GPT-Driven Drug Optimization with Structured Policy Optimization Post-training

Xuefeng Liu* XUEFENG.LIU@UFL.EDU
College of Medicine
University of Florida
Gainesville, FL, USA

Songhao Jiang SHJIANG@UCHICAGO.EDU
Department of Computer Science
University of Chicago
Chicago, IL, USA

Siyu Chen SIYU.CHEN.SC3226@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, Connecticut, USA

Zhuoran Yang ZHUORAN.YANG@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, Connecticut, USA

Yuxin Chen CHENYUXIN@UCHICAGO.EDU
Department of Computer Science
University of Chicago
Chicago, IL, USA

Ian Foster FOSTER@UCHICAGO.EDU
Department of Computer Science
University of Chicago
Chicago, IL, USA

Rick Stevens STEVENS@CS.UCHICAGO.EDU
Department of Computer Science
University of Chicago
Chicago, IL, USA

Abstract

Post-training is essential for steering generative models toward specific objectives. However, despite the growing importance of drug optimization, reinforcement learning algorithms tailored to this setting remain underexplored. In this work, we study the problem of drug optimization and propose a novel reinforcement learning framework for post-training a drug-optimization Generative Pre-trained Transformer (GPT). Our approach improves candidate molecules with respect to target objectives while preserving the desirable chemical properties of the original compounds. This work consists of two main components. (1) DrugImproverGPT, a framework designed to enhance the robustness and efficiency of drug

* Corresponding author. Part of this work was conducted at the University of Chicago.

optimization. It couples a GPT-based generative model with a theoretically grounded Structured Policy Optimization (SPO) algorithm. SPO provides a principled perspective on post-training generative models by explicitly aligning improvements in generated molecules with their corresponding input molecules under specified objectives. (2) A large-scale dataset comprising one million compounds, each annotated with OEDOCK docking scores across five human cancer-related proteins and 24 binding sites from the SARS-CoV-2 virus. Extensive in-silico experiments demonstrate that SPO generates candidates with improved values under the specified computational objectives. DrugImproverGPT is intended as a computational hit-to-lead prioritization framework whose generated candidates require subsequent experimental validation.

1. Introduction

The cost of discovering a new drug through conventional approaches is estimated to range from hundreds of millions to billions of dollars (Dickson and Gagnon, 2009). 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 (Avram et al., 2023), which involves using FDA-approved drugs for different diseases instead of developing new drugs from the ground up. Yet despite some successes (Pushpakom et al., 2019), the effectiveness of drug repurposing has been limited since the drug is usually designed specifically for treating a particular disease. However, the emergence of rapidly evolving virus variants (Hadj Hassine, 2022), such as those associated with SARS-CoV-2 (Yuki et al., 2020), as well as drug resistant cancer cells (Mansoori et al., 2017), has sparked increased interest and an urgent need to expedite the discovery of effective drugs.

In this work, we propose a reinforcement learning-based computational molecular optimization algorithm for prioritizing structural modifications of existing compounds against targets associated with rapidly evolving viruses and cancer, helping to address the aforementioned limitations of drug discovery and drug repurposing. The resulting molecules are computationally optimized under the specified reward functions and should be regarded as candidates for subsequent physics-based, retrospective, and experimental evaluation. RL has achieved superhuman performance in domains such as chess (Lai, 2015), video games (Mnih et al., 2013), and robotics (Brunke et al., 2022). However, despite promising early results (Born et al., 2021; Guimaraes et al., 2017; Jin et al., 2020; Neil et al., 2018; Tan et al., 2022b; Zhang et al., 2023), 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: 1) *Search space complexity*: An RL algorithm for drug discovery needs to demonstrate both sample and computational efficiency, but the overwhelming complexity of the search space (Polya and Read, 2012) renders RL incapable of adequately exploring potential effective actions and states required for policy learning. 2) *Sparse rewards*: In contrast to the

continuous reward environment found in popular environments like DeepMind Control Suite (Tassa et al., 2018) or Meta-World (Yu et al., 2020), drug generation operates within a sparse reward environment where rewards are only obtainable upon a complete molecule. 3) *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. 4) *Preservation of original beneficial properties*: Lastly, as drugs with similar chemical structures should exhibit similar biological/chemical effects (Bender and Glen, 2004), it is crucial to strike a balance between optimizing the drug and preserving the original drug's beneficial properties.

Generalizable Insights about Machine Learning in the Context of Healthcare

We present DrugImproverGPT, a GPT-based drug optimization framework designed to computationally optimize multiple measurable properties of an original molecule in a robust and efficient manner. Within this workflow, we introduce the **S**tructured **P**olicy **O**ptimization (**SPO**) algorithm to utilize the advantage preference of properties improvement to perform direct policy optimization. DrugImproverGPT and SPO effectively tackle the challenges outlined above in the following manner: (1.) *Designing a GPT model for drug optimization.* In this study, we develop a GPT model tailored specifically for drug optimization, incorporating a specialized corpus and custom loss function, among other features. (2.) *Sample complexity, sparsity, and efficiency.* Drug design's sparse rewards challenge pure RL due to complex search spaces. SPO tackles this by pre-training a GPT model with imitation learning based on prior drug SMILES (Weininger, 1988) design experience. It mitigates reward sparsity using Top-K and Top-P Beam Search from GPT for intermediate reward estimates. To reduce the computational cost of docking scores (e.g., via OEDOCK (Kelley et al., 2015)), DrugImproverGPT employs a transformer-based surrogate model for efficient score prediction. (3.) *Property preserving.* To balance encouraging structural preservation through fingerprint-based similarity with optimizing chemical attributes, we use Tanimoto similarity as a critic to maximize similarity between original and generated drugs. Similarity does not guarantee preservation of biological efficacy or broader ADMET behavior; it serves as a structural regularizer within the computational objective. (4.) *RL Post-training.* Our proposed SPO algorithm leverages the advantageous preference of a generated drug over the original drug based on multiple objectives as the policy gradient signal. It performs direct policy improvement on an GPT-based generative model and addresses the sparse reward problem through scaffold and partial molecule improvement. The contributions of DrugImproverGPT are summarized as follows:

- We introduce the DrugImproverGPT framework, which includes a ground-up designed GPT model tailored for drug optimization, along with a novel RL post-training algorithm, SPO, designed for drug optimization with theoretical analysis.
- Through comprehensive in-silico experiments on viral- and cancer-related protein targets, we show that SPO achieves higher computational objective values than the evaluated baselines across most metrics.
- 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 virus. See more details in Appendix E.

2. Related Work

RL Post-training. Post-training, also known as fine-tuning, the generator model is critical for optimizing drugs toward desired properties. Prior RL post-training methodologies aimed at aligning models with feedback from both humans (RLHF (Bai et al., 2022a; Christiano et al., 2017; Ibarz et al., 2018; Touvron et al., 2023)) and AI (RLAIF (Bai et al., 2022b; Leike et al., 2018)), which have recently found applications in the fine-tuning of language models for tasks like text summarization (Böhm et al., 2019; Stiennon et al., 2020; Wu et al., 2021; Ziegler et al., 2019), dialogue generation (Hancock et al., 2019; Jaques et al., 2019; Yi et al., 2019), and language assistance (Bai et al., 2022a). A core feature of RLHF and RLAIF lies in training a reward model or make direct policy improvement from the comparison feedback, such as Rank Responses to align Human Feedback (RRHF) (Yuan et al., 2023), Reward Ranked Finetuning (RAFT) (Dong et al., 2023), Preference Ranking Optimization (PRO) (Song et al., 2023), and Direct Preference Optimization (DPO) (Rafailov et al., 2023). Differing from previous works, our approach does not rely solely on feedback from a single human or AI model; instead, we engage multiple critics to evaluate the advantage preference of the generated vs. original drug based on comprehensive assessments, including factors like solubility. Moreover, we make direct policy improvement by using the advantage preference in standard RL instead of binary feedback.

Grouped and KL-regularized policy optimization. Recent post-training methods also estimate relative advantages among multiple responses sampled from the same prompt and regularize the updated policy toward a reference model. These grouped objectives are conceptually separable from the design of the reward function. In molecular optimization, an improvement-aware grouped baseline can use the same input-anchored reward, partial-molecule reward, and scaffold reward as SPO while replacing the outer policy update with group-relative normalization and explicit KL regularization. We therefore include a matched-budget improvement-aware GRPO-KL baseline in Section 6

GPT model for drug optimization. GPT has been employed in molecule generation (Bagal et al., 2021; Rothchild et al., 2021; Frey et al., 2023) and drug discovery (Bran and Schwaller, 2023; Liu et al., 2024). In contrast, our work focuses on drug optimization, which requires maintaining the original drug's beneficial structure and properties rather than designing from scratch. A notable work in the drug optimization domain is REINVENT 4 (He et al., 2021, 2022; Loeffler et al., 2024), which has developed transformer-based generative models with a strong focus on pretraining. However pretraining facilitates the generation of molecules similar to those in the training dataset, it also inherently limits the scope of exploration due to biases present in the training data. Furthermore, REINVENT 4 categorizes the features, resulting in insensitivity to numerical changes, and the molecules generated lack optimization for specific desired objectives, such as drug-likeness, among others. In contrast, DrugImproverGPT further post-trains the generator with SPO to increase the expected

computational reward of generated molecules relative to their corresponding input molecules. We defer more details into Appendix B.

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: $Q^\pi(s, a) := \mathbb{E}^\pi \left[\sum_{t=0}^{T} R(s_t, a_t) \,|s_0 = s, a_0 = a\right]$, where the expectation is taken over the trajectory following π, and the value function is noted as: $V^\pi(s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s, a)]$.

