Scaffold-Driven GPT Model for Drug Optimization

Xuefeng Liu [1 *] **Songhao Jiang** [1 *] **Ian Foster** [1 2] **Jinbo Xu** [3] **Rick L. Stevens** [1 2]

Abstract

Drug (lead) optimization has become increasingly crucial in light of fast-mutating virus strains and drug-resistant cancer cells. Nevertheless, it remains challenging as it necessitates retaining the beneficial properties of the original drug while simultaneously enhancing desired attributes beyond its scope. In this work, we aim to tackle this challenge by introducing SCAFFOLDGPT, a novel Generative Pretrained Transformer (GPT) designed for drug optimization based on molecular scaffolds. Our work comprises three key components: (1) A three-stage drug optimization approach that integrates pretraining, finetuning, and decoding optimization. (2) A uniquely designed two-phase incremental training approach for pretraining the drug optimization GPT on molecule scaffold with enhanced performance. (3) A token-level decoding optimization strategy, TOP-N, that enabling controlled, reward-guided generation using pretrained/finetuned GPT. We demonstrate via a comprehensive evaluation on COVID and cancer benchmarks that SCAFFOLDGPT outperforms the competing baselines in drug optimization benchmarks, while excelling in preserving original functional scaffold and enhancing desired properties.

1. Introduction

The rise of rapidly mutating virus strains (Hadj Hassine, 2022), exemplified by those related to SARS-CoV-2 (Yuki et al., 2020), along with drug-resistant cancer cells (Mansoori et al., 2017), has heightened the urgency and interest in accelerating the development of effective treatments. However, traditional De Novo drug discovery processes are prohibitively expensive, often costing from hundreds of millions to billions of dollars (Dickson and Gagnon, 2009), due to their extensive and resource-demanding nature. Despite considerable efforts, the success rate of drug discovery remains low, with many candidates failing early in the development process. This has led to an increased focus on drug repurposing, which leverages existing FDA-approved drugs for new therapeutic uses rather than creating new drugs from scratch. While drug repurposing has seen some success (Pushpakom et al., 2019), its effectiveness is often constrained because drugs are typically developed with a high specificity for a particular disease.

Drug (lead) optimization seeks to overcome the limitations inherent in both De Novo drug discovery and drug repurposing by enhancing an existing FDA-approved drug. Lead optimization is becoming an increasingly vital field, yet it remains relatively underexplored compared to de novo drug discovery and repurposing efforts. DrugImprover (Liu et al., 2023a) has started to clearly define the drug optimization problem by using Tanimoto similarity. Additionally, the DrugImprover framework contributes to the drug optimization domain in three key aspects: a detailed workflow for drug optimization, an Advantage-alignment Policy Optimization (APO) reinforcement learning (RL) algorithm to enhance the multi-objective generative model for drug optimization, and an extensive dataset featuring 1 million ligands and their OEDOCK scores for five proteins related to human cancer cells and 24 high-affinity binding sites on the SARS-CoV-2 protein 3CLPro (PDB ID: 7BQY). The DrugImprover framework features a pretrained generative model that is subsequently refined using the APO reinforcement learning algorithm to ensure the molecules produced align with new objectives. Although DrugImprover has demonstrated encouraging outcomes, its effectiveness is limited due to its dependence on the less complex LSTM network architecture, which might lead to limited scalability and capacity, contextual understanding.

On the other hand, generative pretrained language models have shown exceptional performance in diverse areas, from natural language understanding and generation exemplified by ChatGPT (Wu et al., 2023; Ouyang et al., 2022), to text-to-video conversion as seen in SORA (Liu et al., 2024c), and in programming and code generation through platforms like PG-TD (Zhang et al., 2023a) and Copilot (Nguyen and Nadi,

[*]Equal contribution [1]Department of Computer Science, University of Chicago, Chicago, IL, USA [2]Argonne National Laboratory, Lemont, IL, USA [3]Toyota Technological Institute at Chicago, Chicago, IL, USA. Correspondence to: Xuefeng Liu <xuefeng@uchicago.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

2022). Nevertheless, in the field of drug discovery, tools such as DrugGPT (Li et al., 2023), ChatDrug (Liu et al., 2024a) and ChemGPT (Frey et al., 2023), despite making some initial strides, have not yet achieved performance on par with their counterparts in other domains. The field of drug discovery remains in expectation of breakthroughs similar to the ones achieved by LMs in other domains.

