DRUGIMPROVER: Utilizing Reinforcement Learning for Multi-Objective Alignment in Drug Optimization

Xuefeng Liu[1]*, Songhao Jiang[1], Archit Vasan[2], Alexander Brace[1,2], Ozan Gokdemir[1]
Thomas Brettin[2], Fangfang Xia[2], Ian T. Foster[1,2], Rick L. Stevens[1,2]
[1]Department of Computer Science, University of Chicago
[2]Argonne National Laboratory

Abstract

Reinforcement learning from human feedback (RLHF) is a method for enhancing the finetuning of large language models (LLMs), leading to notable performance improvements that can also align better with human values. Building upon the inspiration drawn from RLHF, this research delves into the realm of drug optimization. We employ reinforcement learning to finetune a drug optimization model, enhancing the original drug across multiple target objectives, while retains the beneficial chemical properties of the original drug. Our proposal comprises three primary components: (1) DRUGIMPROVER: A framework tailored for improving robustness and efficiency in drug optimization. (2) A novel Advantage-alignment Policy Optimization (APO) with multi-critic guided exploration algorithm for finetuning the objective-oriented properties. (3) A dataset of 1 million compounds, each with OEDOCK docking scores on 5 human proteins associated with cancer cells and 24 proteins from SARS-CoV-2 virus. We conduct a comprehensive evaluation of APO and demonstrate its effectiveness in improving the original drug across multiple properties. Our code and dataset are made public at: https://github.com/xuefeng-cs/DrugImprover.

1 Introduction

The cost of discovering a new drug through conventional approaches is estimated to range from hundreds of millions to billions of dollars [17]. This high cost is due to the lengthy and resource-intensive nature of the drug discovery and development process, which involves multiple stages, including target identification, lead compound identification, preclinical testing, and clinical trials. Despite significant efforts, the overall success rate in drug discovery is relatively low, with many drug candidates failing to progress beyond the early stages of development. Additionally, the time required to identify an effective drug can vary from several years to over a decade, depending on the complexity of the disease and the efficiency of the drug discovery process. Such concerns are driving a growing trend towards drug repurposing [3], which involves using FDA-approved drugs for different diseases instead of developing new drugs from the ground up. Yet despite some successes [48], the effectiveness of drug repurposing has been limited since the drug is usually designed very specifically for treating a particular disease. However, the emergence of rapidly evolving virus variants [24], such as those associated with SARS-CoV-2 [75], as well as drug resistant cancer cells [39] has sparked increased interest and an urgent need to expedite the discovery of effective drugs.

In this work, we propose a reinforcement learning (RL)-based drug optimization approach to adapt existing drugs to fast-evolving virus variants and cancer cells, helping to address the aforementioned limitations of drug discovery and drug repurposing. RL has achieved superhuman performance in

*Correspondence to: Xuefeng Liu <xuefeng@uchicago.edu>.

The NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development (AI4D3 2023), New Orleans, LA, USA, 2023.

domains such as Chess [31], video games [41], and Robotics [12]. However, despite promising early results [10, 23, 29, 42, 61, 76], RL has yet to attain similar levels of performance for complex real-life problems like drug discovery.

We identified four challenges that have thus far prevented RL from impacting drug design: *Search space complexity*: An RL algorithm for drug discovery needs to demonstrate both sample and computational efficiency. However, the overwhelming complexity of the search space [45] renders RL incapable of adequately exploring potential effective actions and states required for policy learning. *Sparse rewards*: In contrast to the continuous reward environment found in popular environments like DeepMind Control Suite [62] or Meta-World [73], drug generation operates within a sparse reward environment where rewards are only obtainable upon a complete molecule. *Complex scoring criteria*: Generated molecules must fulfill multiple criteria, including solubility and synthesizability, while also achieving a high docking score when targeting a specific site. *Preservation of original beneficial properties*: Lastly, as drugs with similar chemical structures should exhibit similar biological/chemical effects [8], it is crucial to strike a balance between optimizing the drug and preserving the original drug's beneficial properties.

Our contributions. We present DRUGIMPROVER, a drug optimization framework designed to improve various properties of an original drug in a robust and efficient manner. Within this workflow, we introduce the Advantage-alignment Policy Optimization (APO) algorithm to utilize the advantage preference to perform direct policy improvement under the guidance of multiple critics. DRUGIMPROVER and APO effectively tackle the challenges outlined above in the following manner:

(1.) *Sample complexity, sparsity, and computational efficiency.* Because of the sparse reward nature of the drug design, pure RL often finds it challenging to learn a good policy due to the complexity of the search space. To reduce this complexity, APO employs an imitation-learning-based approach to initialize a generator policy with desirable behavior based on prior experience of designing drug SMILES [67] strings. APO also addresses the problem of reward sparsity by adapting Monte-Carlo sampling to obtain estimated rewards for intermediate steps. Finally, because calculating the docking score through virtual screening (such as OEDOCK [30]) is computationally costly [16], DRUGIMPROVER adopts a transformer-based surrogate model to obtain docking scores more efficiently. (2.) *Multiple objectives.* APO employs multiple critics, each of which serves as an evaluator with domain-specific expertise, such as knowledge related to solubility, synthesizability, and other relevant factors. These multiple critics guide the exploration in the drug refinement process toward the improved properties. (3.) *Property preserving.* To preserve the original drug's beneficial properties, throughout the optimization process, it is crucial to balance the preservation of the original drug's beneficial properties with the optimization of other chemical attributes. To achieve this, we use Tanimoto similarity as a critic to maximize the Tanimoto similarity between the original and generated drugs. (4.) *Finetuning.* Our proposed APO algorithm utilizes the advantage preference of a generated drug over the original drug based on multiple objectives as the policy gradient signal and performs direct policy improvement without the need for training an additional reward model.

In summary, our contributions are:

• We introduce DRUGIMPROVER, a framework tailored for efficient drug design. Within DRUGIMPROVER, we propose a novel APO algorithm that performs advantage-alignment policy optimization with multi-critic guided exploration.

• By conducting comprehensive experiments on real world viral and cancer target proteins, we illustrate that APO consistently enhances existing molecules/drugs across all desired objectives, leading to improved drug candidates.

• We release a drug optimization dataset comprising 1 million ligands along with their OEDOCK scores to five proteins associated with cancer: colony stimulating factor 1 receptor (CSF1R) kinase domain (PDB ID: 6T2W), NOP2/Sun RNA methyltransferase 2 (NSUN2) (AlphaFold derived), RNA terminal phosphate cyclase B (RTCB) ligase (PDB ID: 7P3B), and Tet methylcytosine dioxygenase 1 (TET1) (AlphaFold derived), and Wolf-Hirschhorn syndrome candidate 1 (WHSC1) (PDB ID: 7MDN) and 24 high-affinity binding sites on protein SARS-CoV-2: 3CLPro (PDBID: 7BQY) virus.