Language model. Each training corpus includes a start token [BOS], a sequence of tokens $\mathbf{y}$ where each $y_i \in \mathcal{V}$, and a termination action [EOS]. Here, each action $a \in \mathcal{A}$ is represented as a token y in the Transformer's vocabulary $\mathcal{V}$, with $\mathcal{V} := \mathcal{A}$. Each molecule is represented by a sequence of tokens $\mathbf{y}$ to construct a SMILES (Weininger, 1988) string, and this applies to both partial and complete molecules. Let $\circ$ represents string concatenation, and let $\mathcal{V}^*$ denote the Kleene closure of $\mathcal{V}$. We define the set of complete training corpus as:

$$\mathcal{C} := \{[\text{BOS}] \circ \mathbf{v} \circ [\text{EOS}] \mid \mathbf{v} \in \mathcal{V}^*\}. \tag{1}$$

The GPT generator policy π_θ, which is parameterized by a deep neural network (DNN) with learned weights θ, is defined as a product of probability distributions: $\pi_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} \pi_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t})$, where $\pi_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t}) = P(y_t|\mathbf{y}_{<t}, X)$ is a distribution of next token y_t, $\mathbf{y}_{<t} = [y_1, \cdots, y_{t-1}]$, and $\mathbf{x}$ represents an input sequence (prompt). The decoding process in text generation is designed to identify the most probable hypothesis from all potential candidates by resolving the following optimization problem:

$$\mathbf{y}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}_T} \log \pi_\theta(\mathbf{y}|\mathbf{x}). \tag{2}$$

To estimate the expected reward for a partial molecule, we employ Top-PK (Liu et al., 2024) to navigate the exponentially vast search space to form a complete valid molecule. For sampling the token $y_i \sim \text{Top-PK}(\mathbf{y}_{<i}, p, k)|_{\mathbf{x}}$, where Top-PK generates the sequence by recursively picking the top candidates at each step i according to

$$\text{Top-PK}(\mathbf{y}_{<i}, p, k)|_{\mathbf{x}} = \mathcal{A}_{\mathbf{y}_{<i}}, \tag{3}$$

$$\text{where } \mathcal{A}_{\mathbf{y}_{<i}} = \left\{y^1, \ldots, y^j\right\} \in \mathcal{V}^j, \text{ and } j = \min\left\{\arg\min_{j'} \sum_{l=1}^{j'} \pi_\theta\left(y^l|\mathbf{x}, \mathbf{y}_{<i}\right) \geq p,\ k\right\},$$

where the candidates $y^1, \ldots, y^{j'}, \ldots, y^{|\mathcal{V}|} \in \mathcal{V}$ are indexed by descending order of $\pi_\theta(\cdot \,|\, \mathbf{x}, \mathbf{y}_{<i})$, $p \in (0, 1]$ denotes the cumulative probability threshold and k represents the maximum number

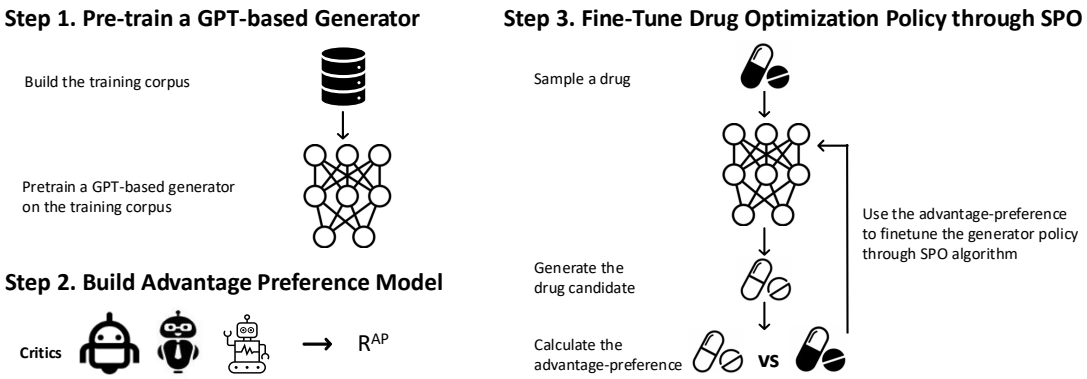

Figure 1: DrugImproverGPT framework. It comprises two major components: (1) A GPT model designed for drug optimization. (2) A Structured Policy Optimization (SPO) algorithm aims to fine-tune the GPT model for drug improvement across desired properties.

of candidates for the next tokens. BON (Gao et al., 2023) sampling technique at inference time generates N samples which are then ranked by reward model. Then top ranked candidate is selected, which can expressed as

$$\mathrm{BON}\left(\mathbf{y}_{<i}, N, R\right)|_{\mathbf{x},p,k} = \max_{\mathbf{Y}_j \in \{\mathbf{Y}_1,\cdots,\mathbf{Y}_N\}} R\left(\mathbf{Y}_j\right), \tag{4}$$

$$\text{where } Y_j = \left[\mathbf{y}_{<i}, y_i, \cdots, y_T\right]_j, \text{ and } y_i \sim \textsc{Top-PK}\left(\mathbf{y}_{<i}, p, k\right)|_{\mathbf{x}}. \tag{5}$$

Drug optimization. We formalize the drug optimization problem within the framework of MDP. Given a dataset consisting of real-world structured sequences represented as SMILES (Weininger, 1988) strings, our objective is to train a GPT model π_θ to generate a high-quality sequence denoted as $\mathbf{y}_T = (y_1, \ldots, y_t, \ldots, y_T), y_t \in \mathcal{V}$, and aim to outperforming an input sequence X in desired properties. The length of the output 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 $\pi_\theta\left(y_t|\mathbf{y}_{<t}, X\right)$ operates in a stochastic manner, the state transition function $\mathcal{P}$ becomes deterministic once an action has been chosen. To estimate the Q value, we reference the REINFORCE algorithm (Williams, 1992), which we define as $Q\left(s = \mathbf{y}_{<T}, a = y_T\right) = R\left(\mathbf{y}_T\right)$.

4. The DrugImproverGPT Framework

In this work, we propose DrugImproverGPT as in Fig. 1, which comprises two major components: (1) A GPT model designed from the ground up for drug optimization. (2) A Structured Policy Optimization (SPO) algorithm with theoretical support. We introduce each part in details as follows.

4.1. Designing & pretraining a GPT generator

In drug optimization, firstly, we construct a molecule pair (X, Y), where X represents for original molecule, and Y represents for the target optimized molecule. We randomly select non-duplicated pair (X, Y) from ZINC15 (Sterling and Irwin, 2015) dataset (See Appendix D.6), and added the pair to the training set by meeting the following criteria of Tanimoto Similarity (Bajusz et al., 2015) and molecule scaffold (Landrum et al., 2013):

$$\text{Tanimoto}\,(X, Y) > 0.5 \quad \text{or} \quad \text{Scaffold}(X) = \text{Scaffold}(Y),$$

The motivation for such a form is to provide an initialization for a diversified molecule pair while constraining the similarity between the pair. After obtaining the training set of molecule pairs, we formed the training corpus as follows:

$$\mathcal{C} = \left\{ \langle S \rangle, \underbrace{x_1, \cdots, x_T}_{\text{source molecule X}}, \langle L \rangle, \underbrace{y_1, \cdots, y_T}_{\text{target molecule Y}} \right\}, \tag{6}$$

where $\langle S \rangle$ stands for the source ligand and $\langle L \rangle$ stands for the target ligand. We include visualization towards the corpus in Appendix. We enhance the training by concentrating on pairs of molecules through Causal Language Modeling (CLM) (Vaswani et al., 2017). The parameters θ of the GPT generator π_θ are trained through the minimization of the negative log-likelihood (NLL) for the complete molecular pair across the entire training corpus. This process is described as follows:

$$\text{NLL}\,(X, Y) = -\log P\,(Y|X) = -\log \prod_{l=1}^{T} P\,(y_l|\mathbf{y}_{<l}, X) = -\sum_{l=1}^{T} \log P\,(y_l|\mathbf{y}_{<l}, X)\,, \tag{7}$$

where T signifies the total number of tokens related to Y. The NLL measures the likelihood of transforming a specific original molecule into a designated target molecule. Given the goal of drug optimization, we aim for the generated drugs to resemble the originals. Therefore, we have incorporated a regularization term into the loss in Equation (7), which penalizes the NLL if the sequence does not adhere to a specified similarity metric. Finally, we propose the following loss function:

$$\mathcal{L} = \frac{1}{|\mathcal{C}|} \sum_{(X,Y)\in\mathcal{C}} \frac{\lambda \cdot \text{NLL}\,(X, Y)}{(1-\lambda) \cdot \text{Similarity}\,(X, Y)}, \tag{8}$$

where $\lambda \in (0, 1)$. Consequently, training the model with the loss function described in Eqn. (8) can generate the corresponding target molecule when provided with a source molecule. However, this approach primarily focuses on maximizing likelihood without considering specific metrics of interest, making it unsuitable for optimizing objectives that differ from those in its training set, as encoded in π_θ. Therefore, these generation algorithms cannot be directly applied to design molecules that fulfill various objectives, such as attaining a high docking score at a specific target site or improving upon specific desired metrics. We aim to further refine the GPT model to generate specific improved outcomes using reinforcement learning techniques in the next phase.

Algorithm 1 Structured Policy Optimization (SPO)

Require: GPT model π_θ; roll-out policy π_β; a pre-train dataset $\mathcal{B}$, critics $\mathbf{C}$.