Several challenges have hindered the impact of Language Models (LMs) on drug design: Firstly, the molecules created need to satisfy multiple criteria such as solubility and synthesizability, and must also secure a high docking score against a specific target site. However, existing drug discovery LMs generally only undergo pretraining with molecules and do not focus on enhancing multiple attributes simultaneously. Secondly, given that drugs sharing similar chemical structures tend to exhibit comparable biological or chemical effects (Bender and Glen, 2004), it is crucial to optimize the drug while preserving the beneficial chemical structure of the original molecule. Lastly, the current methodologies in drug discovery LMs primarily focus on maximizing likelihood during the decoding phase, rather than customizing the optimization process to meet specific goals.

REINVENT 4 (He et al., 2021; 2022; Loeffler et al., 2024) achieves the state of art performance in LM-based drug optimization by using the Transformer model and conducting experiments with randomly selected ligand pairs that maintain constrained Tanimoto Similarity. While this method successfully produces ligands with high Tanimoto Similarity, it encounters difficulties in consistently improving the properties of the original drug and tends to simply maximize the likelihood of molecule generation.

In this work, we propose SCAFFOLDGPT, a novel scaffold-based GPT with three-stage optimization process for drug optimization. This optimization process is designed to enhance existing drugs to rapidly evolving virus variants and cancer cells, overcoming the limitation of earlier drug optimization efforts. The contributions of SCAFFOLDGPT are summarized as follows:

• A framework for drug optimization using GPT that involves a three-step optimization process. The underlying motivation is that each stage is complementary to each other, enhances the performance of the preceding one. Furthermore, we have conducted ablation studies to illustrate the performance gains achieved at each step.

• A scaffolds-based GPT with a novel two-phase incremental training specifically designed for drug optimization. The motivation for incremental training is to conduct local optimization before embarking on global optimization by breaking down the whole training corpus into knowledge pieces in an incremental order. We also conducted an ablation study in table 3 to demonstrate the effectiveness of the incremental training.

• A novel token-level decoding optimization strategy, TOP-N, utilizes pretrained GPT to enable controlled, reward-guided generation that aligns with targeted objectives.

• Through extensive experiments and ablation studies on real-world viral and cancer-related benchmarks, we demonstrate that SCAFFOLDGPT outperforms the competing baselines and reliably improves upon existing molecules/drugs in terms of desired targeted objectives while preserving original scaffold, resulting in superior drug candidates.

2. Related Work

2.1. Reinforcement Learning and Finetuning

Reinforcement learning (Tan et al., 2022a) has become a fundamental strategy in drug design (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., 2022a; Zhang et al., 2023b; Zhou et al., 2019), focusing on optimizing rewards that aggregate predicted property scores from various pharmaceutical predictors. Traditional reinforcement learning methods in drug discovery often neglected molecular structure constraints, leading to significant structural changes and unsynthesizable compounds. Our research differs by refining existing drugs to enhance their attributes without redesigning them from scratch and using reinforcement learning to improve a pretrained language model generator, rather than starting a new. We also incorporate methods like Advantage-aligned Policy Optimization (APO) (Liu et al., 2023a), which assigns rewards based on advantage preference over the original molecule, to fine-tune a Transformer model, ensuring it aligns with multiple pharmaceutical objectives while preserving molecular structure. This approach, which includes controllable decoding, refines the model beyond traditional reinforcement learning fine-tuning stages.

2.2. Planning with GPT Models

Several studies have leveraged planning algorithms to enhance text outputs for a variety of NLP tasks. These include approaches like beam search optimization, machine translation improvements (Scialom et al., 2021; Leblond et al., 2021; Chaffin et al., 2021). The PG-TD (Zhang et al., 2023a) method is tailored for code generation using a singular reward function, whereas ERP (Liu et al., 2024b) introduces a novel concept by considering the certainty of each generated token along with an *e*-step forward entropy measurement to gauge potential outcomes. It has been shown that ERP effectively balances exploration and exploitation within molecular structures, leading to the discovery of high-reward molecules. Unlike previous studies that focus solely

on planning with pre-trained language models, our approach incorporates a novel decoding optimization as a critical final step in the algorithm. Moreover, our focus is on optimizing an existing drug rather than creating a De Novo one from the ground up.

