Experimental results show that nearly all of the PROTACs designed using our model have better degradation and most of the molecules have better PK property than the original PROTACs, indicating the feasibility of our model. Compared to Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model, our model generates PROTACs with better properties, demonstrating the superiority of TRL-PROTAC in PROTAC generation. Furthermore, the TRL-PROTAC method enables the design of PROTACs targeting specific proteins, offering the advantage of further considering the binding affinity between PROTACs and the target proteins. To sum up, our strategy takes into account the integration of binding affinity, degradation, and pharmacokinetic properties for PROTACs.

Compared to traditional small molecules, the complexity of PROTACs design lies in their ternary structure. Naturally, two approaches can be considered for generating PROTACs. One is to directly generate a complete PROTAC molecule and then divide it into warhead, E3 ligand, and linker parts. Subsequently, the interaction between the warhead and E3 ligand with the respective target proteins is checked. However, the challenge with this approach is how to divide the PROTAC molecule and validate its ability to interact with the target protein after division. The other approach involves designing warheads and E3 ligands separately for the target proteins and then using a deep learning model to generate the linker that connects these two parts and optimizes the entire PROTAC. The difficulty here lies in how to connect these two components effectively. TRL-PROTAC follows the latter approach and provides a novel solution to these challenges, making it a remarkable exploration of PROTAC generation from scratch.

While our approach allows for PROTAC design against target proteins, it still has limitations. It can only be applied to design PROTACs for protein targets with existing ligand interactions and cannot be used for protein targets with no prior ligand interactions. Additionally, the warheads and E3 ligands generated by the AlphaDrug model, while exhibiting a high binding affinity to proteins, contain numerous heterocyclic structures, resulting in elevated PK properties. Further investigations will focus on exploring methodologies to

reduce the structural complexity of the ligands while preserving their heightened binding affinities to target proteins.

References

1. Guenette, R.G., Yang, S.W., Min, J., Pei, B., Potts, P.R.: Target and tissue selectivity of protac degraders. Chemical Society Reviews (2022)
2. Guedeney, N., Cornu, M., Schwalen, F., Kieffer, C., Voisin-Chiret, A.S.: Protac technology: A new drug design for chemical biology with many challenges in drug discovery. Drug Discovery Today 28(1), 103395 (2023)
3. Chen, Y., Tandon, I., Heelan, W., Wang, Y., Tang, W., Hu, Q.: Proteolysis-targeting chimera (protac) delivery system: Advancing protein degraders towards clinical translation. Chemical Society Reviews 51(13), 5330–5350 (2022)
4. Cao, C., He, M., Wang, L., He, Y., Rao, Y.: Chemistries of bifunctional protac degraders. Chemical Society Reviews (2022)
5. Chirnomas, D., Hornberger, K.R., Crews, C.M.: Protein degraders enter the clinic—a new approach to cancer therapy. Nature Reviews Clinical Oncology 20(4), 265–278 (2023)
6. Ocaña, A., Pandiella, A.: Proteolysis targeting chimeras (protacs) in cancer therapy. Journal of Experimental & Clinical Cancer Research 39(1), 189 (2020)
7. Zhou, Q.Q., Xiao, H.T., Yang, F., Wang, Y.D., Li, P., Zheng, Z.G.: Advancing targeted protein degradation for metabolic diseases therapy. Pharmacological Research p. 106627 (2022)
8. Li, F., Hu, Q., Zhang, X., Sun, R., Liu, Z., Wu, S., Tian, S., Ma, X., Dai, Z., Yang, X., et al.: Deepprotacs is a deep learning-based targeted degradation predictor for protacs. Nature Communications 13(1), 7133 (2022)
9. Yang, J., Li, Z., Wu, W.K.K., Yu, S., Xu, Z., Chu, Q., Zhang, Q.: Deep learning identifies explainable reasoning paths of mechanism of action for drug repurposing from multilayer biological network. Briefings in Bioinformatics 23(6), bbac469 (2022)
10. Baptista, D., Ferreira, P.G., Rocha, M.: Deep learning for drug response prediction in cancer. Briefings in bioinformatics 22(1), 360–379 (2021)
11. Krentzel, D., Shorte, S.L., Zimmer, C.: Deep learning in image-based phenotypic drug discovery. Trends in Cell Biology (2023)
12. Deng, J., Yang, Z., Ojima, I., Samaras, D., Wang, F.: Artificial intelligence in drug discovery: applications and techniques. Briefings in Bioinformatics 23(1) (2022)
13. Shao, D., Dai, Y., Li, N., Cao, X., Zhao, W., Cheng, L., Rong, Z., Huang, L., Wang, Y., Zhao, J.: Artificial intelligence in clinical research of cancers. Briefings in Bioinformatics 23(1), bbab523 (2022)
14. Lv, H., Shi, L., Berkenpas, J.W., Dao, F.Y., Zulfiqar, H., Ding, H., Zhang, Y., Yang, L., Cao, R.: Application of artificial intelligence and machine learning for covid-19 drug discovery and vaccine design. Briefings in Bioinformatics 22(6), bbab320 (2021)
15. Zeng, X., Wang, F., Luo, Y., Kang, S.g., Tang, J., Lightstone, F.C., Fang, E.F., Cornell, W., Nussinov, R., Cheng, F.: Deep generative molecular design reshapes drug discovery. Cell Reports Medicine p. 100794 (2022)
16. Cerchia, C., Lavecchia, A.: New avenues in artificial-intelligence-assisted drug discovery. Drug Discovery Today p. 103516 (2023)
17. Ma, X.H., Jia, J., Zhu, F., Xue, Y., Li, Z.R., Chen, Y.Z.: Comparative analysis of machine learning methods in ligand-based virtual screening of large compound libraries. Combinatorial chemistry & high throughput screening 12(4), 344–357 (2009)
18. Ye, Q., Hsieh, C.Y., Yang, Z., Kang, Y., Chen, J., Cao, D., He, S., Hou, T.: A unified drug–target interaction prediction framework based on knowledge graph and recommendation system. Nature communications 12(1), 6775 (2021)

19. Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.C., et al.: ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 34(14), 2499–2502 (2018)
20. Qian, H., Lin, C., Zhao, D., Tu, S., Xu, L.: Alphadrug: protein target specific de novo molecular generation. PNAS Nexus 1(4), pgac227 (2022)
21. Olivecrona, M., Blaschke, T., Engkvist, O., Chen, H.: Molecular de-novo design through deep reinforcement learning. Journal of cheminformatics 9(1), 1–14 (2017)
22. Yi, Z., Li, S., Yu, J., Tan, Y., Wu, Q., Yuan, H., Wang, T.: Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In: Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings 13. pp. 554–566. Springer (2017)
23. Ghaffari, A., Abdollahi, H., Khoshayand, M., Bozchalooi, I.S., Dadgar, A., Rafiee-Tehrani, M.: Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. International journal of pharmaceutics 327(1-2), 126–138 (2006)
24. Karim, M.R., Cochez, M., Jares, J.B., Uddin, M., Beyan, O., Decker, S.: Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. In: Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics. pp. 113–123 (2019)
25. Huang, K., Xiao, C., Glass, L.M., Sun, J.: Moltrans: molecular interaction transformer for drug–target interaction prediction. Bioinformatics 37(6), 830–836 (2021)
26. Zheng, S., Tan, Y., Wang, Z., Li, C., Zhang, Z., Sang, X., Chen, H., Yang, Y.: Accelerated rational protac design via deep learning and molecular simulations. Nature Machine Intelligence 4(9), 739–748 (2022)
27. Yang, Y., Zheng, S., Su, S., Zhao, C., Xu, J., Chen, H.: Syntalinker: automatic fragment linking with deep conditional transformer neural networks. Chemical science 11(31), 8312–8322 (2020)
28. Tolpin, D., Shimony, S.: Mcts based on simple regret. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 26, pp. 570–576 (2012)
29. Perez, D., Samothrakis, S., Lucas, S.: Knowledge-based fast evolutionary mcts for general video game playing. In: 2014 IEEE Conference on Computational Intelligence and Games. pp. 1–8. IEEE (2014)
30. Rosin, C.D.: Multi-armed bandits with episode context. Annals of Mathematics and Artificial Intelligence 61(3), 203–230 (2011)
31. Grechishnikova, D.: Transformer neural network for protein-specific de novo drug generation as a machine translation problem. Scientific reports 11(1), 321 (2021)
32. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2019)
33. Landrum, G., et al.: Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum 8 (2013)
34. DeGoey, D.A., Chen, H.J., Cox, P.B., Wendt, M.D.: Beyond the rule of 5: lessons learned from abbvie's drugs and compound collection: miniperspective. Journal of Medicinal Chemistry 61(7), 2636–2651 (2017)
35. Sewak, M., Sewak, M.: Policy-based reinforcement learning approaches: Stochastic policy gradient and the reinforce algorithm. Deep Reinforcement Learning: Frontiers of Artificial Intelligence pp. 127–140 (2019)
36. Zhang, J., Koppel, A., Bedi, A.S., Szepesvari, C., Wang, M.: Variational policy gradient method for reinforcement learning with general utilities. Advances in Neural Information Processing Systems 33, 4572–4583 (2020)
37. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
38. Alegre, L.N., Bazzan, A.L., et al.: Knowledge transfer in multi-objective multi-agent reinforcement learning via generalized policy improvement. Computer Science and Information Systems (00), 71–71 (2023)