1: Pre-train π_θ using loss function (8) and training corpus (6) through CLM objective.
2: $\beta \leftarrow \theta$.
3: **for** $n = 1, \ldots, N$ **do**
4: $X \sim \rho_0$, where $\rho_0 \in \Delta(\mathcal{B})$.
5: Generate $Y_{1:T} = (y_t, \ldots, y_T) \sim \pi_\theta(\cdot|X)$.
▷ /* incorporating scaffold and partial reward */
6: Compute advantage preference R^{AP} (17) by incorporating scaffold and partial molecule component.
7: Update generator θ via policy gradient by (13)(14).
8: $\beta \leftarrow \theta$.
9: **end for**

4.2. Structured policy optimization

Normalized reward. In this work, we adopt the approach of Liu et al. (2024) to construct the reward for multiple critics. Given an ensemble of critics

$$\mathbf{C}(\mathbf{y}_T) = [C^{\text{Druglikeness}}(\mathbf{y}_T), C^{\text{Solubility}}(\mathbf{y}_T), C^{\text{Synthesizability}}(\mathbf{y}_T), C^{\text{Docking}}(\mathbf{y}_T)],$$

where $\mathbf{y}_T := s_T, C : \mathbf{y}_T \to \mathbb{R}$. We leverage the RDKit (RDkit) chemoinformatics package to calculate the listed critics: **Druglikeness:** The druglikeness measure the likelihood of a molecule being suitable candidate for a drug. **Solubility:** We use the water–octanol partition coefficient, LogP, as a physicochemical proxy related to hydrophobicity and aqueous solubility. **Synthetizability:** This parameter quantifies the ease (score of 1) or difficulty (score of 10) associated with synthesizing a given molecule (Ertl and Schuffenhauer, 2009). **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 D.9) to output this score. Here we design the reward function to align the drug optimization with multiple objectives. For a fully generated SMILES sequence, we derive the following normalized reward function based on assessments from multiple critics with equal weight (Liu et al., 2024) as follows:

$$R_c(\mathbf{y}_T) := R_c(\mathbf{y}_T|X) = \beta \cdot \text{Norm}\left(C^{\text{Tanimoto}}(X, \mathbf{y}_T)\right) + \sum_{i=0}^{|\mathbf{C}|-1} \lambda \cdot \text{Norm}\left(C_i(\mathbf{y}_T)\right), \quad (9)$$

where $\lambda = \frac{1-\beta}{|\mathbf{C}|+1}$. We use Norm[1] to normalize different attributes onto the same scale. In this study, we employ the Tanimoto similarity[2] calculation C^{Tanimoto} to quantify the chemical similarity between the generated compound and the original drug. Essentially, this calculation involves first computing Morgan Fingerprints (Rogers and Hahn, 2010) for each molecule and then measuring the Jaccard distance (Jaccard, 1912) (i.e., intersection over union) between the two fingerprints. In addition, we follow the equal-weight normalization

1. Here, we define Norm as min-max normalization to scale the attributes onto the range [-10, 10].
2. High Similarity limits structural diversity. We aim to reach a reasonable level of Tanimoto Similarity.

used in prior work to preserve comparability across methods. This composite metric should not be interpreted as implying that all properties are equally important or equally difficult to optimize. In particular, improvements in readily optimized physicochemical metrics may contribute to the aggregate score in the same way as improvements in docking. We therefore report every component separately and treat the average normalized reward as a benchmark summary rather than a definitive medicinal-chemistry utility function.

Structured policy gradient with *Complete* molecular optimization. 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, the advantage function (Sutton et al., 1999) is defined as follows: $\boldsymbol{A}^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - b(s)$. 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:

$$V(s) = \mathbb{E}_{a\sim\pi_\theta(s)}[Q(s,a)] = \mathbb{E}_{y_t\sim\pi_\theta(\mathbf{y}_{<t})}[Q(\mathbf{y}_{<t}, y_t)].$$

Thus, we have advantage function as

$$\boldsymbol{A}^{\pi_\theta}(s,a) = \boldsymbol{A}^{\pi_\theta}(\mathbf{y}_{<t}, y_t) = Q^{\pi_\theta}(\mathbf{y}_{<t}, y_t) - V^{\pi_\theta}(\mathbf{y}_{<t}).$$

Here we employ the one-step RL (Brandfonbrener et al., 2021; Peng et al., 2019) 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 $Y_{1:T} = \mathbf{y}_T$ as a single action generated by the policy π_θ. Subsequently, we receive rewards from critics, and the episode concludes. This leads to the formulation of our advantage function as follows:

$$\boldsymbol{A}^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s_0, \mathbf{y}_T) - V^{\pi_\theta}(s_0) = R_c(Y_{1:T}) - R_c(X),$$

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

$$r^{\texttt{AP}}(Y_{1:T}, X) = R_c(Y_{1:T}) - R_c(X), \tag{10}$$

***Partial* molecular optimization.** Nonetheless, the reward function only supports a reward value for a completed sequence for global optimization. In contrast with previous approaches, we also aim to make improvement on partial molecule for local optimization by uniformly sampling a subsequence $Y_{1:j}, j \sim \mathcal{U}([T])$. To achieve this, we employ a Roll-in-Roll-out (RIRO) (Cheng et al., 2020; Liu et al., 2023b; Ross and Bagnell, 2014) scheduling, utilizing a roll-out policy denoted as π_β (same as learner policy in our experiment) to sample the unknown last $T-j$ tokens through BON approach (3). We propose a novel notion of advantage function, termed Advantage Preference (AP), by incorporating partial molecule improvement into (10):

Definition 1 (Advantage preference) *We define advantage preference as a combination of rewards from both complete and partial molecules:*

$$R^{\texttt{AP}}(Y_{1:T}, X) = \lambda \cdot \mathbb{E}_{j\in\mathcal{U}([T])} r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}, X) + (1-\lambda)\cdot r^{\texttt{AP}}(Y_{1:T}, X), \tag{11}$$

where

$$r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}; X_{1:T}) = \mathbb{E}_{\rho}[R_c(Y_{1:T}) - R_c(X_{1:T}) \mid X_{1:j}, Y_{1:j}],$$
$$\rho = Y_{j+1:T} \sim \texttt{BON}(Y_{1:j}), X_{j+1:T} \sim \texttt{BON}(X_{1:j}),$$

$\lambda \in [0, 1]$ *is set to* $\frac{1}{2}$ *in the experiments, and* $r^{\texttt{AP}}(Y_{1:T}; X_{1:T})$ *is defined in Eq.* (10).

The advantage preference of (11) will be employed directly in the policy gradient (13) to finetune the generator policy π_θ. 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 X at the end of the sequence as follows:

$$J(\theta) = \mathbb{E}_{X \sim \rho_0, Y_{1:T} \sim \pi_\theta(\cdot \mid X)} \left[R^{\texttt{AP}}(Y_{1:T}, X) \right], \tag{12}$$

Thus, we have gradient g as follows:

$$\mathbb{E}_{X \sim \rho_0, Y_{1:T} \sim \pi_\theta(\cdot|X)} \left[\nabla_\theta \log \pi_\theta (Y_{1:T}|X) \cdot R^{\texttt{AP}}(X, Y_{1:T}) \right], \tag{13}$$

where $Y_{1:T}$ is the generated sequence from π_θ and X 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:

$$\theta \leftarrow \theta + \alpha_n g, \tag{14}$$

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

5. Theoretical Analysis

We analyze the optimization properties of the SPO objective. Under the assumptions stated below, we show that its reward-densified objective shares an optimizer with the corresponding full-sequence objective and that its gradient can be decomposed into partial and terminal reward signals. Recall that we have optimization target $J(\theta)$ defined in (12) with advantage function $R^{\texttt{AP}}$ defined in (11). Define $J_0(\pi) = \mathbb{E}_{X \sim \rho_0}[r^{\pi}_{\texttt{AP}}(X)] = \mathbb{E}^{\pi}[R_c(Y_{1:T}) - R_c(X)]$ as the "standard" RL metric. We say that BON *strictly improves* over suboptimal molecule if

$$r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}; X) > r^{\texttt{AP}}(Y_{1:T}, X), \quad \forall j \in [T], \tag{15}$$

for any $Y_{1:T}$ such that $R_c(Y_{1:T}) < \max_{Y'_{1:T}} R_c(Y'_{1:T})$.

SPO can Find the Optimizer. Our first result compare the maximizers of $J(\cdot)$ under the SPO framework to those of $J_0(\pi) = \mathbb{E}^{\pi}[R_c(Y_{1:T})]$.

Lemma 2 *If BON finds a sequence that strictly improves over the current molecule in the sense of* (15)*, any policy* π^* *maximizes* $J(\pi)$ *if and only if it maximizes the original reward* $J_0(\pi)$.

Given the fact that these two optimization targets share the same optimizer, we next study the benefit of using our definition J for gradient update.

Densifying the Reward Signal. We remark that using $J(\pi)$ has the advantage of densifying the reward signal, thus making policy optimization easier. In fact, each $r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}, X)$ serves as a reward signal for choosing the next action Y_{j+1} at state $(Y_{1:j}, X)$.

Lemma 3 *Gradient g defined in (13) can be rewritten as*

$$g = \frac{\lambda}{T}\sum_{t=1}^{T}\mathbb{E}^{\pi}_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:t}\,|\,X) r^{\texttt{AP}}_{\texttt{BON}(t)}(Y_{1:T}, X)\right] + (1-\lambda)\cdot\mathbb{E}^{\pi}_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:T}\,|\,X) r^{\texttt{AP}}(Y_{1:T}, X)\right]. \tag{16}$$

The last term corresponds to the reward at the end of the generation of the molecule, while the first term provides "partial" reward on each generation step. This gradient form suggests that incorporating partial molecule's advantage function densifies the reward signal along the trajectory, enabling the learning agent to explore more efficiently (Riedmiller et al., 2018; Vecerik et al., 2017). We defer the proof of Lemma 2 and Lemma 3 to Appendix A.

Scope of the theoretical results. Lemmas 2 and 3 concern optimizer equivalence and reward-signal densification under the stated assumptions. They do not guarantee that a generated molecule has improved biological efficacy, experimental binding affinity, safety, or properties absent from the reward function.

Scaffold optimization. Building on complete and partial molecular optimization, we further introduce a scaffold-aware optimization term that complements both objectives. The molecular scaffold plays a central role in medicinal chemistry, as it captures the core structural motif that largely determines molecular identity, chemical feasibility, and structure–activity relationships.

To explicitly account for scaffold quality during optimization, we incorporate a scaffold-based reward signal that evaluates the potential of a given scaffold independent of specific side-chain decorations. Concretely, we first extract the Bemis–Murcko scaffold from the complete molecule using RDKit. To assess scaffold value, we leverage the ScaffoldGPT (Liu et al., 2025) model, which conditions on a given scaffold and stochastically generates complete molecules consistent with that scaffold. The quality of the scaffold is then estimated via Monte Carlo sampling over these generated completions.