2 Related Work

2.1 Imitation learning

Imitation learning (IL) is a machine learning technique whereby an agent learns to perform a task by mimicking the actions and behaviors of an expert demonstrator. IL has demonstrated its advantage over pure RL in improving the sample complexity of search space and reward sparsity [38]. Offline IL methods, such as behavioral cloning [46], necessitate an offline dataset of trajectories collected from one or more experts, which can result in cascading errors within the learner's policy. In contrast, interactive IL methods, exemplified by DAgger [52] and AggreVaTe [51], employ Roll-in-Roll-out (RIRO) scheduling, which assumes that the learner starts with a default roll-in from the initial state and then actively switches to an expert to roll out for the remaining steps in the trajectory. However, previous RIRO scheduling work assumed that an expert is readily available to conduct roll-outs from any given state. In practice, such an expert may not always be accessible. Additionally, it assumed that once a roll-out has commenced, it cannot return to a previous intermediate state. In our work, we combine RIRO from interactive IL with Monte Carlo sampling, establishing both a learner policy π_θ^G and a guide policy π^β. Importantly, our guide policy may be identical to the learner's policy, serving the dual purpose of conducting roll-outs and estimating the returns of intermediate states through Monte Carlo sampling.

2.2 Learning from multiple experts

IL often assumes the presence of a near-optimal expert to imitate; however, accessing such an expert may be costly or even impossible. In reality, it is often easier to access suboptimal oracles, each with unique expertise. Therefore, learning from multiple experts becomes an increasingly important topic, although identifying state-wise expertise remains a challenge. Several methods, such as EXP3, EXP4 [2], and CAMS [35, 36], have approached the challenge of learning from multiple oracles by framing it as a contextual bandit or online learning problem. However, these techniques cannot address sequential decision-making problems like Markov decision processes (MDPs) because they lack the capability to incorporate state information. In the realms of RL and IL, MAMBA [13], MAPS [38], and RPI [37] have explored IL from multiple experts. While MAMBA selects an oracle to query at random, which compromises its sample efficiency, MAPS introduces active policy selection and active state exploration for settings with multiple experts, enhancing the sample efficiency of the learning process. In contrast to previous approaches, our work takes a different approach. Instead of selecting a specific expert to imitate, we treat each expert as a domain-specific reward function and utilize an ensemble of these reward functions as critics to guide the learner's policy towards exploration in molecular space.

2.3 Reinforcement learning for molecule generation

One prominent approach in drug design employs RL [60] to maximize an expected reward defined as the sum of predicted property scores as generated by property predictors. In terms of representation, existing works in RL for drug design have predominantly operated on SMILES string representations [10, 23, 42, 43, 47, 56, 61, 76, 77] or graph-based representations [1, 21, 29, 71]. In our research, we have chosen to employ the SMILES representation. However, previous studies have primarily focused on discovering new drugs, frequently overlooking molecular structure constraints during policy improvement. This oversight can lead to drastic changes in structure or functional groups, making most of the generated compounds unsynthesizable. In contrast, our work concentrates on optimizing existing drugs while preserving their beneficial properties, rather than creating entirely new ones from scratch.

2.4 RL finetuning and AI alignment

To achieve drug improvement, finetuning the generator model is critical. Our approach is closely connected to prior methodologies aimed at aligning models with feedback from both humans (RLHF [6, 14, 26, 63]) and AI (RLAIF [7, 34]), which have more recently found applications in the fine-tuning of language models for tasks like text summarization [9, 58, 69, 78], dialogue generation [25, 28, 70], and language assistance [6]. One of the core features of RLHF and RLAIF lies in training a reward model from the comparison feedback, such as Rank Responses to align

Human Feedback (RRHF) [74], Reward Ranked Finetuning (RAFT) [18], Preference Ranking Optimization (PRO) [55], and Direct Preference Optimization (DPO) [49]. Differing from previous works, our approach doesn't rely solely on feedback from a single human or AI model; instead, we utilize multiple critics to evaluate the advantage preference of generated drug against original one based on comprehensive assessments, including factors like solubility, and more. Moreover, we make direct policy improvement by using the advantage preference in standard RL without training an additional reward model.

3 Preliminaries

Markov decision process. We consider a finite-horizon Markov Decision Process (MDP) $\mathcal{M}_0 = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, T \rangle$ with state space $\mathcal{S}$, action space $\mathcal{A}$, deterministic transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}'$, unknown reward function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and horizon T. We assume access to a set of K critics each represents a domain experts, defined as $\mathbf{C} = \left\{C^k\right\}_{k=1}^{\mathcal{K}}$, where $C : s_T \to \mathbb{R}$ and s_T represents a final state. The policy $\pi : \mathcal{S} \to \mathcal{A}$ maps the current state to a distribution over actions. Given an initial state distribution $\rho_0 \in \Delta(\mathcal{S})$, we define d_t^π as the distribution over states at time t under policy π. The goal is to train a policy to maximize the expected long-term reward. The quality of the policy can be measured by the Q-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as:

$$Q^\pi(s, a) := \mathbb{E}^\pi \left[\sum_{t=0}^{T} R(s_t, a_t) \, | s_0 = s, a_0 = a \right], \tag{1}$$

where the expectation is taken over the trajectory following π, and the value function is as follows:

$$V^\pi(s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s, a)]. \tag{2}$$

Drug generation process. We formalize the drug generation problem within the framework of Markov Decision Processes (MDP). Given a dataset consisting of real-world structured sequences represented as SMILES [67] strings, our objective is to train a generative policy π_θ^G to generate a high-quality sequence denoted as $Y_{1:T} = (y_1, \ldots, y_t, \ldots, y_T), y_t \in \mathcal{Y}$. Here, $\mathcal{Y}$ represents the vocabulary of potential SMILES tokens, constituting the action space denoted as $\mathcal{A}$. The length of the sequence, denoted as T, represents the planning horizon. At time step t, the state s_{t-1} comprises the currently generated tokens $(y_1, \ldots, y_{t-1})$, and the action a corresponds to the next token y_t to be selected. While the policy model $\pi_\theta^G(y_t | Y_{1:t-1})$ operates in a stochastic manner, the state transition function $\mathcal{P}$ becomes deterministic once an action has been chosen. The primary objective of the generator policy π_θ^G is to initiate the generation process from an initial state Y_1 and maximize the expected final reward at the end of the sequence:

$$J(\theta) = \mathbb{E}_{Y_1 \sim d_0^{\pi_\theta^G}}[r_T | \theta], \tag{3}$$

where r_T represents the reward associated with a fully generated sequence. To estimate the Q value, we reference the REINFORCE algorithm [68], which we define as follows:

$$Q(s = Y_{1:T-1}, a = y_T) = R(Y_{1:T}). \tag{4}$$

Nonetheless, the reward function only supports a reward value for a completed sequence. In our case, we aim to compute the Q for partial sequences at intermediate time steps, accounting for the expected future reward upon sequence completion. To achieve this, we employ a Monte Carlo search approach and Roll-in-Roll-out (RIRO) [13, 38, 51] scheduling, utilizing a roll-out policy denoted as π_β to sample the unknown last $T - t$ tokens. We represent an N-time Monte Carlo search as follows:

$$\left\{Y_{1:T}^1, ..., Y_{1:T}^N\right\} = MC^{\pi_\beta}(Y_{1:t}; N), \tag{5}$$

where $Y_{t+1:T}^N$ is sampled based on the roll-out policy π_β and the current state $Y_{1:t}^n$ is stochastically sampled via the roll-in policy π_θ^G. In our experiment, we set π_β to be identical to the learner policy π_θ^G, although it can alternatively be an oracle policy if one is accessible. To enhance the precision of expected Q value assessment, we execute the roll-out policy from the current state to the end of the sequence N times and estimate its averaged rewards on a batch of complete samples. Thus:

$$Q(s = Y_{1:t-1}, a = y_t) = \begin{cases} \frac{1}{N} \sum_{n=1}^{N} R(Y_{1:T}^n), \text{where } Y_{1:T}^n \in MC^{\pi_\theta^G}(Y_{1:t}; N), \text{if t} < T, \\ R(Y_{1:t}), \text{if t} = T. \end{cases} \tag{6}$$

Algorithm 1 Advantage-alignment policy optimization with multi-critic guided exploration

Require: generator policy π_θ^G; roll-out policy π_β; a pre-train dataset $\mathcal{B}$, critics $\mathbf{C}$ with weights $\mathbf{W}$.

Initialize π_θ^G with random weight θ.
Pre-train π_θ^G usng MLE on $\mathcal{B}$.
$\beta \leftarrow \theta$.
for $n = 1, \ldots, N$ **do**
    $s_0 \sim \rho_0$, where $\rho_0 \in \Delta(\mathcal{B})$.
    Generate a sequence $Y_{1:T} = (y_t, \ldots, y_T) \sim \pi_\theta^G(\cdot|s_0)$.
    Compute advantage preference $R^{\text{Advantage-Preference}}$ by (6)(9)(13)(14).
    Update generator parameters via policy gradient by (16)(17).
    $\beta \leftarrow \theta$.

Here, $Q^{\pi_\theta^G}(s, a)$ stands for the action-value function, which represents the expected reward at state s of taking action $a \sim \pi_\theta^G(s)$ and following the current policy π_θ^G to complete the sequence. Policy gradient optimizes a parameterized policy to maximize the expected total reward by repeatedly estimating the gradient $g := \nabla_\theta J(\theta)$. There is a general form for the policy gradient [53, 59]:

$$g = \sum_{t=1}^{T} \mathbb{E}_{y_t \sim \pi_\theta^G(y_t|Y_{1:t-1})} \left[\nabla_\theta \log \pi_\theta^G(y_t|Y_{1:t-1}) \cdot \Phi_t \right], \tag{7}$$

where Φ_t could be in several forms. One common choice for Φ_t in previous drug discovery work is $Q(Y_{1:t-1}, y_t)$ [23, 72].

Limitations of previous work. 1) Prior studies concentrated primarily on the discovery of new drugs from the ground up [1, 47, 76]. In contrast, we focus on the relatively less explored, yet highly practical and significant, issue of drug optimization. In drug optimization, the goal is to enhance an existing drug according to multiple objectives while preserving a similar chemical structure. 2) Earlier research employed Q [23, 72] in gradient calculations, which can introduce high variance and potentially lead to divergence. Our advantage-alignment policy gradient approach avoids this problem.

4 DRUGIMPROVER Framework for Drug Optimization

In this work, we propose DRUGIMPROVER framework as in Fig. 1, which comprises two major components: (1) An Advantage-alignment Policy Optimization with multi-critic guided exploration algorithm (APO), and (2) A dedicated workflow tailored for drug optimization, aimed at enhancing both robustness and computational efficiency. We introduce each part in detail as follows.

4.1 Advantage-alignment policy optimization with multi-critic guidance algorithm

Multi-critic guidance. Given an ensemble of critics

$$\mathbf{C}(s_0, s_T) = \left[C^{\text{Druglikeness}}(s_T), C^{\text{Solubility}}(s_T), C^{\text{Synthesizability}}(s_T), C^{\text{Docking}}(s_T), C^{\text{Tanimoto}}(s_0, s_T) \right],$$

where $C : Y_{1:T} \to \mathbb{R}$. Here we design the reward function to align the drug optimization with multiple objectives. Also, we need to preset a weight array over the objectives,

$$\mathbf{W} = \left[W^{\text{Druglikeness}}, W^{\text{Solubility}}, W^{\text{Synthesizability}}, -1 \cdot W^{\text{Docking}}, W^{\text{Tanimoto}} \right]. \tag{8}$$

The weights represent the importance of each objective. For a fully generated SMILE sequence, we derive the following multi-step accumulated reward function based on assessments from multiple critics

$$R_c(Y_{1:T}) := R_c(Y_{1:T}|s_0) = \sum_{t=1}^{T} \sum_{n=1}^{N} \text{Norm}\left(\mathbf{C}(s_0, Y_{1:T}^n)\right) \cdot \mathbf{W}, \quad Y_{1:T}^n \in MC^{\pi_\theta^G}(Y_{1:t}; N). \tag{9}$$

We use Norm[2] to normalize different attributes onto the same scale. In this study, we employ the Tanimoto similarity calculation $C^{\text{Tani-Similarity}}$ to quantify the chemical similarity between the

[2]Here, we define Norm as min-max normalization to scale the attributes onto the range [-10, 10].

generated compound and the original drug. Essentially, this calculation involves first computing Morgan Fingerprints [50] for each molecule and then measuring the Jaccard distance [27] (i.e., intersection over union) between the two fingerprints.