2.3. Drug (lead) Optimization

Recent drug optimization efforts have focused primarily on a limited array of drug properties while often disregarding the docking score, an essential metric for evaluating structural compatibility with a target (Zhou et al., 2019; Erikawa et al., 2021). DrugEx v3 (Liu et al., 2023b) seeks to resolve this deficiency by using 3D molecular graphs that encapsulate more comprehensive data such as chemical valence rules. Nevertheless, this method's complexity poses challenges in generating molecules that closely resemble the originals. The resulting lower similarity could account for the diminished efficacy of graph-based methods, as deviations from the original molecular structures result in the loss of vital chemical properties. Conversely, our approach manages to preserve a decent level of similarity.

Diffusion-based efforts (Alakhdar et al., 2024; Morehead and Cheng, 2024) either focus on De Novo drug discovery while neglecting the essential similarity to the original molecule, or overlook the docking score towards a binding target. Molsearch (Sun et al., 2022) is a Monte Carlo tree search (MCTS)-driven approach for molecular generation and optimization, and MIMOSA (Fu et al., 2021) is a GNN sampling-based method leveraging graph-based molecular optimization. DrugImprover (Liu et al., 2023a) effectively begins to redefine the drug optimization challenge by employing reinforcement learning with a mix of multiple objectives as rewards. Still, it employs an LSTM in its generative model, which faces limitations in scalability, capacity, and understanding context. In contrast, our method applies a Transformer as the primary generative model, enhanced further with Advantage-aligned Policy Optimization (APO) for exploratory purposes and an optimized decoder for superior performance.

2.4. Language Models for Drug Discovery

Large language models like MolGPT (Bagal et al., 2021) and ChemGPT (Frey et al., 2023) have been applied to molecular generation using SMILES (Weininger, 1988) and SELFIES (Krenn et al., 2020), showing promise in drug discovery (Rothchild et al., 2021; Wang et al., 2022b). While they outperform traditional methods in some areas, their impact is often limited to minor molecular edits (Murakumo et al., 2023). We build on this by integrating GPTs with reinforcement learning and controlled decoding to meet multiple drug design objectives. Unlike models like REINVENT4 (He et al., 2021; 2022; Loeffler et al., 2024), which mainly rely on pretraining and tend to generate molecules similar to inputs, our method leverages SMILES and scaffolds as prompts to better balance molecular diversity and similarity, potentially improving performance.

3. Preliminaries

In the following sections, we detail MDP, Language Model (LM) and drug discovery, complete with their mathematical notations, and integrate them within the framework of Markov decision processes.

Markov decision processes. Let us define a finite-horizon Markov decision process (MDP) (Puterman, 2014) $\mathcal{M}_0 = \langle \mathcal{S}, \mathcal{A}, T, \mathcal{P}, R \rangle$. In this context, $\mathcal{S}$ represents a finite set of states, while $\mathcal{A}$ comprises a finite set of actions. The term T denotes the planning horizon. The function $\mathcal{P}$, defined as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}'$, describes the deterministic transition dynamics that combine a state s with an action a, with an episode concluding once the agent executes the termination action. Additionally, the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ assigns scores exclusively to complete molecules, assigning a reward of 0 to partial molecules. The effectiveness of a policy can be evaluated using the Q-value function, denoted as $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and defined by the following equation: $Q^\pi(s,a) := \mathbb{E}^\pi \left[\sum_{t=0}^{T} R(s_t, a_t) \mid s_0 = s, a_0 = a \right]$, where the expectation is based on the trajectory determined by the policy π. The associated value function is given by: $V^\pi(s) := \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q^\pi(s,a) \right]$.