With this scaffold-aware reward, the advantage preference objective in Eq. (11) is extended as follows:

$$R^{\texttt{AP}}(Y_{1:T}, X) = \lambda\cdot\mathbb{E}_{j\in\mathcal{U}([T])} r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}, X_{1:T}) + \beta\cdot r^{\texttt{AP}}(Y_{1:T}, X_{1:T}) + (1-\lambda-\beta)\cdot r^{\texttt{AP}}_{\text{scaffold}}(Y_{1:T}, X_{1:T}), \tag{17}$$

where $0 \le \alpha + \beta \le 1$ and the scaffold-based advantage term is defined as

$$r^{\texttt{AP}}_{\text{scaffold}}(Y_{1:T}, X_{1:T}) = \mathbb{E}_\rho[R_c(Y_{1:T}) - R_c(X_{1:T})\,|\,X_{\text{scaffold}}, Y_{\text{scaffold}}],$$
$$\text{with } \rho: Y_{j+1:T}\sim \text{ScaffoldGPT}(Y_{\text{scaffold}}), X_{j+1:T}\sim \text{ScaffoldGPT}(X_{\text{scaffold}}).$$

The theory developed for complete and partial molecular optimization extends naturally to scaffold optimization. The combination of complete-, scaffold-, and partial-molecule optimization serves as the default configuration of SPO.

Target	Algorithm	Avg Norm Reward ⋆↑	Avg Top 10 % Norm Reward ↑	Docking ↓	Druglikeliness ↑	Synthesizability ↓	Solubility ↑
3CLPro	**Original**	0.524	0.689	-8.687	0.654	3.097	2.455
(PDBID:	**MMP** (Loeffler et al., 2024)	0.564 ± 0.003	0.680 ± 0.001	-8.184 ± 0.069	0.672 ± 0.003	2.658 ± 0.006	3.114 ± 0.067
7BQY)	**Similarity (≥ 0.5)** (Loeffler et al., 2024)	0.572 ± 0.001	0.686 ± 0.001	-8.158 ± 0.007	**0.686** ± 0.004	2.583 ± 0.010	3.121 ± 0.018
	Similarity ([0.5, 0.7)]) (Loeffler et al., 2024)	0.575 ± 0.002	0.686 ± 0.004	-8.171 ± 0.053	0.676 ± 0.002	2.588 ± 0.018	3.309 ± 0.032
	Similarity (≥ 0.7) (Loeffler et al., 2024)	0.560 ± 0.002	0.677 ± 0.002	-8.187 ± 0.024	0.668 ± 0.007	2.699 ± 0.007	3.120 ± 0.013
	Scaffold (Loeffler et al., 2024)	0.552 ± 0.004	0.678 ± 0.009	-8.081 ± 0.049	0.675 ± 0.002	2.741 ± 0.014	3.002 ± 0.040
	Scaffold Generic (Loeffler et al., 2024)	0.567 ± 0.001	0.680 ± 0.007	-8.078 ± 0.056	0.680 ± 0.005	2.613 ± 0.002	3.173 ± 0.046
	Molsearch (Sun et al., 2022)	0.518 ± 0.002	**0.693** ± 0.003	-8.506 ± 0.038	0.656 ± 0.004	3.110 ± 0.010	2.448 ± 0.032
	MIMOSA (Fu et al., 2021)	0.530 ± 0.003	0.690 ± 0.005	**-8.764** ± 0.048	0.649 ± 0.003	3.148 ± 0.023	2.732 ± 0.027
	DrugEx v3 (Liu et al., 2023c)	0.532 ± 0.003	0.653 ± 0.004	-8.089 ± 0.039	0.583 ± 0.005	3.095 ± 0.018	3.932 ± 0.031
	DrugImproverGPT (Ours)	**0.635** ± 0.003	**0.710** ± 0.003	-8.620 ± 0.028	0.681 ± 0.004	**2.307** ± 0.015	**4.115** ± 0.028
RTCB	**Original**	0.538	0.705	-8.538	0.716	2.984	2.283
(PDBID:	**MMP** (Loeffler et al., 2024)	0.583 ± 0.000	0.700 ± 0.001	-8.466 ± 0.012	0.709 ± 0.001	2.599 ± 0.004	2.978 ± 0.021
4DWQ)	**Similarity (≥ 0.5)** (Loeffler et al., 2024)	0.593 ± 0.004	0.705 ± 0.001	-8.581 ± 0.096	0.715 ± 0.007	2.561 ± 0.005	3.065 ± 0.048
	Similarity ([0.5, 0.7)]) (Loeffler et al., 2024)	0.591 ± 0.002	0.709 ± 0.003	-8.526 ± 0.002	0.710 ± 0.006	2.561 ± 0.001	3.116 ± 0.089
	Similarity (≥ 0.7) (Loeffler et al., 2024)	0.585 ± 0.001	0.705 ± 0.004	-8.584 ± 0.019	0.723 ± 0.000	2.604 ± 0.003	2.841 ± 0.009
	Scaffold (Loeffler et al., 2024)	0.581 ± 0.001	0.701 ± 0.004	-8.524 ± 0.011	0.718 ± 0.003	2.618 ± 0.021	2.840 ± 0.018
	Scaffold Generic (Loeffler et al., 2024)	0.592 ± 0.003	0.704 ± 0.004	-8.590 ± 0.033	0.725 ± 0.001	2.542 ± 0.018	2.916 ± 0.004
	Molsearch (Sun et al., 2022)	0.548 ± 0.002	0.731 ±0.002	-8.750 ± 0.028	0.730 ± 0.003	2.981 ± 0.014	2.290 ± 0.027
	MIMOSA (Fu et al., 2021)	0.553 ± 0.002	0.721 ± 0.002	-8.980 ± 0.039	0.716 ± 0.002	3.066 ± 0.017	2.491 ± 0.019
	DrugEx v3 (Liu et al., 2023c)	0.642 ± 0.002	0.754 ± 0.002	-8.762 ± 0.037	0.583 ± 0.002	2.488 ± 0.015	**5.827** ± 0.017
	DrugImproverGPT (Ours)	**0.695** ± 0.002	**0.791** ± 0.002	**-9.867** ± 0.031	**0.765** ± 0.002	**2.266** ± 0.012	4.012 ± 0.023

Table 1: **Main results.** A comparison of seven baselines including Original, six baselines from REINVENT 4 {MMP, Similarity ≥ 0.5, Similarity $\in [0.5, 0.7)$, Similarity ≥ 0.7, Scaffold, Scaffold Generic}, Molsearch, MIMOSA, DrugEx v3, and DrugImproverGPT (with complete, scaffold and partial-molecule optimization) on multiple objectives based on 3CLPro and RTCB datasets with Tanimoto Similarity above 0.6. The top two results are highlighted as **1st** and 2nd. Results are reported for 5 experimental runs.

6. Experiments

The language model. We utilize the Byte Pair Encoding (BPE) method (Gage, 1994; Sennrich et al., 2015) to initially pre-train our tokenizer using raw SMILES strings and GPT-2-like Transformers for causal language modeling. We train on the standard 11M Drug-like Zinc dataset, excluding entries with empty scaffold SMILES. The dataset is divided into a 90/10 split for training and validation, respectively. (We defer more details to Appendix D.6).

Baselines. In this study, we employ several baseline models, including Molsearch (Sun et al., 2022), a search-driven approach leveraging Monte Carlo Tree Search (MCTS) for molecular generation and optimization, MIMOSA (Fu et al., 2021), a graph-based method for molecular optimization based on sampling and DrugEx v3 (Liu et al., 2023c), a scaffold-based drug optimization using transformer-based reinforcement learning. Moreover, we integrate

the state-of-the-art model, Mol2Mol (He et al., 2021, 2022) from REINVENT 4 (Loeffler et al., 2024), which trains a transformer to adhere to the Matched Molecular Pair (MMP) guidelines (Tyrchan and Evertsson, 2017). Specifically, given a set $\{\{X, Y, Z\}\}$, where X represents the source molecule, Y denotes the target molecule, and Z signifies the property change between X and Y, the model learns a mapping from $\{X, Z\} \in \mathcal{X} \times \mathcal{Z} \implies Y \in \mathcal{Y}$ during training. REINVENT 4 defines six types of property changes for Z, including MMP for user-specified alterations, various similarity thresholds, and scaffold-based modifications where molecules share the same scaffold or a generic scaffold.

Datasets. Utilizing the latest Cancer and COVID dataset proposed in this paper (See Appendix E for more details) for RL fine-tuning across all baseline models and our proposed method, which consists 1 million compounds from the ZINC15 dataset docked to the 3CLPro (PDB ID: 7BQY) protein associated with SARS-CoV-2 and the RTCB (PDB ID: 4DWQ) human cancer protein.

Critics and evaluation metric. In addition to the metrics introduced in structured policy optimization section, we further added the following two metrics: **Average Top 10% Norm Reward:** It is the average of the normalized reward of the top 10% of molecules. **Average Norm Reward:** It is the average of the normalized values of all metrics across valid molecules. This is the most important target metric.

6.1. Experimental results

Table 1 demonstrates that the DrugImproverGPT algorithm outperforms all competing baselines, including the original and six variants from the current leading method, REINVENT 4, across most performance metrics for both viral and cancer-related benchmarks. It improves diversified properties and significantly enhances the critical metric of average normalized reward. DrugImproverGPT excels over REINVENT 4 primarily because REINVENT 4 concentrates on pretraining with restricted similarity and fails to effectively enhance the properties of generated drugs, limiting exploration of potentially high-reward molecular spaces. In contrast, DrugImproverGPT employs SPO to explore high-reward spaces while maintaining reasonable similarity, increasing the probability of generating sequences with positive advantages and decreasing it for negative ones. Additionally, SPO supports optimization at multiple levels—including entire molecules, scaffolds, and partial structures—enabling both global and local optimization. This results in faster convergence and improved performance (see Appendix D.1 for additional experiments on target 6T2W and further details on similarity).