39. Fu, Q., Kang, Y., Gao, Z., Wu, H., Hu, F., Chen, J., Zhong, S.: Multi-threading parallel reinforcement learning. International Journal of Computer Applications in Technology 61(4), 278–286 (2019)
40. Weng, G., Shen, C., Cao, D., Gao, J., Dong, X., He, Q., Yang, B., Li, D., Wu, J., Hou, T.: Protac-db: an online database of protacs. Nucleic acids research 49(D1), D1381–D1387 (2021)
41. Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al.: Chembl: a large-scale bioactivity database for drug discovery. Nucleic acids research 40(D1), D1100–D1107 (2012)
42. Dimitropoulos, D., Ionides, J., Henrick, K.: Using msdchem to search the pdb ligand dictionary. Current protocols in bioinformatics 15(1), 14–3 (2006)
43. Golovin, A., Dimitropoulos, D., Oldfield, T., Rachedi, A., Henrick, K.: Msdsite: a database search and retrieval system for the analysis and viewing of bound ligands and active sites. Proteins: Structure, Function, and Bioinformatics 58(1), 190–199 (2005)
44. Golovin, A., Oldfield, T.J., Tate, J.G., Velankar, S., Barton, G.J., Boutselakis, H., Dimitropoulos, D., Fillon, J., Hussain, A., Ionides, J.M., et al.: E-msd: an integrated data resource for bioinformatics. Nucleic Acids Research 32(suppl_1), D211–D216 (2004)
45. Consortium, U.: Uniprot: a hub for protein information. Nucleic acids research 43(D1), D204–D212 (2015)
46. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic acids research 28(1), 235–242 (2000)
47. Bakan, A., Meireles, L.M., Bahar, I.: Prody: protein dynamics inferred from theory and experiments. Bioinformatics 27(11), 1575–1577 (2011)
48. O'Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R.: Open babel: An open chemical toolbox. Journal of cheminformatics 3(1), 1–14 (2011)
49. Yin, L.: Cmplot: circle manhattan plot. r package version 3.6 (2020)
50. Wang, J., Yu, J., Lipka, A.E., Zhang, Z.: Interpretation of manhattan plots and other outputs of genome-wide association studies. In: Genome-Wide Association Studies, pp. 63–80. Springer (2022)
51. Wickham, H., Wickham, H.: Data analysis. ggplot2: elegant graphics for data analysis pp. 189–201 (2016)

Yuhao Dai is a graduate student at School of Computer Science and Technology, Soochow University. He is mainly engaged in deep generative models, bioinformatics and molecular generation.

Fei Zhu is an associate professor at School of Computer Science and Technology, Soochow University. He is a senior member of China Computer Federation. His main research interests include deep learning, reinforcement learning, protein interaction, and pattern recognition.

Received: March 27, 2024; Accepted: June 10, 2024.