LM. We define the state space $\mathcal{S}$ as the set of all possible molecule, where each molecule is represented as a state s that includes a start token [BOS], a molecule with SMILES (Weininger, 1988) representation string, and a termination action [EOS]. We define the set of complete molecules as

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

where $\mathcal{Y}_t \subseteq \mathcal{S}_t|_{t \in [T]}$ represents the hypothesis space at step t (sequence length t), $\mathcal{V}^*$ represents the Kleene closure of Transformer's vocabulary $\mathcal{V}$, with $\mathcal{V} := \mathcal{A}$, and $\circ$ indicating string concatenation. Each action $a \in \mathcal{A}$ is represented as token $y \in \mathcal{V}$. In this work, we train a GPT policy π_θ to acquire prior knowledge for generating valid molecules based on a given set of molecules $\mathcal{B}$. We define the generator policy π_θ, with learned weights θ, as the 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 $\pi_\theta(\cdot|\mathbf{x}, \mathbf{y}_{<t})$ is a distribution, $\mathbf{x}$ is an input sequence, and $\mathbf{y}_{<1} = y_0 :=$ [BOS]. The decoding process in text generation involves identifying the most likely hypothesis by optimizing the objective: $\mathbf{y}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}_T} \log \pi_\theta(\mathbf{y}|\mathbf{x})$.

Drug Optimization. Given an initial drug candidate $X = (x_1, \cdots, x_T)$ and a drug optimization policy π_θ, the goal in drug optimization is to find the optimal policy π_{θ^*} that

Algorithm 1 SCAFFOLDGPT

Require: GPT-based generator policy π_θ; critics $\mathbf{C}$.
Initialize π_θ with GPT2-like Transformer with random weight θ.
▷ /* Stage 1: Pretrain GPT-based generator */
Build the training corpus $C_{\text{Phase 1}}$ (4), $C_{\text{Phase 2}}$ (5).
Pre-train BPE tokenizer and π_θ on $C_{\text{Phase 1}}$ via CLM objective (3).
Pre-train π_θ on $C_{\text{Phase 2}}$.
▷ /* Stage 2: APO fine-tuning */
for $n = 1, \ldots, N$ **do**
  $s_0 \sim \rho_0$, where $\rho_0 \in \Delta(\mathcal{B})$.
  Generate $Y_{1:T} = (y_t, \ldots, y_T) \sim \pi_\theta(\cdot|S)$.
  Compute advantage preference R^{AP} by (9)(10).
  Update generator θ via policy gradient by (11).
▷ /* Stage 3: Token-level Decoding Optimization */
Optimize the generation of π_θ via TOP-N (12) decoding strategy.

maximize the following objective:

$$\pi_{\theta^*} = \arg\max_{\pi_\theta} \mathbb{E}_{X\sim d_0}[R(Y) - R(X)|\theta, X], \quad (2)$$

where $Y = \pi_\theta(\cdot|X), Y_{1:T} = (y_1, \ldots, y_t, \ldots, y_T), y_t \in \mathcal{V}$.

Limitation of previous works: DrugImprover, which utilizes LSTM networks, has limitations in scalability, capacity, and contextual understanding, especially when compared to versions that use GPT Models. The current state of the art, REINVENT 4, employs the Transformer architecture; however, it mainly focuses on pretraining with constrained similarity, which restricts its capability to explore molecular spaces that might offer high rewards beyond its training set. In this work, we address these limitations by proposing SCAFFOLDGPT.

4. The SCAFFOLDGPT Algorithm

Stage 1. Pretrain a GPT generator. Let us note the GPT generator policy as π_θ, which computes the probability p of the occurrence of the tth token in a target molecule Y. It takes into account all preceding tokens $\mathbf{y}_{<t} = [y_1, ..., y_{t-1}]$ in the target, as well as the scaffold compound S, which is noted as $\pi_\theta(y_t|\mathbf{y}_{<t}, S) = p(y_t|\mathbf{y}_{<t}, S)$. The parameters θ of the generator policy π_θ are trained using the training corpus set through the minimization of the negative log-likelihood (NLL) for the complete SMILES strings across the entire set. This process is described as follows:

$$NLL = -\log P(Y|S) = -\sum_{t=1}^{T} \log P(y_t|y_{t-1}, ..., y_1, S)$$
$$= -\sum_{t=1}^{T} \log \pi_\theta(y_t|y_{1:t-1}, S), \quad (3)$$

where T signifies the total number of tokens related to Y. The NLL quantifies the probability of converting a specific scaffold into a designated target molecule.