SPO adjusting. Note that the performance curve of Tanimoto similarity in Fig. 2 initially decreases and then increases. This trend aligns ideally with the RL-based molecule generation improvement process. The initial decrease occurs because SPO relaxes the structural constraints of the original molecule to achieve greater improvement in the diversified properties of the generated molecules. 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 reduces the likelihood of drastic structural changes that might result in

Target	Algorithm	Avg Norm Reward ⋆↑	Avg Top 10 % Norm Reward ↑	Docking ↓	Druglikeliness ↑	Synthesizability ↓	Solubility ↑	Validity ↑
3CLPro	**SPO (Without Both)**	0.561	0.666	-8.283	0.614	2.740	3.597	0.902
(PDBID:7BQY)	**SPO (Without Scaffold)**	0.601	0.692	-8.163	0.676	2.381	3.673	0.844
	SPO (Without Partial)	0.585	0.683	-8.248	0.664	2.533	3.520	0.766
	SPO (Ours)	**0.635**	**0.710**	**-8.620**	**0.681**	**2.307**	**4.115**	**0.922**
RTCB	**SPO (Without Both)**	0.592	0.724	-8.318	0.618	2.527	3.832	0.879
(PDBID:4DWQ)	**SPO (Without Scaffold)**	0.694	0.754	-9.462	**0.794**	**2.077**	3.712	**0.964**
	SPO (Without Partial)	0.691	0.788	**-9.918**	0.752	2.291	3.968	0.922
	SPO (Ours)	**0.695**	**0.791**	-9.867	0.765	2.266	**4.012**	0.922

Table 2: **Ablation study.** A comparison of SPO and its individual components for complete-, scaffold-, and partial-molecule optimization. The full SPO consistently outperforms the variants across most metrics.

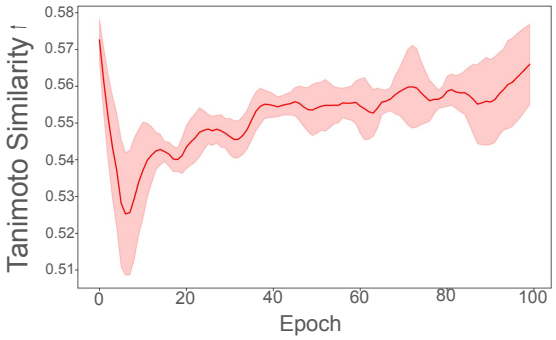

Figure 2: Tanimoto Similarity over five random seeds.

unsynthesizable compounds. This process, illustrated in Figure 2, showcases SPO's capability to automatically adjust the optimization of various properties to achieve global optimization. See Appendix D.2 for details.

Ablation study on SPO. SPO is distinguished from prior RL-based methods by jointly integrating three complementary optimization components: complete-molecule optimization, scaffold-level optimization, and partial-molecule improvement. In Table 2, we conduct a systematic ablation study across these components. The results demonstrate that each component contributes positively to overall performance, and that their combination yields the strongest improvements across nearly all evaluation metrics. In particular, partial-molecule improvement plays a critical role by densifying the reward signal through localized edits, consistent with our theoretical analysis, while scaffold-level optimization provides an intermediate structural constraint that balances global exploration and local refinement. Together, these components enable effective coordination between global and local optimization, leading to faster convergence and superior performance. See Appendix D.3 for additional ablation results on novelty and diversity.

Comparison with improvement-aware GRPO-KL To isolate the effect of the SPO optimizer from the generator, reward formulation, surrogate model, and compute budget,

Method	Avg Norm ↑	Avg Top 10% ↑	Docking ↓	Drug-likeness ↑	Synthesizability ↓	Solubility ↑
Improvement-aware GRPO-KL	0.686	0.786	-9.611	0.721	2.268	**4.347**
SPO (Ours)	**0.695**	**0.791**	**-9.867**	**0.765**	**2.266**	4.012

Table 3: Matched-budget comparison between SPO and improvement-aware GRPO-KL on the RTCB cancer dataset. Both methods use the same pretrained generator, improvement-aware reward, partial- and scaffold-level rewards, surrogate scoring pipeline, and compute budget.

we implement an improvement-aware GRPO-KL baseline. For each input molecule, the baseline uses the same input-anchored improvement reward, complete-molecule reward, partial-molecule reward, scaffold reward, pretrained GPT generator, decoding configuration, docking surrogate, training molecules, and total computation as SPO. The baseline differs only in replacing SPO's advantage-preference policy update with group-relative advantage normalization and explicit KL regularization to the pretrained policy.

Under this matched configuration, SPO achieves higher point estimates for average normalized reward, top-10% normalized reward, docking, drug-likeness, and synthesizability, while GRPO-KL achieves higher solubility. These results suggest that the SPO update can provide benefits beyond the shared generator and structured reward design, although repeated-seed experiments are needed to quantify the statistical robustness of the differences.

7. Discussion

We introduce the DrugImproverGPT framework, comprising a GPT model specialized for drug optimization and SPO, a structured policy optimization algorithm for RL post-training tailored to this task. We provide both rigorous theoretical analysis and comprehensive empirical evaluation of SPO, demonstrating its effectiveness in aligning the GPT model with target objectives while enabling efficient policy gradient updates based on advantage preferences. Specifically, SPO maximizes improvements in desired molecular properties relative to the original compound, while preserving its essential characteristics. We further evaluate SPO on benchmarks involving SARS-CoV-2 and human cancer, where the optimized compounds show substantial improvements over their starting points and outperform current state-of-the-art methods across multiple properties. A matched-budget comparison with improvement-aware GRPO-KL provides additional evidence that the SPO policy update contributes beyond the shared generator and structured reward design. These results support DrugImproverGPT as an in-silico hit-to-lead prioritization framework. Overall, this work opens new avenues for advancing drug optimization and highlights promising directions for future research, such as incorporating graph-based information, which we leave for subsequent exploration.

Limitations DrugImproverGPT is a computational hit-to-lead prioritization framework rather than an experimentally validated drug-discovery pipeline. Its reward functions, including learned docking predictions, QED, LogP, synthetic-accessibility scores, and fingerprint similarity, are imperfect proxies for biological and medicinal-chemistry quality and may be susceptible to reward overfitting or distribution shift. Although sanitization, novelty, unique-

ness, chemical-diversity, and external docking analyses provide complementary robustness evidence, they do not replace retrospective or prospective experimental validation.

The equal-weight aggregate reward is used for benchmark comparability and does not represent an optimal medicinal-chemistry weighting of the objectives. The current evaluation also omits toxicity, pharmacokinetics, selectivity, off-target effects, metabolic stability, and broader ADMET properties. Applying the framework to a new target requires obtaining target-specific docking labels and validating or retraining the docking surrogate. Finally, the SMILES-based generator does not explicitly model three-dimensional protein–ligand interactions. Future work should incorporate structure-aware generators, uncertainty-aware reward models, broader ADMET predictors, retrospective active-compound benchmarks, molecular-dynamics analysis, and prospective wet-lab validation.

Acknowledgements

This work is supported in part 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.

References

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.

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.

Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. MolGPT: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, 2021.

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*, 2022a.

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*, 2022b.

Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7:1–13, 2015.

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

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.

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.

Andres M Bran and Philippe Schwaller. Transformers and large language models for chemistry and drug discovery. *arXiv preprint arXiv:2310.06083*, 2023.

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.

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.

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.

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.

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

Ji Ding, Shidi Tang, Zheming Mei, Lingyue Wang, Qinqin Huang, Haifeng Hu, Ming Ling, and Jiansheng Wu. Vina-gpu 2.0: further accelerating autodock vina and its derivatives with graphics processing units. *Journal of chemical information and modeling*, 63(7): 1982–1998, 2023.

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.

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.

Nathan C Frey, Ryan Soklaski, Simon Axelrod, Siddharth Samsi, Rafael Gomez-Bombarelli, Connor W Coley, and Vijay Gadepally. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5(11):1297–1305, 2023.

Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M Glass, and Jimeng Sun. Mimosa: Multi-constraint molecule sampling for molecule optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 125–133, 2021.

Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. Searching for high-value molecules using reinforcement learning and transformers. *arXiv preprint arXiv:2310.02902*, 2023.

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.

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.

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

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

Jiazhen He, Huifang You, Emil Sandström, Eva Nittinger, Esben Jannik Bjerrum, Christian Tyrchan, Werngard Czechtizky, and Ola Engkvist. Molecular optimization by capturing chemist's intuition using deep neural networks. *Journal of cheminformatics*, 13(1):1–17, 2021.

Jiazhen He, Eva Nittinger, Christian Tyrchan, Werngard Czechtizky, Atanas Patronov, Esben Jannik Bjerrum, and Ola Engkvist. Transformer-based molecular optimization beyond matched molecular pairs. *Journal of cheminformatics*, 14(1):18, 2022.

Jiazhen He, Alessandro Tibo, Jon Paul Janet, Eva Nittinger, Christian Tyrchan, Werngard Czechtizky, and Ola Engkvist. Evaluation of reinforcement learning in transformer-based molecular design. *Journal of Cheminformatics*, 16(1):95, 2024.

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.

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

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.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

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.

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.

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

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

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.

Xuefeng Liu, Songhao Jiang, Archit Vasan, Alexander Brace, Ozan Gokdemir, Thomas Brettin, and Fangfang Xia. Drugimprover: Utilizing reinforcement learning for multi-objective alignment in drug optimization. In *NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development*, 2023a.

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, 2023b.

Xuefeng Liu, Chih-Chan Tien, Peng Ding, Songhao Jiang, and Stevens Rick. Entropy-reinforced planning with large language models for de novo drug discovery. *ICML*, 2024.

Xuefeng Liu, Songhao Jiang, Ian Foster, Jinbo Xu, and Rick Stevens. Scaffoldgpt: A scaffold-based gpt model for drug optimization. *arXiv preprint arXiv:2502.06891*, 2025.