Advantage-alignment policy gradient. The return, denoted as Q^{π}, often exhibits significant variance across multiple episodes. One approach to mitigate this issue is to subtract a baseline $b(s)$ from each Q. The baseline function can be any function, provided that it remains invariant with respect to a. For a generator policy π_θ^G, the advantage function [59] is defined as follows:

$$\boldsymbol{A}^{\pi_\theta^G}(s,a) = Q^{\pi_\theta^G}(s,a) - b(s) \tag{10}$$

A natural choice for the baseline is the value function $V^{\pi}(s)$, which represents the expected reward at a given state s under policy π. The value function can be expressed as follows:

$$V(s) = \mathbb{E}_{a\sim\pi_\theta^G(s)}[Q(s,a)] = \mathbb{E}_{y_t\sim\pi_\theta^G(Y_{1:t-1})}[Q(Y_{1:t-1}, y_t)] \tag{11}$$

Thus, we have advantage function as

$$\boldsymbol{A}^{\pi_\theta^G}(s,a) = \boldsymbol{A}^{\pi_\theta^G}(Y_{1:t-1}, y_t) = Q^{\pi_\theta^G}(Y_{1:t-1}, y_t) - V^{\pi_\theta^G}(Y_{1:t-1}). \tag{12}$$

Remark 4.1. When compared to $Q^{\pi_\theta^G}$ as described in (7), the selection of $\boldsymbol{A}^{\pi_\theta^G}$ tends to result in potentially lower variance. This assertion can be intuitively justified by considering the interpretation of the policy gradient: the direction of a step in the policy gradient should increase the probability of actions better than the average and decrease the probability of actions worse than the average. The advantage function essentially gauges whether an action is superior or inferior to the policy's default behavior. Consequently, we opt to designate Φ_t as the advantage function $\boldsymbol{A}^{\pi_\theta^G}$. This choice ensures that the gradient term $\Phi_t \nabla_\theta \log \pi_\theta^G(a_t|s_t)$ aligns with an increase in $\pi_\theta^G(a_t|s_t)$ only when $\boldsymbol{A}^{\pi_\theta^G}(a,s) > 0$. This is contrast with previous method using Q. For a more thorough examination of the variance of policy gradient estimators and the impact of employing baseline, please refer to [22].

Drug optimization. In the drug optimization problem, the primary objective is different from objective of drug generation problem in (3). In this work, we employ the one-step RL [11, 44] method and regard the drug optimization method as a sequence to sequence language generation task. Rather than treating each token as an individual action, we treat the entire sequence $Y_{1:T}$ as a single action generated by the policy π_θ^G. Subsequently, we receive rewards from critics, and the episode concludes. This leads to the formulation of our advantage function as follows:

$$\boldsymbol{A}^{\pi_\theta^G}(s,a) = Q^{\pi_\theta^G}(Y_{1:t-1}, y_t)\,|_{t=T} - V^{\pi_\theta^G}(s_0) = R_c(Y_{1:T}) - R_c(s_0), \tag{13}$$

where s_0 is the initial state sequence drawn from the distribution ρ_0, which corresponds to our buffer known as $\mathcal{B}$ containing selected SMILES strings. Thus, by applying amplifier γ, the advantage preference of the generated versus the original drug is

$$R^{\text{Advantage-Preference}}(s_0, Y_{1:T}) = \gamma_n (R_c(Y_{1:T}) - R_c(s_0)), \tag{14}$$

where $\gamma_n \in \mathbb{R}^+$ represents an amplifier of advantage preference at n-th episode, $n \in [N]$, that controls the aggressiveness or conservatism in performing policy gradient updates. The advantage preference of (14) will be employed directly in the policy gradient (16) to finetune the generator policy π_θ^G. The rationale behind the advantage preference is to produce a sequence that surpasses the initial state sequence s_0 in every objective. In this work, our objective is to maximize the expected final advantage preference compared to the original drug s_0 at the end of the sequence as follows

$$J(\theta) = \mathbb{E}_{s_0\sim\rho_0}\left[R_T^{\text{Advantage-Preference}}|s_0, \theta\right], \tag{15}$$

Thus, we have gradient as follows:

$$g = \mathbb{E}_{s_0\sim\rho_0}\left[\sum_{Y_{1:T}\sim\pi_\theta^G(\cdot|s_0)} \nabla_\theta \log \pi_\theta^G(Y_{1:T}|s_0) \cdot R^{\text{Advantage-Preference}}(s_0, Y_{1:T})\right], \tag{16}$$

where $Y_{1:T}$ is the generated sequence from π_θ^G and s_0 is the original drug. As the expectation $\mathbb{E}[\cdot]$ can be approximated through sampling techniques, we proceed to update the generator's parameters as follows:

$$\theta \leftarrow \theta + \alpha_n g, \tag{17}$$

where $\alpha \in \mathbb{R}^+$ denotes the learning rate at n-th episode.

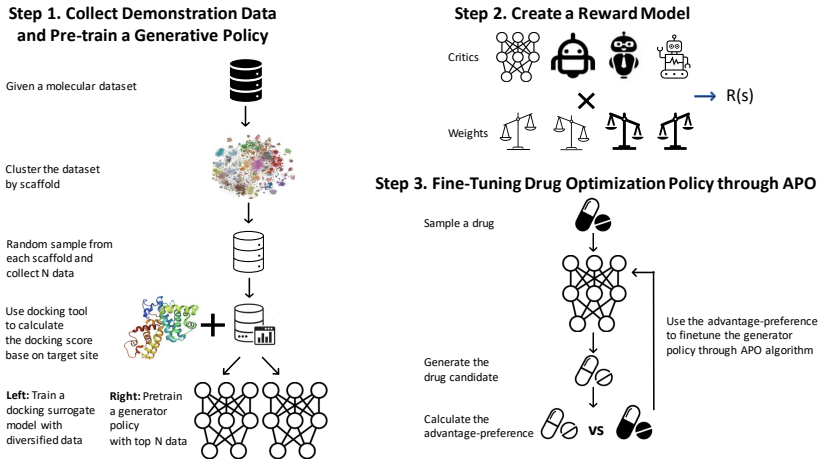


Figure 1: DRUGIMPROVER framework. It comprises two major components: (1) An Advantage-alignment Policy Optimization with multi-critic guided exploration algorithm (APO). (2) A workflow tailored for drug optimization, aimed at enhancing both robustness and computational efficiency.

4.2 DRUGIMPROVER framework

Step 1: Training a docking surrogate model $C^{\text{surrogate}}$. To begin, we perform a scaffold-based clustering procedure on 2 million drug-like molecules selected from the ZINC15 dataset [57]. Subsequently, we conduct stratified sampling within these clusters, resulting in the acquisition of a chemically diverse subset comprising 1 million representative ligands. The next step involves employing OEDOCK, for which we developed a parallel workflow using Parsl [4] and Colmena [66] to leverage the supercomputing resources of Polaris at the Argonne Leadership Computing Facility (ALCF) [20] which calculates the docking scores of the ligands when interacting with the binding target site of SARS-CoV-2 and RTCB. Upon the completion of this extensive docking run, we employ both the molecules and their corresponding docking scores to train a docking surrogate [64], which we then employ as a critic. This critic takes ligands as input and generates estimated docking scores as output; its primary function is to guide the exploration process in drug design. We also apply the surrogate model to predict the docking scores for the remaining molecules within ZINC15. Informed by these predictions and the scaffold-based clusters, we proceed to sample 2 million molecules, which serve as the training data for our replay buffer denoted as $\mathcal{B}$. This optimization process seeks to strike a balance between achieving high docking scores and maintaining chemical diversity within the dataset.