In this project, we employ pre-training to harness large quantities of unlabeled text to construct a basic foundation model of language understanding. This foundation model can subsequently be fine-tuned and tailored to meet various specialized goals. In this work, we propose a novel framework for pre-training a Transformer and linking scaffolds with complete molecules, based on a SMILES (Simplified Molecular Input Line Entry System) (Weininger, 1988) string representation of the molecule. Here, we define $\mathcal{S}$ as the scaffold space and $\mathcal{Y}$ as the target molecule space. With $\mathcal{P} = \{(s, y)|s, y \in \mathcal{S} \times \mathcal{Y}\}$, we denote the set of scaffold and molecular pairs from $\mathcal{S}$ and $\mathcal{Y}$. In this notation, s represents the scaffold of the target molecule, and y is the corresponding target molecule. We initially pre-trained our tokenizer using the Byte Pair Encoding (BPE) method (Gage, 1994; Sennrich et al., 2015). Building on the pre-trained BPE tokenizer, we propose a two-phase incremental training approach, as illustrated in Fig. 1, to notably enhance the model's ability to improve the validity of inferring the target molecule from its scaffold.

Incremental training. The rationale for incremental training is to conduct local optimization before embarking on global optimization. Therefore, we divide our training into two phases. In the first phase, we focus on training a GPT exclusively for molecules using Causal Language Modeling (CLM). CLM utilizes an autoregressive method where the model is trained to predict the next token in a sequence by considering only the tokens that precede it. The phase 1 corpus is designed as follows:

$$C_{\text{Phase 1}} = \left\{ [BOS], \langle L \rangle, \underbrace{y_1, \cdots, y_T}_{\text{target molecule Y}}, [EOS] \right\}, \quad (4)$$

where $\langle L \rangle$ is the token for ligand.In the second phase, building upon the success of the GPT model developed in phase 1, which demonstrated high accuracy in molecular generation, we advance the training by focusing on pairs of scaffolds and molecules using CLM. The phase 2 corpus, $C_{\text{Phase 2}}$, is

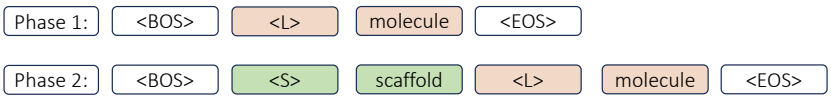


Figure 1: Two-phase incremental training of SCAFFOLDGPT. The first phase concentrates on recognizing the molecule, while the second phase builds connections between its scaffold and the original molecule.

as follows:

$$\left\{[BOS], \langle S\rangle, \underbrace{s_1, \cdots, s_T}_{\text{source scaffold S}}, \langle L\rangle, \underbrace{y_1, \cdots, y_T}_{\text{target molecule Y}}, [EOS]\right\}, \tag{5}$$

where $\langle S\rangle$: scaffold. Consequently, the model can generate the appropriate molecule when given a scaffold. However, since a single scaffold may correspond to multiple molecules, we further refine the GPT-based generator policy, π_θ, to target specific outcomes by applying reinforcement learning finetuning in the next stage.

Stage 2. RL finetuning. Fine-tuning a generative model is crucial for producing outcomes that meet specific objectives. In this study, we use the Advantage-alignment Policy Optimization (APO) (Liu et al., 2023a) algorithm to fine-tune the pretrained GPT-based generator policy π_θ. This approach steers the model from a given scaffold towards the targeted molecule, simultaneously improving multiple properties.

In this work, we adopt the define of reward function from Liu et al. (2024b), and regarded each pharmaceutical property as a critic C and got an ensemble critics $\mathbf{C}$ as follows:

$$[C^{\text{Druglikeness}}, C^{\text{Solubility}}, C^{\text{Synthesizability}}, C^{\text{Docking}}, C^{\text{Tanimoto}}],$$

where each critic $C : Y \to \mathbb{R}$ acts as a distinct evaluator for a specific pharmaceutical attribute. We built the reward function as follow:

$$\begin{aligned} R^{\text{norm}}(Y) &:= R^{\text{norm}}(Y|S) \\ &= \lambda \cdot \text{Norm}\left(C^{\text{Tanimoto}}(S, Y)\right) + \sum_{i=0}^{|\mathbf{C}|-1} \lambda \cdot \text{Norm}\left(C_i(Y)\right), \end{aligned} \tag{6}$$

where Norm is employed to standardize diverse attributes to a consistent scale. In this instance, Norm refers to the process of min-max normalization, which is used to adjust the attributes so they fit within the $[0, 1]$ range. We defer the details of critics to §5.1.