Xuhan Liu, Kai Ye, Herman WT van Vlijmen, Adriaan P IJzerman, and Gerard JP van Westen. Drugex v3: scaffold-constrained drug design with graph transformer-based reinforcement learning. *Journal of Cheminformatics*, 15(1):24, 2023c.

Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

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.

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.

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.

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.

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.

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

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

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

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.

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.

RDkit. RDkit: Open-source cheminformatics software. https://www.rdkit.org. Accessed Oct 2023.

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR, 2018.

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

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

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.

Daniel Rothchild, Alex Tamkin, Julie Yu, Ujval Misra, and Joseph Gonzalez. C5T5: Controllable generation of organic molecules with transformers. *arXiv preprint arXiv:2108.10307*, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

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.

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.

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

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.

Mengying Sun, Jing Xing, Han Meng, Huijun Wang, Bin Chen, and Jiayu Zhou. Molsearch: search-based multi-objective molecular generation and property optimization. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4724–4732, 2022.

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.

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, 2022a.

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, 2022b.

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.

Alexander Telepov, Artem Tsypin, Kuzma Khrabrov, Sergey Yakukhnov, Pavel Strashnov, Petr Zhilyaev, Egor Rumiantsev, Daniel Ezhov, Manvel Avetisian, Olga Popova, et al. Freed++: Improving rl agents for fragment-based molecule generation by thorough reproduction. *arXiv preprint arXiv:2401.09840*, 2024.

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.

Christian Tyrchan and Emma Evertsson. Matched molecular pair analysis in short: algorithms, applications and limitations. *Computational and structural biotechnology journal*, 15:86–90, 2017.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Chenran Wang, Yang Chen, Yuan Zhang, Keqiao Li, Menghan Lin, Feng Pan, Wei Wu, and Jinfeng Zhang. A reinforcement learning approach for protein–ligand binding pose prediction. *BMC bioinformatics*, 23(1):1–18, 2022.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

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.

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

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.

Shenghao Wu, Tianyi Liu, Zhirui Wang, Wen Yan, and Yingxiang Yang. Rlcg: When reinforcement learning meets coarse graining. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation. *Advances in Neural Information Processing Systems*, 34:7924–7936, 2021.

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.

Naruki Yoshikawa, Kei Terayama, Masato Sumita, Teruki Homma, Kenta Oono, and Koji Tsuda. Population-based de novo molecule generation, using grammatical evolution. *Chemistry Letters*, 47(11):1431–1434, 2018.

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.

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.

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.

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

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.

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.

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.

Appendix A. Proofs of the theoretical results

Proof [Proof of Lemma 2] For simplicity, let us take out the shift terms $R_c(X)$ in $r^{\texttt{AP}}$ and $R_c(X_{1:T})$ in $r^{\texttt{AP}}_{\texttt{BON}(j)}$ for a while when defining J. Since the shift term $R_c(X)$ (or its BON counterpart) are independent of the current policy π, such an operation does not influence the definition of optimal policy for J. One can always split $J = \lambda J_{\texttt{BON}} + (1-\lambda) J_0$, where $J_{\texttt{BON}} = \mathbb{E}_\pi \mathbb{E}_{j\in\mathcal{U}([T])} r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}, X)$, $\lambda \in [0,1]$ is a temperature value. Take $\pi^\star$ to be an optimizer of J_0. By definition,

$$J_0(\pi^\star) \geq J(\pi) \geq J_0(\pi)$$

as BON cannot be worse than the current molecule. Thus, any policy that maximizes J_0 should also be a maximizer to J. On the other hand, notice that

$$J_0(\pi^\star) - J(\pi) \geq \lambda \cdot (J_0(\pi^\star) - J_0(\pi)) \geq 0 \tag{18}$$

since the BON reward should not exceed the optimal reward. Hence, any policy that maximize J should also maximize J_0 since the optimizer of J gives optimal value equal to $J_0(\pi^\star)$. Hence, we prove the claim. ■

Proof [Proof of Lemma 3] Denote by π_θ the policy parameterized by θ. When doing policy optimization, we note that

$$\begin{aligned}
g &= \mathbb{E}^\pi_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:T}\,|\,X) R^{\texttt{AP}}(Y_{1:T}, X)\right] \\
&= \mathbb{E}^\pi_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:T}\,|\,X)\left((1-\lambda)\, r^{\texttt{AP}}(Y_{1:T}, X) + \lambda \mathbb{E}_{j\in\mathcal{U}([T])} r^{\texttt{AP}}_{\texttt{BON}(j)}(Y_{1:T}, X)\right)\right] \\
&= \frac{\lambda}{T}\sum_{t=1}^{T} \mathbb{E}^\pi_{X\sim\rho_0}\left[\left(\nabla_\theta \log \pi_\theta(Y_{1:t}\,|\,X) + \nabla_\theta \log \pi_\theta(Y_{t+1:T}\,|\,Y_{1:t}, X)\right) r^{\texttt{AP}}_{\texttt{BON}(t)}(Y_{1:T}, X)\right] \\
&\quad + (1-\lambda)\cdot \mathbb{E}^\pi_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:T}\,|\,X) r^{\texttt{AP}}(Y_{1:T}, X)\right] \\
&= \frac{\lambda}{T}\sum_{t=1}^{T} \mathbb{E}^\pi_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:t}\,|\,X) r^{\texttt{AP}}_{\texttt{BON}(t)}(Y_{1:T}, X)\right] \\
&\quad + (1-\lambda)\cdot \mathbb{E}^\pi_{X\sim\rho_0}\left[\nabla_\theta \log \pi_\theta(Y_{1:T}\,|\,X) r^{\texttt{AP}}(Y_{1:T}, X)\right],
\end{aligned} \tag{19}$$

where the last equality holds by noting that

$$\mathbb{E}^\pi[\nabla_\theta \log \pi_\theta(Y_{t+1:T}\,|\,Y_{1:t}, X)\,|\,Y_{1:t}, X] = 0.$$

■

Appendix B. Related works

Several studies have implemented reinforcement learning (RL) for drug discovery. FREED (Yang et al., 2021) and FREED++ (Telepov et al., 2024) use RL to train a policy network based on the SAC algorithm, redefining the action space as molecule fragments for attachment, which helps identify new fragments to attach and potential sites to form new chemical

bonds. ERP (Liu et al., 2024) integrates Monte Carlo Tree Search (MCTS) with LLMs to enhance sample efficiency in molecule design. Additionally, there is a growing trend of using diffusion (Wang et al., 2024; Watson et al., 2023) for conditional molecule design. However, these approaches adopt existing RL algorithms and focus on De Novo drug discovery, whereas our work concentrates on drug optimization. For drug optimization, REINVENT 4 (Loeffler et al., 2024), ChemRLformer (Ghugare et al., 2023) and MoleculeDesign (He et al., 2024) employ RL to pretrain a GPT model that generates SMILES strings, guiding the generation towards molecules with desirable properties for drug optimization tasks. Both projects use the REINFORCE algorithm. MolDQN (Zhou et al., 2019) adapts the Deep Q-Learning algorithm for GPT-style sequential generation. In contrast, our work introduces a novel structured policy optimization algorithm specifically tailored for drug optimization.

For better positioning of this work, we highlighted the key differences and compared our contribution against a few related works in this domain in the problem setup in Table 4.

	RL Algorithm	Framework	Theoretical Support	GPT Architecture	Drug Optimization Task	New Dataset
REINVENT 4 (Loeffler et al., 2024)	REINFORCE (Williams, 1992)	×	×	✓	✓	×
MOLDQN (Zhou et al., 2019)	Deep Q-Learning	×	×	✓	✓	×
FREED (Yang et al., 2021)	SAC (fragment-based)	×	×	×	De Novo	×
FREED++ (Telepov et al., 2024)	SAC (fragment-based)	×	×	×	De Novo	×
ChemRLformer (Ghugare et al., 2023)	REINFORCE (Williams, 1992)	×	×	✓	✓	×
ERP (Liu et al., 2024)	Entropy-Reinforced Planning	×	×	GPT-guided MCTS	De Novo	×
Molecular Design (He et al., 2024)	REINFORCE (Williams, 1992)	×	×	✓	✓	×
DrugImproverGPT (ours)	Structured Policy Optimization	✓ DrugImproverGPT Workflow	✓	✓	✓	✓ Drug Optimization Dataset

Table 4: Algorithms Characteristics.

Imitation learning. Imitation learning (IL) is a technique where an agent learns by mimicking an expert's actions. IL outperforms pure RL by reducing the complexity and sparsity of the search space (Liu et al., 2023b). Offline IL methods, like behavioral cloning (Pomerleau, 1988), require a dataset of expert trajectories but can lead to errors in the learner's policy. In contrast, interactive IL methods, such as DAgger (Ross et al., 2011) and AggreVaTe (Ross and Bagnell, 2014), use Roll-in-Roll-out (RIRO) scheduling, where learners initially follow their policy but switch to expert guidance for trajectory completion. However, these methods assume constant expert availability, which is impractical, and do not allow returning to previous states once a rollout begins. Our work integrates RIRO with Top-K and Top-P sampling (Liu et al., 2024), creating a guide policy be same as learner policy that conducts roll-outs on any state and estimates returns, improving upon traditional RIRO limitations.

Reinforcement learning. One prominent approach in drug design employs RL (Tan et al., 2022a) 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 (Born et al., 2021; Guimaraes et al., 2017; Neil et al., 2018; Olivecrona et al., 2017; Popova et al., 2018; Ståhl et al., 2019; Tan et al., 2022b; Wang et al., 2022; Zhang et al., 2023; Zhou et al., 2019) or graph-based representations (Atance et al., 2022; Gottipati et al., 2020; Jin et al., 2020; Wu et al., 2022; You et al., 2018). Traditional methods, such as genetic algorithms modified for molecular graphs (Yoshikawa et al., 2018) and Monte Carlo tree search applied to molecular graphs (Jensen, 2019), have been employed. These studies primarily concentrate on the De Novo drug discovery challenge instead of drug optimization. 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. MIMOSA and DrugEx v3 (Fu et al., 2021; Liu et al., 2023c) represents the most recent approaches based on graph structures for drug optimization. However, it falls short of finetuning capability for drug optimization, an issue that our work has successfully addressed.