Step 2: Pre-training a generator policy π_θ^G. Next, we perform random sampling of ligands from $\mathcal{B}$ based on each scaffold. Subsequently, we employ the sampled data to pre-train the π_θ^G policy using a self-supervised imitation learning approach. In this context, each ligand within $\mathcal{B}$ is considered a complete drug generation trajectory, comprising a sequence of states and actions. This pre-training procedure enables the π_θ^G policy to mimic the process of generating a high-quality ligand with a promising chemical structure. Analogous to how AlphaGo [54] learns from expert prior experiences, this approach reduces the non-trivial sample complexity when compared to training π_θ^G from scratch, which would necessitate extensive exploration efforts within the high-dimensional space.

Step 3: Performing objective-oriented policy finetuning: Finally, we fine tune the original drug based generator on the following objectives: (1) The docking surrogate model. (2) Solubility. (3) Synthetizability. (4) Tanimoto similarity to the initial molecule. Each objective serves as a domain-specific critic, with each critic individually specializing in and optimizing for a specific molecular property. These reward critics are tailored to optimize the learner policy π_θ^G with reward signals that align with their respective specialties. We balance these objectives by assigning different weight to each critic. Subsequently, we apply Algorithm 1 to finetune the learner policy π_θ^G for improving the drug optimization process.

Target site	Algorithm	Druglikeliness ↑		Synthesizability ↑		Solubility ↑		Docking score ↓		Tanimoto similarity ↑
3CLPro (PDBID: 7BQY)	**MLE**	0.14	(0%)	0.11	(0%)	0.10	(0%)	-1.48	(0%)	–
	ORGAN	0.37	(170%)	0.53	(368 %)	0.31	(207 %)	-4.32	(191 %)	–
	Naive RL	0.40	(198 %)	0.62	(441 %)	0.35	(251 %)	-4.96	(234 %)	–
	APO (Ours)	**0.45**	**(233%)**	**0.69**	**(506%)**	**0.40**	**(303 %)**	**-5.73**	**(286 %)**	0.959
RTCB (PDBID: 4DWQ)	**MLE**	0.14	(0%)	0.11	(0%)	0.10	(0%)	-1.61	(0%)	–
	ORGAN	0.39	(187 %)	0.64	(461 %)	0.35	(246 %)	-6.04	(274%)	–
	Naive RL	0.39	(185 %)	0.66	(478 %)	0.36	(260 %)	-6.61	(281 %)	–
	APO (Ours)	**0.46**	**(237%)**	**0.77**	**(577%)**	**0.42**	**(323%)**	**-6.98**	**(332%)**	0.940

Table 1: **Main results.** A comparison between three baselines {MLE, ORGAN, Naive RL} with APO on objectives {druglikeness, synthesizability, solubility, docking score, Tanimoto similarity} based on 3CLPro and RTCB datasets. The presented values represent the mean values of generated molecules and Tanimoto similarity is measured on valid molecules. Values displayed in bold indicate notable improvements, and the percentage of improvement over the MLE baselines is enclosed in parentheses.

5 EXPERIMENTS

5.1 Experiment Setup

Baselines and sequence generative model. In our experimental setup, we compare our approach against three representative baselines: Maximum Likelihood Estimation (MLE) and ORGAN [23] and Naive RL. For the MLE baseline, we utilize the pretrained LSTM-based generator π_θ^G, without proceeding further to finetune the model. ORGAN is a RL-based method for drug discovery, utilizing policy gradient based on the Q-value and employing a combination of a discriminator and domain objectives as rewards." Naive RL is using the same architecture as ORGAN, except that it gives zero weight to the discriminator. Appendix (A.3, A.5, A.7) provides further details.

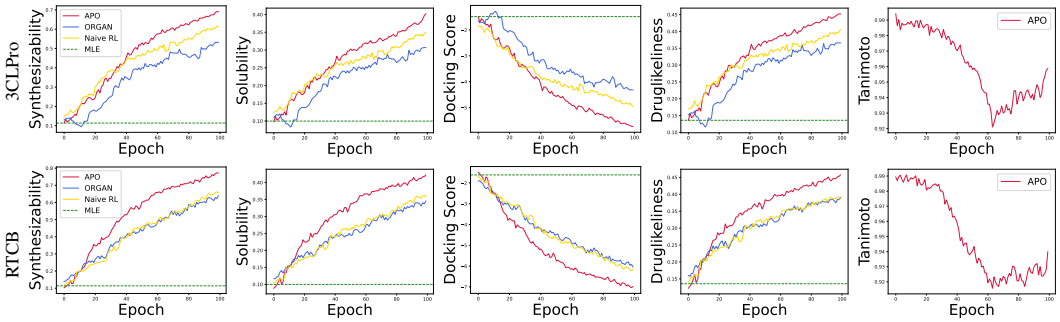
Molecules and vocabulary. Molecules can be depicted as textual sequences through the usage of SMILES notation, a method that captures the topological characteristics of a molecule based on well-defined chemical bonding principles. In the SMILES notation for small molecules, each character represents an atom or a bond in the molecule. The character set in SMILES sequence forms the vocabulary or action space in our setting. The SMILES representation adheres to predefined grammar rules. (See more details in Appendix A.1)

Datasets. The dataset used for training the surrogate models is built with a similar scheme as in an earlier virtual screening on SARS-CoV-2 targets [5, 15]. Each datapoint has an input SMILES string representing the molecule and an output docking score. The receptors used are prepared with the OEDOCK application and FPocket [33] is used if the protein active site is unknown. The score for each molecule is determined by inputting the molecular structure and receptor to OEDOCK and computing the minimum Chemgauss4 score over the ensemble of poses in the docking simulation. A set of 1 million orderable compounds within the ZINC15 dataset were docked to the 3CLPro (PDBID: 7BQY) SARS-CoV-2 protein and the RTCB (PDBID: 4DWQ) cancer protein. The resulting datasets are used for training two separate surrogate models for each protein.

Critics and evaluation metric. In this study, we evaluate the efficacy of APO in generating molecules with desirable attributes within the context of pharmaceutical drug discovery. We leverage the RDKit [32] chemoinformatics package and employ various performance metrics as follows: **Druglikeness:** The druglikeness measure the likelihood of a molecule being suitable candidate for a drug. **Solubility:** This metric assesses the likelihood of a molecule's ability to mix with water, commonly referred to as the water-octanol partition coefficient (LogP). Calculation is performed using RDKit's Crippen function. **Synthetizability:** This parameter quantifies the ease (score of 1) or difficulty (score of 0) associated with synthesizing a given molecule [19]. **Docking Score:** The docking score assesses the drug's potential to bind and inhibit the target site. To enable efficient computation, we employ a docking surrogate model (See Appendix A.4) to output this score.