In this work, we use BON (Gao et al., 2023) (Best of N) search to estimate the reward for a prompt (partial molecule). BON can be formulated as the following:

$$\begin{aligned} &\text{BON}\left(\mathbf{y}_{<i}, N, R\right)|_{S,p,k} = \max_{\mathbf{Y}_j \in \{\mathbf{Y}_1, \cdots, \mathbf{Y}_N\}} R\left(\mathbf{Y}_j\right), \\ &\quad \text{where } Y_j = \left[\mathbf{y}_{<i}, y_i, \cdots, y_T\right]_j, \\ &\quad \text{and } y_i \sim \text{TOP-PK}\left(\mathbf{y}_{<i}, p, k\right)|_S. \end{aligned} \tag{7}$$

where TOP-PK (Liu et al., 2024b) defined as follows:

$$\begin{aligned} &\text{TOP-PK}\left(\mathbf{y}_{<i}, p, k\right)|_S = \mathcal{A}_{\mathbf{y}_{<i}}, \\ &\quad \text{where } \mathcal{A}_{\mathbf{y}_{<i}} = \{y_1, \ldots, y_j\}, y_i \in \mathcal{V}, \\ &\quad j = \min\left\{\arg\min_{j'} \sum_{i=1}^{j'} \pi_\theta\left(y_i|S, \mathbf{y}_{<i}\right) \geq p, k\right\}, \\ &\quad \text{and } \pi_\theta\left(y_g|S, \mathbf{y}_{<i}\right) > \pi_\theta\left(y_h|S, \mathbf{y}_{<i}\right), \text{if } g < h, \end{aligned} \tag{8}$$

where $p \in (0, 1]$ represents the maximum cumulative probability, and k denotes the maximum number of candidates for the next tokens.

For each pair consisting of a scaffold and a molecule, we create 8 new molecules from the scaffold using TOP-PK (8) and BON (7) method to select the best one (the one with the highest normalized reward) to serve as the foundation for the final reward calculation.

$$R_c\left(Y|S\right) = R^{\text{norm}}(Y)|_S, \quad Y \in \text{BON}\left(y_0, N, R\right)|_{S,p,k}. \tag{9}$$

APO makes policy gradient based on the advantage preference (Liu et al., 2023a), which is defined as

$$R^{\text{AP}}\left(Y_{1:T}, S\right) = R_c\left(Y_{1:T}\right) - R_c\left(S\right), \tag{10}$$

and perform APO policy gradient with follows:

$$g = \mathbb{E}_{S\sim\rho_0, Y_{1:T}\sim\pi_\theta(\cdot|S)}\left[\nabla_\theta \log \pi_\theta\left(Y_{1:T}|S\right) \cdot R^{\text{AP}}\left(S, Y_{1:T}\right)\right], \tag{11}$$

Stage 3. Token-level Controllable decoding generation. Ultimately, the current GPT-based decoder focuses mainly on maximizing likelihood, neglecting specific metrics of interest. This approach limits its effectiveness in optimizing objectives that diverge from those in its training set, particularly in generating desired molecules. In this study, we introduce controllable decoding after fine-tuning with APO. We present a new approach, TOP-N, to direct the generation process towards enhancements in the optimization objective, as detailed below:

$$\begin{aligned} &Y^\star \sim \left\{\left[\mathbf{y}_{<i}, y_i, \cdots, y_T\right]\right\}, \\ &\quad \text{where } y_i \sim \text{TOP-N}\left(\mathbf{y}_{<i}, p, k, n\right)|_S \\ &\quad \text{TOP-N}\left(\mathbf{y}_{<i}, p, k, n\right)|_S = \mathcal{A}_{\mathbf{y}_{<i}}, \\ &\quad \mathcal{A}_{\mathbf{y}_{<i}} = \{y_1, \ldots, y_n\}, y_i \in \mathcal{V}, |\mathcal{A}| \leq k, \\ &\quad \text{and } R\left(\text{BON}\left(\mathbf{y}_{<i} \circ y_g, N, R\right)|_{S,p,k}\right) \\ &\quad \geq R\left(\text{BON}\left(\mathbf{y}_{<i} \circ y_h, N, R\right)|_{S,p,k}\right), \forall g < h, \end{aligned} \tag{12}$$