Appendix C. Limitations of previous work.

1) Prior studies concentrated primarily on the discovery of new drugs from the ground up (Atance et al., 2022; Popova et al., 2018; Zhang et al., 2023). In contrast, we focus on the relatively less explored, yet highly practical and significant, issue of drug optimization. 2) There is currently no RL finetuning algorithm specifically designed for drug optimization problems. 3) The current state-of-the-art model, REINVENT 4, prioritizes pretraining with constrained similarity, thereby restricting its ability to explore molecular spaces with potentially high rewards beyond its training set. Our drug optimization GPT, together with the Structured Policy Optimization approach, addresses these limitations.

Appendix D. Experiments

D.1. Additional experiment on human cancer dataset with PDB: 6t2w

In this section, we conduct an additional experiment on a human cancer dataset with PDB: 6t2w. DrugImproverGPT outperforms competing baselines across most metrics.

Method	Validity ↑	Avg Norm Reward* ↑	Avg Top 10% Norm Reward ↑	Docking ↓	Druglikeliness ↑	Synthesizability ↓	Solubility ↑
mmp	0.916	0.493	0.570	-5.303	0.732	2.549	3.181
similarity	0.938	0.491	0.568	-5.239	0.732	2.518	3.112
medium_similarity	0.875	0.495	0.570	-5.258	0.729	2.493	3.238
high_similarity	0.897	0.485	0.563	-5.192	0.739	2.606	3.025
scaffold	0.838	0.483	**0.570**	-5.246	0.743	2.552	2.792
scaffold_generic	0.895	0.490	0.570	-5.240	0.738	2.511	2.992
Molsearch	**1.000**	0.447	0.555	-5.157	0.728	2.983	2.405
MIMOSA	0.783	0.445	0.556	-5.131	0.735	**3.042**	2.689
DrugImproverGPT (Ours)	**1.000**	**0.496**	0.557	**-5.325**	**0.762**	2.708	**3.429**

Table 5: Comparison of various methods on drug discovery metrics on target PDB: 6t2w. * represents the top-priority target objective.

D.2. Properties improvement associated with Tanimoto Similarity change in Fig. 2

Table 6 demonstrates the diverse properties improvement associated with Tanimoto Similarity change in Fig. 2. The target objective, the average norm reward metric, is gradually improving.

Epoch	Avg Norm Reward	Synthesizability	QED	Docking Score
1	0.5611	2.7371	0.6818	-8.7473
10	0.5609	2.4871	0.6342	-8.4668
20	0.5553	2.4138	0.6855	-8.8338
30	0.5687	2.4528	0.6537	-8.7461
40	0.5732	2.4486	0.6745	-8.9764
50	0.5705	2.4300	0.6944	-8.9911
60	0.5722	2.3943	0.7089	-9.0177
70	0.5725	2.3367	0.7351	-9.0060
80	0.5789	2.4050	0.6973	-9.0684
90	0.5913	2.2744	0.7455	-9.1541
100	0.5932	2.0167	0.8155	-9.7156

Table 6: Performance Metrics Across Epochs

D.3. Ablation study of novelty and diversity

We further explore the novelty and diversity of SPO, both with and without partial molecule enhancements. Novelty is assessed by verifying if the generated molecule/SMILES is present in the original dataset, assigning a value of 0 if it exists and 1 if it does not. Diversity is measured by examining if the generated molecule/SMILES is repeated within the generated dataset, indicating a duplication if two distinct prompts or molecules yield the same outcome. The findings demonstrate that the molecules created through our method are completely new in comparison to the original molecules. Moreover, our technique has attained a decent level of diversity.

Data	Method	Novelty	Diversity
3CLPro	SPO w/o both	1	0.98
	SPO w/o scaffold	1	0.95
	SPO w/o Partial	1	0.98
	SPO	1	0.88
RTCB	SPO w/o both	1	0.98
	SPO w/o scaffold	1	0.69
	SPO w/o partial	1	0.67
	SPO	1	0.81

Table 7: Comparison of SPO with and without partial molecule improvements across cancer and covid datasets in terms of novelty and diversity.

Metric	RTCB	3CLPro
Sanitization pass rate ↑	92.19%	92.19%
Invalid SMILES rate ↓	7.81%	7.81%
Mean Tanimoto similarity to input ↑	0.462	0.509
Novelty ↑	100.00%	100.00%
Unique rate ↑	70.34%	86.44%
Chemical diversity ↑	0.367	0.329

Table 8: Robustness analysis of SPO generations. Novelty measures the fraction of valid generated molecules absent from the reference set; unique rate measures the fraction of nonduplicate valid generations; chemical diversity is one minus the mean pairwise fingerprint similarity.

D.4. Robustness and reward-exploitation analysis

To move beyond the indirect signals (validity, SA, Tanimoto trajectory), we report a direct robustness analysis of SPO's generations: Generations are predominantly valid (92.19% sanitization) and 100% novel relative to the input set, while retaining moderate similarity to the originals (mean Tanimoto around 0.46–0.51), consistent with localized optimization rather than unconstrained drift. If the partial/scaffold rewards were inducing reward hacking, we would expect validity to collapse and novelty/diversity to degrade; instead both remain high. We transparently note that some redundancy remains (moderate pairwise similarity among generations).

D.5. Tanimoto similarity

In our experiment, we set a Tanimoto Similarity (TM) threshold above 0.6, as noted in the caption of Table 1 and further detailed in Table 9. Extremely high TM scores are not desirable, as they may lead to diminished structural diversity. To balance similarity and diversity, we selected 0.6 as a practical threshold to ensure controlled structural variation among the generated molecules.

Table 9: Similarity Comparison of Various Methods on COVID and Cancer Benchmarks

Method	COVID	Cancer
MMP	0.862	0.845
Similarity (≥ 0.5)	0.782	0.760
Similarity (≥ 0.7)	0.881	0.740
Scaffold	0.776	0.875
Scaffold Generic	0.801	0.735
MolSearch	0.969	0.950
MIMOSA	0.959	0.945
DrugEx v3	0.495	0.393
Ours	0.786	0.784

D.6. Pre-training and fine-tuning dataset

We utilized the ZINC dataset, filtering for Standard, In-Stock, and Drug-Like molecules, resulting in approximately 11 million molecules. The new pre-training dataset is constructed by randomly selecting two molecules from the ZINC dataset that meet the proposed criteria (as described in Equation 6). This new pre-training dataset comprises 10 million molecules, with a 90/10 training/validation split.

For fine-tuning, we employ one million compounds from the ZINC15 dataset, docked to the 3CLPro protein (PDB ID: 7BQY), which is linked to SARS-CoV-2, and the RTCB protein (PDB ID: 4DWQ), associated with human cancer. These data are obtained from the latest Cancer and COVID dataset by Liu et al. (2023a) and are used across all baselines.

D.7. Generation with finetuned model

The epoch with highest historical average normalized reward (as detailed in experiments section) is selected for generation. With this epoch and corresponding weights, we apply TOP-PK(Liu et al., 2024) for generation.

D.8. Baseline REINVENT 4

Following are detailed description of six different kinds of property change Z included in REINVENT He et al. (2022, 2021)

- MMP: There are user-defined desirable property changes between molecules X and Y.
- Similarity ≥ 0.5: The Tanimoto similarity between molecules X and Y exceeds 0.5.
- Similarity $\in [0.5, 0.7)$: The Tanimoto similarity for the pair (X, Y) lies between 0.5 and 0.7.
- Similarity ≥ 0.7: The Tanimoto similarity between molecules X and Y exceeds 0.7.
- Scaffold: Molecules X and Y share the same scaffold.
- Scaffold generic: Molecules X and Y share the same generic scaffold.

D.9. Surrogate model

The surrogate model (Vasan et al., 2023) is a simplified variant of a BERT-like transformer architecture, commonly utilized in natural language processing tasks. Within this model, tokenized SMILES strings are initially inputted and subsequently undergo positional embedding. The outputs are then fed into a series of five transformer blocks, each comprising a multi-head attention layer (with 21 heads), a dropout layer, layer normalization with residual connection, and a feedforward network. This feedforward network is composed of two dense layers followed by dropout and layer normalization with residual connection. Following the stack of transformer blocks, a final feedforward network is employed to generate the predicted docking score. The validation r^2 values are 0.842 for 3CLPro and 0.73 for the RTCB dataset.

D.10. Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of a supercomputer, where each node contains 64 CPU cores and 4 A100 GPUs. The training time for each model was approximately 3 hours. We conducted other experiments on a cluster that includes CPU nodes with approximately 280 cores and GPU nodes with approximately 110 Nvidia GPUs, ranging from Titan X to A6000, mostly set up in 4- and 8-GPU configurations. Pretraining utilizes 8 GPUs, while SPO uses a single GPU. Both processes employ either V100 or A100 GPUs. Based on the computing infrastructure, we obtained the wall-time comparison in Table 10 as follows.

Methods	Total Run Time
Pretrain	24h
SPO	8h

Table 10: Wall-time comparison between different methods.

D.11. Hyperparameters and architectures

Table 11 and Table 12 provide a list of hyperparameter settings we used for our experiments. A selection of 1280 molecules from each of the RTCB and 3CLPro datasets, with docking scores ranging from -14 to -6, is used for SPO finetuning and experimentation. This range is based on (Liu et al., 2024). Furthermore, when calculating the average normalized reward for the original molecule, where similarity is not considered, we assign a weight of $[0.25] \times 4$ to docking, druglikeliness, synthesizability, and solubility. Moreover, when the generated SMILES is invalid, meaning that calculating the reward R_c is not possible, we have two options: the first is to directly subtract the reward of the original SMILES (i.e., $-R_c(X)$), or alternatively, we can consider the advantage preference as zero.