5.2 Experimental results

Table 1 and Fig. 2 demonstrate that APO outperforms both MLE and the RL-based drug discovery baseline, ORGAN and Naive RL, across all the performance metrics for both viral and cancer-related proteins. Furthermore, APO not only surpasses all the baseline methods but also achieves a



(a) Synthesizability ↑ (b) Solubility ↑ (c) Docking score ↓ (d) Druglikeliness ↑ (e) Tanimoto ↑

Figure 2: Visualize the performance curve associated with Table 1, featuring APO (in red), and the baselines: MLE (in green), ORGAN (in blue), Naive RL (in yellow) .

high Tanimoto similarity compared to the original drug. This suggests that it retains the beneficial properties of the original drugs while enhancing others.

Two main factors contribute to APO outperforming ORGAN. The first factor is the APO algorithm employs advantage-alignment, which increases the probability of generating the sequence only when it exhibits a positive preference advantage and decreases it when the advantage is negative. In contrast, ORGAN consistently increases the probability of sampled actions for positive rewards, leading to faster convergence of the APO algorithm. Additionally, APO employs the Tanimoto similarity constraint, which enables the generator policy to explore a nearby molecular domain in relation to the original one. This increases the probability of preserving chemical scaffolds and functional groups that are beneficial for binding to target proteins and dissolving in solvents. Note that the performance curve of Tanimoto similarity in Fig. 2e initially decreases and then increases. This trend aligns ideally with the RL-based molecule generation improvement process. The initial decrease occurs because RL reduces the complexity of the original molecule to enhance the validity of the generator policy while improving the generated molecules' diversified properties. This causes the molecule to deviate from its original structure, leading to a decrease in Tanimoto similarity. Subsequently, there is a gradual increase in the trend as the generated molecules reach a decent level of diverse properties and begin optimizing their structure towards that of the original molecule, resulting in an increasing trend in Tanimoto similarity. Finally, the generated molecules not only improve the desired properties but also achieve a high Tanimoto similarity to the original drug. This reduces the likelihood of drastic structural changes that might result in unsynthesizable compounds. This process demonstrates that APO achieves a balance between optimizing desired properties and preserving the beneficial properties of the original drug.

6 Conclusion

We present DRUGIMPROVER, a practical and effective framework for drug optimization. Within the framework, we introduce APO, an advantage-alignment policy gradient algorithm with multi-critic guided exploration. This algorithm aims to align the generator policy with objectives from multiple critics and performs policy gradient updates based on the advantage preference. APO seeks to achieve maximal improvement based on the original drug while maintaining its necessary properties. Finally, we evaluate the docking score of our optimized compounds to two proteins, 3CLPro and RTCB, which are target proteins of SARS-CoV-2 and human cancer, respective. Our results reveal that our optimized compounds exhibit significantly stronger binding affinity to both proteins compared to compounds generated using baseline methods. Moreover, our compounds outperform those from the baseline method across all performance metrics, including solubility and synthesizability. Our research opens up new possibilities for enhancing drug optimization and inspires future investigations into addressing challenges within the realm of drug optimization. This includes exploring areas like the integration of graph information, a facet that our current work does not tackle.

Acknowledgements

This work is supported by the RadBio-AI project (DE-AC02-06CH11357), U.S. Department of Energy Office of Science, Office of Biological and Environment Research, the Improve project under contract (75N91019F00134, 75N91019D00024, 89233218CNA000001, DE-AC02-06-CH11357, DE-AC52-07NA27344, DE-AC05-00OR22725), the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Author contributions

X. Liu led the project, proposed the research ideas, scoped the technical content, designed the experiments and wrote the paper. S. Jiang conducted the experiments and joined the dicussion. A. Vasan provided the docking surrogate models, prepared the datasets and joined the discussion. A. Brace and O. Gokdemir joined the discussion. T. Brettin, F. Xia, and I. Foster supervised the project. R. Stevens supported and supervised the project.

References

[1] Sara Romeo Atance, Juan Viguera Diez, Ola Engkvist, Simon Olsson, and Rocío Mercado. De novo drug design using reinforcement learning with graph-based deep generative models. *Journal of Chemical Information and Modeling*, 62(20):4863–4872, 2022. 3, 5

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002. 3

[3] Sorin Avram, Thomas B Wilson, Ramona Curpan, Liliana Halip, Ana Borota, Alina Bora, Cristian G Bologa, Jayme Holmes, Jeffrey Knockel, Jeremy J Yang, et al. DrugCentral 2023 extends human clinical data and integrates veterinary drugs. *Nucleic Acids Research*, 51(D1): D1276–D1287, 2023. 1

[4] Yadu Babuji, Anna Woodard, Zhuozhao Li, Daniel S Katz, Ben Clifford, Rohan Kumar, Lukasz Lacinski, Ryan Chard, Justin M Wozniak, Ian Foster, et al. Parsl: Pervasive parallel programming in python. In *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, pages 25–36, 2019. 7

[5] Yadu Babuji, Ben Blaiszik, Tom Brettin, Kyle Chard, Ryan Chard, Austin Clyde, Ian Foster, Zhi Hong, Shantenu Jha, Zhuozhao Li, et al. Targeting sars-cov-2 with ai-and hpc-enabled lead generation: A first data release. *arXiv preprint arXiv:2006.02431*, 2020. 8

[6] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 3

[7] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022. 3

[8] Andreas Bender and Robert C Glen. Molecular similarity: a key technique in molecular informatics. *Organic & biomolecular chemistry*, 2(22):3204–3218, 2004. 2

[9] Florian Böhm, Yang Gao, Christian M Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. Better rewards yield better summaries: Learning to summarise without references. *arXiv preprint arXiv:1909.01214*, 2019. 3

[10] Jannis Born, Matteo Manica, Ali Oskooei, Joris Cadow, Greta Markert, and María Rodríguez Martínez. Paccmannrl: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. *Iscience*, 24(4), 2021. 2, 3

[11] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. *Advances in neural information processing systems*, 34:4933–4946, 2021. 6

[12] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022. 2

[13] Ching-An Cheng, Andrey Kolobov, and Alekh Agarwal. Policy improvement via imitation of multiple oracles. *Advances in Neural Information Processing Systems*, 33:5587–5598, 2020. 3, 4

[14] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 3

[15] Austin Clyde, Xuefeng Liu, Thomas Brettin, Hyunseung Yoo, Alexander Partin, Yadu Babuji, Ben Blaiszik, Jamaludin Mohd-Yusof, Andre Merzky, Matteo Turilli, et al. AI-accelerated protein-ligand docking for SARS-CoV-2 is 100-fold faster with no significant change in detection. *Scientific Reports*, 13(1):2105, 2023. 8

[16] Austin Clyde, Xuefeng Liu, Thomas Brettin, Hyunseung Yoo, Alexander Partin, Yadu Babuji, Ben Blaiszik, Jamaludin Mohd-Yusof, Andre Merzky, Matteo Turilli, et al. Ai-accelerated protein-ligand docking for sars-cov-2 is 100-fold faster with no significant change in detection. *Scientific Reports*, 13(1):2105, 2023. 2