Target	Algorithm	Validity ↑	Avg Norm Reward ↑*	Avg Top 10 % Norm Reward ↑	Docking ↓	Druglikeliness ↑	Synthesizability ↓	Solubility ↑	Similarity ↑
3CLPro	**Original**	-	0.533	0.689	-8.698	0.682	3.920	2.471	-
(PDBID:	**MMP** (Loeffler et al., 2024)	0.995 ± 0.001	0.628 ± 0.001	0.718 ± 0.000	-8.259 ± 0.004	0.691 ± 0.001	2.682 ± 0.004	3.109 ± 0.020	0.862 ± 0.000
7BQY)	**Similarity (≥ 0.5)** (Loeffler et al., 2024)	0.995 ± 0.001	0.615 ± 0.000	0.706 ± 0.001	-8.165 ± 0.024	0.697 ± 0.004	2.621 ± 0.006	3.180 ± 0.029	0.782 ± 0.001
	Similarity ([0.5, 0.7]) (Loeffler et al., 2024)	0.995 ± 0.001	0.612 ± 0.001	0.701 ± 0.001	-8.187 ± 0.010	0.691 ± 0.001	2.611 ± 0.009	3.240 ± 0.014	0.756 ± 0.003
	Similarity (≥ 0.7) (Loeffler et al., 2024)	0.995 ± 0.001	0.628 ± 0.001	0.718 ± 0.001	-8.214 ± 0.002	0.691 ± 0.002	2.717 ± 0.002	3.080 ± 0.016	0.881 ± 0.002
	Scaffold (Loeffler et al., 2024)	0.995 ± 0.001	0.602 ± 0.001	0.703 ± 0.002	-8.116 ± 0.002	0.695 ± 0.001	2.728 ± 0.008	2.968 ± 0.038	0.776 ± 0.001
	Scaffold Generic (Loeffler et al., 2024)	0.994 ± 0.001	0.617 ± 0.001	0.710 ± 0.002	-8.179 ± 0.012	0.701 ± 0.000	2.645 ± 0.008	3.090 ± 0.029	0.801 ± 0.000
	DrugImprover (Liu et al., 2023a)	0.884 ± 0.005	0.432 ± 0.002	0.493 ± 0.005	-6.726 ± 0.007	0.506 ± 0.002	**1.306** ± 0.010	2.057 ± 0.011	0.531 ± 0.002
	Molsearch (Sun et al., 2022)	**1.000** ± 0.001	0.616 ± 0.001	0.726 ± 0.002	-8.855 ± 0.040	0.686 ± 0.001	3.105 ± 0.006	2.452 ± 0.008	**0.969** ± 0.001
	MIMOSA (Fu et al., 2021)	0.985 ± 0.008	0.622 ± 0.001	0.734 ± 0.002	-8.800 ± 0.015	0.677 ± 0.004	3.105 ± 0.008	2.711 ± 0.010	0.959 ± 0.001
	DrugEx v3 (Liu et al., 2023b)	**1.000** ± 0.001	0.524 ± 0.001	0.613 ± 0.001	-8.089 ± 0.013	0.583 ± 0.002	3.095 ± 0.005	3.932 ± 0.008	0.495 ± 0.001
	SCAFFOLDGPT (w/o APO & TOP-N)	0.951 ± 0.004	0.587 ± 0.004	0.693 ± 0.004	-8.238 ± 0.101	0.659 ± 0.014	2.865 ± 0.038	2.999 ± 0.163	0.754 ± 0.005
	SCAFFOLDGPT (w/o TOP-N)	0.857 ± 0.061	0.627 ± 0.009	0.717 ± 0.004	-8.583 ± 0.075	0.727 ± 0.019	2.566 ± 0.088	3.388 ± 0.095	0.717 ± 0.028