Parameter	**Value**
Pretraining	
Learning rate	$5 \times e^{-5}$
Batch size	24
Optimizer	Adam
# of Epochs	10
Model # of Params	$124M$

Table 11: Hyperparameters for pretraining.

D.12. Computational efficiency

DrugImproverGPT offers computational efficiency that is comparable or even slightly superior to REINVENT 4. For instance, processing 100 steps for one molecule takes about 2 minutes with our approach, whereas REINVENT 4 requires 3 minutes. The RL algorithm conducts extensive exploitation and exploration across the vast molecular space. To decrease latency

Parameter	Value
Shared	
# of Molecules Optimized	256
Learning Rate	1×10^{-5}
Optimizer	Adam
# of Epochs for Training	100
Batch size	64
Best-of-N	$[4, 6, 8]$
TopK	$[10, 15, 20]$
TopP	$[0.85, 0.9, 0.95]$
SPO Objective Weight	
Tamimoto Similarity	$[0.2, 0.4, 0.6, 0.8]$
Other Four Objectives	$(1 - W(Sim))/4$
Tamimoto Similarity Settings	
RDKit Fingerprint Generator	rdFingerprintGenerator.GetRDKitFPGenerator
Fingerprint Size	1024
maxPath	7
SPO Other	
Normalize Min/Max	$[-10, 10]$
Advantage preference with invalid generated SMILES	
3CLPro	$[0, -R_c(X)]$
RTCB	$[0, -R_c(X)]$

Table 12: Hyperparameters for SPO.

and make RL feasible, we employ a docking surrogate model. Using AutoDock Vina (Ding et al., 2023) involves converting SMILES to sdf files and then to pdbqt, which further increases latency. As shown in the table below, this conversion process can extend the runtime by a factor of 200.

Table 13: Average Times for Query and Molecule Preprocessing

Method	**Avg time per query**	**Avg time for Molecule Preprocessing**
Ours	<0.01s	Not needed
QuickVina GPU	∼1.5s	∼0.2s

In addition, we used AutoDock Vina for the final evaluation, as shown in the Table 13 below: The result in Table 14 aligned with our surrogate approach that our approach outperforms REINVENT.

Table 14: Docking Scores on Cancer Dataset

Cancer Dataset	Docking Score
REINVENT	-7.9
Ours	-8.7

D.13. Training corpus visualization

For better understanding the training corpus, Fig. 3 shows an example and corresponding visualization towards training corpus described in (6). The two selected molecules/SMILES have the similarity of 0.52.

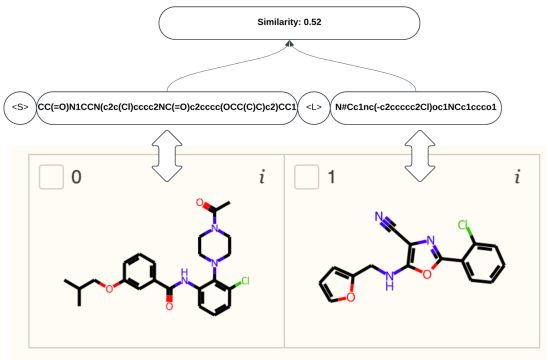

Figure 3: Training corpus example and visualization

D.14. Baselines with RL fine-tuning

The initial version of REINVENT4 (He et al., 2021, 2022) introduced pre-trained models, while the updated version (Loeffler et al., 2024) included Mol2Mol (He et al., 2021, 2022) and allowed RL fine-tuning via REINFORCE (Williams, 1992), but without empirical results. We evaluated REINVENT4 with both pre-training and RL fine-tuning, using the same pre-trained ZINC dataset as our approach. Molsearch employs MCTS for molecular optimization, and MIMOSA (Fu et al., 2021) uses MCMC for efficient sampling. DrugEx v3 (Liu et al., 2023c) leverages graph transformers and reinforcement learning for scaffold-based optimization. Our method outperforms all REINVENT4 variants (Loeffler et al., 2024) in the cancer and COVID-19 datasets, demonstrating superior performance over both the pre-trained and fine-tuned versions of Mol2Mol (He et al., 2021, 2022).

Unlike REINVENT4, which relies on REINFORCE—a standard RL method—our DrugImproverGPT uses the SPO algorithm, designed for targeted drug optimization. By leveraging advantage preference and partial molecule components, our method more effectively optimizes target molecules.

D.15. Binding sites of 3clpro and RTCB

The binding sites of 3CLpro and RTCB are depicted in Fig. 4.

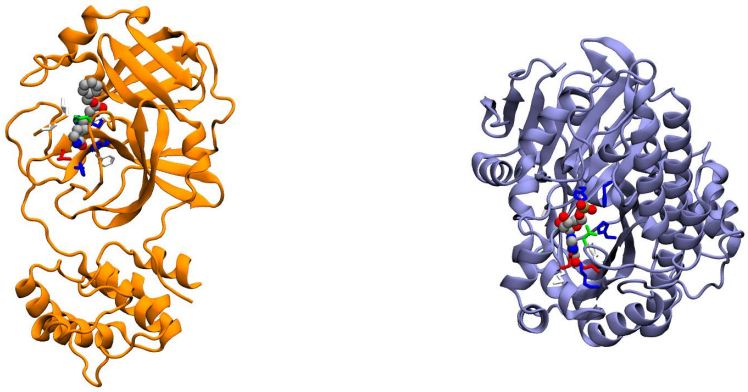

Figure 4: The binding sites of proteins 3CLPro (PDB ID: 7BQY) (**Left**) and RTCB (PDB ID: 4DWQ) (**Right**). Open Eye software are used to identify atoms around the crystallized compound as binding sites.

Appendix E. Dataset details

For each dataset proposed, it is a orderable subset of the ZINC15 dataset. Creating these subsets was mainly a manual process, involving the identification of compounds that are either in stock or can be shipped within three weeks from various suppliers. Subsequently, we performed a random sampling to select 1 million compounds.

For proteins with available structures containing bound ligands, we used X-ray crystallographic data to locate ligand density regions and defined the pocket as a rectangular box enclosing that area. For proteins without bound ligands, we employed FPocket to identify the top-ranked pocket and similarly defined the pocket with a rectangular box around that region. And therefore for each dataset we proposed, only one pocket is used for docking. The validation r^2 values are 0.842 for 3CLPro and 0.73 for the RTCB dataset (two datasets used in experiments section.)

The datasets created in this work including the following files:

- ST_MODEL: The trained surrogate model for SARS-CoV-2 proteins.
- ST_MODEL_rtcb: The trained surrogate model for RTCB Human-Ligase cancer target.
- 24 *.csv files for SARS-CoV-2 proteins under folder data/COVIDRec: The training and validation SMILES string data docked on SARS-CoV-2 receptors as in Table 15
  - PDB-based receptor structures used in molecular docking or structure-based drug discovery. The entries include:
    - 3CLPro (main protease)
    - NPRBD (N-terminal domain of spike RBD)
    - NSP10, NSP15, NSP16, NSP13 (non-structural proteins)
    - PLPro (papain-like protease)

    - RDRP (RNA-dependent RNA polymerase)
  - Each entry follows a format like:
    - [Protein]_[PDB ID]_[Chain(s)]_[Pocket or version ID]_[Binding site]

Receptor	**Receptor**	**Receptor**
3CLPro_7BQY_A_1_F	NPRBD_6VYO_AB_1_F	NPRBD_6VYO_A_1_F
NPRBD_6VYO_BC_1_F	NPRBD_6VYO_CD_1_F	NPRBD_6VYO_DA_1_F
NSP10-16_6W61_AB_1_F	NSP10-16_6W61_AB_2_F	NSP10_6W61_B_1_F
NSP15_6VWW_AB_1_F	NSP15_6VWW_A_1_F	NSP15_6VWW_A_2_F
NSP15_6W01_AB_1_F	NSP15_6W01_A_1_F	NSP15_6W01_A_2_F
NSP15_6W01_A_3_H	NSP16_6W61_A_1_H	Nsp13.helicase_m1_pocket2
Nsp13.helicase_m3_pocket2	PLPro_6W9C_A_2_F	RDRP_6M71_A_2_F
RDRP_6M71_A_3_F	RDRP_6M71_A_4_F	RDRP_7BV1_A_1_F

Table 15: List of SARS-CoV-2 receptors

- 5 *.csv files for human cancer proteins under folder data/CancerRep: The training and validation SMILES string data docked on human cancer proteins including 6T2W, NSUN2, RTCB, WHSC, WRN.
  - the receptor proteins used for docking
    - 6T2W (cancer-relevant protein structure)
    - NSUN2 (RNA methyltransferase implicated in cancer progression)
    - RTCB (RNA ligase involved in RNA repair and stress response)
    - WHSC (Wolf-Hirschhorn syndrome candidate, possibly linked to tumor suppression)
    - WRN (Werner syndrome ATP-dependent helicase, involved in DNA repair and genomic stability)
- Each folder in data/COVIDRec and data/CancerRep includes: model.weights.h5: model weights SMILES*.csv: 1 million SMILES their docking scores. We also provide code within data/SurrogateInf on how to use the surrogate model for inference.
- 3CLPro_7BQY_A_1_F.oeb: The 3CLPro OpenEye receptor file.
- rtcb-7p3b-receptor-5GP-A-DU-601.oedu: The RTCB OpenEye receptor file.
- We include an extended dataset of 1 million SMILES strings from the ZINC15 dataset, their docking scores (as determined by OpenEye FRED) to 24 COVID and 5 cancer-target receptors and surrogate model weights for each corresponding receptor.
- We provide code within data/SurrogateInf on how to use the surrogate model for inference.

E.1. Usage of the proposed dataset in the paper

To further elaborate, this newly proposed dataset is utilized for RL fine-tuning across all baseline models and are not employed in the pretraining phase. For pretraining, we rely on molecules from the ZINC database, as detailed in Appendix D.6. We formed molecular pairs from the ZINC dataset, adhering to the guidelines used for creating the pretraining corpora of each baseline method. The specific rules for generating our molecular pairs are outlined in pretraining section of our approach.