[17] Michael Dickson and Jean Paul Gagnon. The cost of new drug discovery and development. *Discovery medicine*, 4(22):172–179, 2009. 1

[18] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023. 4

[19] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009. 8

[20] Argonne Leadership Computing Facility. `https://www.alcf.anl.gov/polaris`, last accessed on 10-2-2023. 7, 16

[21] Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International conference on machine learning*, pages 3668–3679. PMLR, 2020. 3

[22] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9), 2004. 6

[23] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017. 2, 3, 5, 8, 16

[24] Ikbel Hadj Hassine. Covid-19 vaccines and variants of concern: A review. *Reviews in medical virology*, 32(4):e2313, 2022. 1

[25] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*, 2019. 3

[26] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018. 3

[27] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912. 6

[28] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019. 3

[29] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pages 4849–4859. PMLR, 2020. 2, 3

[30] Brian P Kelley, Scott P Brown, Gregory L Warren, and Steven W Muchmore. Posit: flexible shape-guided docking for pose prediction. *Journal of Chemical Information and Modeling*, 55 (8):1771–1780, 2015. 2

[31] Matthew Lai. Giraffe: Using deep reinforcement learning to play chess. *arXiv preprint arXiv:1509.01549*, 2015. 2

[32] Greg Landrum et al. RDkit: Open-source cheminformatics software. https://www.rdkit.org. Accessed Oct 2023. 8

[33] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10(1):1–11, 2009. 8

[34] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: A research direction. *arXiv preprint arXiv:1811.07871*, 2018. 3

[35] Xuefeng Liu, Fangfang Xia, Rick Stevens, and Yuxin Chen. Contextual active online model selection with expert advice. In *ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2022. 3

[36] Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. Cost-effective online contextual model selection. *arXiv preprint arXiv:2207.06030*, 2022. 3

[37] Xuefeng Liu, Takuma Yoneda, Rick L Stevens, Matthew R Walter, and Yuxin Chen. Blending imitation and reinforcement learning for robust policy improvement. *arXiv preprint arXiv:2310.01737*, 2023. 3

[38] Xuefeng Liu, Takuma Yoneda, Chaoqi Wang, Matthew R Walter, and Yuxin Chen. Active policy improvement from multiple black-box oracles. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 22320–22337, 2023. 3, 4

[39] Behzad Mansoori, Ali Mohammadi, Sadaf Davudian, Solmaz Shirjang, and Behzad Baradaran. The different mechanisms of cancer drug resistance: a brief review. *Advanced pharmaceutical bulletin*, 7(3):339, 2017. 1

[40] Mark McGann. Fred pose prediction and virtual screening accuracy. *Journal of chemical information and modeling*, 51(3):578–596, 2011. 17

[41] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[42] Daniel Neil, Marwin Segler, Laura Guasch, Mohamed Ahmed, Dean Plumbley, Matthew Sellwood, and Nathan Brown. Exploring deep recurrent models with reinforcement learning for molecule design. In *ICLR*, 2018. 2, 3

[43] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):1–14, 2017. 3

[44] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. 6

[45] Georg Polya and Ronald C Read. *Combinatorial enumeration of groups, graphs, and chemical compounds*. Springer Science & Business Media, 2012. 2

[46] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3

[47] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018. 3, 5

[48] Sudeep Pushpakom, Francesco Iorio, Patrick A Eyers, K Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, Alan Norris, Philippe Sanseau, David Cavalla, and Munir Pirmohamed. Drug repurposing: Progress, challenges and recommendations. *Nature Reviews Drug Discovery*, 18(1):41–58, 2019. 1

[49] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. 4

[50] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010. 6

[51] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. 3, 4

[52] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 3

[53] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 5

[54] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 7

[55] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*, 2023. 4

[56] Niclas Ståhl, Goran Falkman, Alexander Karlsson, Gunnar Mathiason, and Jonas Bostrom. Deep reinforcement learning for multiparameter optimization in de novo drug design. *Journal of chemical information and modeling*, 59(7):3166–3176, 2019. 3

[57] Teague Sterling and John J Irwin. ZINC15–ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. 7

[58] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. 3

[59] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999. 5, 6

[60] Ryan K Tan, Yang Liu, and Lei Xie. Reinforcement learning for systems pharmacology-oriented and personalized drug design. *Expert Opinion on Drug Discovery*, 17(8):849–863, 2022. 3

[61] Youhai Tan, Lingxue Dai, Weifeng Huang, Yinfeng Guo, Shuangjia Zheng, Jinping Lei, Hongming Chen, and Yuedong Yang. Drlinker: Deep reinforcement learning for optimization in fragment linking design. *Journal of Chemical Information and Modeling*, 62(23):5907–5917, 2022. 2, 3

[62] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2

[63] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3

[64] Archit Vasan, Thomas Brettin, Rick Stevens, Arvind Ramanathan, and Venkatram Vishwanath. Scalable lead prediction with transformers using hpc resources. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 123–123, 2023. 7

[65] Archit Vasan, Rick Stevens, Arvind Ramanathan, and Vishwanath Venkatram. Benchmarking language-based docking models. 2023. 16

[66] Logan Ward, Ganesh Sivaraman, J Gregory Pauloski, Yadu Babuji, Ryan Chard, Naveen Dandu, Paul C Redfern, Rajeev S Assary, Kyle Chard, Larry A Curtiss, et al. Colmena: Scalable machine-learning-based steering of ensemble simulations for high performance computing. In *2021 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC)*, pages 9–20. IEEE, 2021. 7

[67] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988. 2, 4

[68] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. 4

[69] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021. 3

[70] Sanghyun Yi, Rahul Goel, Chandra Khatri, Alessandra Cervone, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015*, 2019. 3

[71] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018. 3

[72] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI Conference on Artificial Intelligence*, 2017. 5

[73] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 2

[74] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023. 4

[75] Koichi Yuki, Miho Fujiogi, and Sophia Koutsogiannaki. Covid-19 pathophysiology: A review. *Clinical immunology*, 215:108427, 2020. 1

[76] Yunjiang Zhang, Shuyuan Li, Miaojuan Xing, Qing Yuan, Hong He, and Shaorui Sun. Universal approach to de novo drug design for target proteins using deep reinforcement learning. *ACS omega*, 8(6):5464–5474, 2023. 2, 3, 5

[77] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):10752, 2019. 3

[78] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. 3

A Appendix

A.1 Molecules and vocabulary

Here's a breakdown of what each character means in the SMILES string:

Atoms: *Capital letters*: Represent the element symbols for atoms. For example, "C" stands for carbon, "H" for hydrogen, "O" for oxygen, "N" for nitrogen, and so on. *Lowercase letters:* Used to specify the configuration of certain atoms, such as "c" indicating a carbon atom in an aromatic ring.

Bonds: *Single Bond(-)*: Represented by a hyphen (-), signifying a single covalent bond between two adjacent atoms. *Double Bond(=)*: Represented by an equal sign (=), indicating a double covalent bond between two adjacent atoms. *Triple Bond(#)*: Represented by a pound sign (#), denoting a triple covalent bond between two adjacent atoms. *Aromatic Bond (":")*: Represented by two consecutive colons (":"), signifying an aromatic bond in an aromatic ring structure.

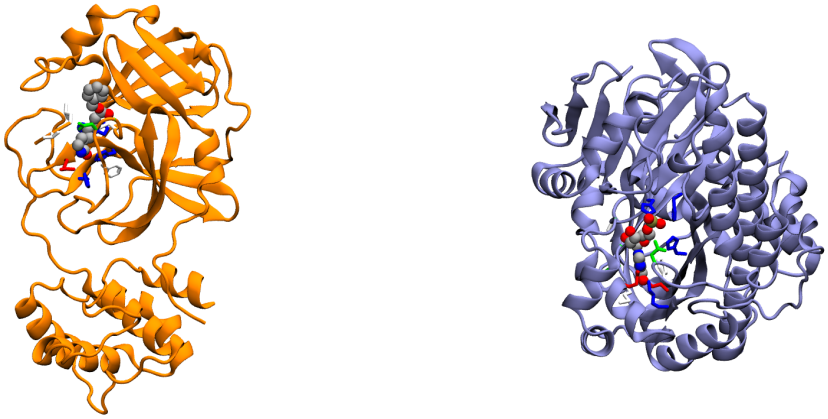
Numbers: *Subscript Numbers:* Positioned after an atom symbol to specify the number of that particular atom in the molecule.

Parentheses (and): *Parentheses:* Employed to group atoms or substructures together.

Dot (.) and Plus (+): *Dot (.)* may be used to separate distinct fragments or components of a molecule. *Plus (+)*is used to indicate the presence of charged ions, such as "[Na+]" representing a sodium ion.

Other Characters *Brackets ([and]):* May be used to enclose isotopic information or intricate substructures. *Slash (/) and Backslash ():* Sometimes used to denote stereochemistry. *Ampersand (&):* Used to represent a bridge bond in complex molecular structures.

A.2 Binding sites of 3clpro and RTCB



(a) 3CLPro. (b) RTCB.

Figure 3: The binding sites of proteins 3CLPro (PDB ID: 7BQY) (**Left**) and RTCB (PDB ID: 4DWQ) (**Right**). Binding sites are defined around the crystallized compound using Open Eye software.

A.3 The sequence generative model

The sequence generative model. To simulate the real-world structured sequences, we consider a language model to capture the dependency of the tokens. In this work, we use a RNN with LSTM cells as π_θ^G to generate the real data distribution $p\left(x_t|x_1,\ldots,x_{t-1}\right)$. Maximum Likelihood Estimation (MLE) aims to minimize the cross-entropy between the true data distribution p and our approximation q, which is expressed as $\mathbb{E}_{x\sim p}[\log q(x)]$.

ORGAN. ORGAN [23] is a generative model designed to optimize sequence distributions. It achieves this by leveraging a combination of domain-specific metrics (objective) and adversarial feedback obtained from a discriminator. The balance between these two components is maintained through a tunable parameter. Within the ORGAN architecture, the generator is constructed as an RNN equipped with LSTM cells. In contrast, the discriminator employs a Convolutional Neural Network (CNN) specifically tailored for text classification tasks. Notably, the Wasserstein distance is chosen as the loss function for the discriminator, ensuring enhanced stability during training.

Naive RL. ORGAN employs a combination of a discriminator and domain objectives as rewards. And by setting the weight of discriminator to be zero, the model ignores the discriminator and becomes a "Naive" RL algorithm [23].

A.4 Surrogate model

The surrogate model [65] is a simplified version of a BERT-like transformer, which is widely used in natural language processing. In the model, tokenized SMILES strings are inputted and then positionally embedded. Outputs are then passed to a stack of five transformer blocks, each containing a multi-head attention layer (21 heads), dropout layer, layer normalization with residual connection, and feed forward network. The feed forward network consists of two dense layers followed by dropout and layer normalization with residual connection. After the transformer block stack, a final feed forward network is used to output the predicted docking score.

A.5 Setup

Setup. To guarantee an equitable assessment, every algorithm (ORGAN, Naive RL, and APO), is trained using an identical pretrained LSTM-based generator π_θ^G. During the training of ORGAN and Naive RL, we adhere to the multi-objective training approach described in [23], which involves alternating between objectives (synthesizability, solubility, docking score and druglikeliness). Specifically, each epoch of ORGAN is dedicated to a different objective, cycling through them for a total of 25 epochs per objective. APO enhances all objectives simultaneously in each epoch.

A.6 Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of the Polaris supercomputer at the Argonne Leadership Computing Facility where each node contains CPUs (64 cores) and 4 A100 GPU nodes [20]. The training time for each model was approximately 3 hours. We conducted other RL experiments on a cluster that includes CPU nodes (approximately 280 cores) and GPU nodes (approximately 110 Nvidia GPUs, ranging from Titan X to A6000, set up mostly in 4- and 8-GPU configurations). Based on the computing infrastructure, we obtained the wall-time comparison in Table 2 as follows.

Methods	Total Run Time
ORGAN	13h
Naive RL	12h
APO	21h

Table 2: Wall-time comparison between different methods.

A.7 Hyperparameters and architectures

Table 3 provides a list of hyperparameter settings we used for our experiments.

Parameter	Value
Shared	
Learning rate	1×10^{-4}
Optimizer	Adam
Nonlinearity	ReLU
# of Epochs for Training	100
APO Objective Weight	
Docking Score	0.15
Druglikeliness	0.15
Synthesizability	0.15
Solubility	0.15
Tamimoto Similarity	0.4
APO Other	
Amplifier	100 (3CLPro), 10 (RTCB)
Fingerprint Size	16
Normalize Min/Max	$[-10, 10]$

Table 3: Hyperparameters.

A.8 Code and data availability

For all code and data used in experiments, please refer to https://github.com/xuefeng-cs/DrugImprover. We release a drug optimization dataset comprising 1 million ligands along with their OEDOCK scores to five proteins associated with cancer: colony stimulating factor 1 receptor (CSF1R) kinase domain (PDB ID: 6T2W), NOP2/Sun RNA methyltransferase 2 (NSUN2) (AlphaFold derived), RNA terminal phosphate cyclase B (RTCB) ligase (PDB ID: 7P3B), and Tet methylcytosine dioxygenase 1 (TET1) (AlphaFold derived), and Wolf-Hirschhorn syndrome candidate 1 (WHSC1) (PDB ID: 7MDN) as well as one protein from SARS-COV2: 3CLPro (PDBID: 7BQY). The receptor file generated from OpenEye is also released here. All docking was generated via OpenEye FRED docking. Additionally, we release the pretrained model for each protein [40].