	SCAFFOLDGPT (w/o APO)	0.998 ± 0.001	0.666 ± 0.000	0.740 ± 0.001	-9.312 ± 0.018	0.734 ± 0.002	2.698 ± 0.006	3.676 ± 0.006	0.813 ± 0.002
	SCAFFOLDGPT	0.944 ± 0.094	**0.675** ± 0.031	**0.740** ± 0.015	**-9.343** ± 0.440	**0.746** ± 0.028	2.453 ± 0.154	**3.913** ± 0.358	0.745 ± 0.032
RTCB	**Original**	-	0.536	0.698	-8.572	0.709	3.005	2.299	-
(PDBID:	**MMP** (Loeffler et al., 2024)	0.998 ± 0.001	0.636 ± 0.000	0.731 ± 0.001	-8.465 ± 0.021	0.709 ± 0.001	2.599 ± 0.004	3.013 ± 0.013	0.845 ± 0.001
4DWQ)	**Similarity (≥ 0.5)** (Loeffler et al., 2024)	0.999 ± 0.001	0.626 ± 0.000	0.723 ± 0.001	-8.511 ± 0.012	0.713 ± 0.002	2.543 ± 0.002	3.082 ± 0.031	0.760 ± 0.000
	Similarity ([0.5, 0.7]) (Loeffler et al., 2024)	0.999 ± 0.001	0.622 ± 0.001	0.718 ± 0.000	-8.486 ± 0.021	0.713 ± 0.003	2.542 ± 0.005	3.101 ± 0.005	0.740 ± 0.001
	Similarity (≥ 0.7) (Loeffler et al., 2024)	0.999 ± 0.001	0.639 ± 0.000	0.734 ± 0.001	-8.496 ± 0.009	0.718 ± 0.001	2.628 ± 0.001	2.868 ± 0.003	0.875 ± 0.002
	Scaffold (Loeffler et al., 2024)	0.998 ± 0.001	0.609 ± 0.001	0.718 ± 0.000	-8.508 ± 0.026	0.711 ± 0.000	2.627 ± 0.002	2.803 ± 0.010	0.735 ± 0.002
	Scaffold Generic (Loeffler et al., 2024)	0.998 ± 0.001	0.625 ± 0.001	0.722 ± 0.000	-8.544 ± 0.009	0.722 ± 0.002	2.551 ± 0.010	2.898 ± 0.005	0.768 ± 0.004
	DrugImprover (Liu et al., 2023a)	0.920 ± 0.008	0.478 ± 0.001	0.618 ± 0.002	-8.701 ± 0.037	0.486 ± 0.002	**1.181** ± 0.010	2.026 ± 0.013	0.427 ± 0.001
	Molsearch (Sun et al., 2022)	**1.000** ± 0.001	0.625 ± 0.001	0.742 ± 0.001	-8.747 ± 0.009	0.719 ± 0.001	3.012 ± 0.004	2.273 ± 0.005	**0.950** ± 0.001
	MIMOSA (Fu et al., 2021)	0.989 ± 0.001	0.631 ± 0.001	0.749 ± 0.001	-8.972 ± 0.011	0.706 ± 0.003	3.080 ± 0.007	2.561 ± 0.008	0.945 ± 0.001
	DrugEx v3 (Liu et al., 2023b)	**1.000** ± 0.001	0.592 ± 0.001	0.668 ± 0.001	-8.762 ± 0.010	0.583 ± 0.002	2.488 ± 0.005	5.827 ± 0.010	0.393 ± 0.001
	SCAFFOLDGPT (w/o APO & TOP-N)	0.956 ± 0.004	0.582 ± 0.007	0.700 ± 0.008	-8.214 ± 0.125	0.686 ± 0.017	2.788 ± 0.056	2.781 ± 0.214	0.707 ± 0.005
	SCAFFOLDGPT (w/o TOP-N)	0.811 ± 0.074	0.639 ± 0.004	0.723 ± 0.005	-8.808 ± 0.071	0.741 ± 0.013	2.521 ± 0.081	3.279 ± 0.067	0.730 ± 0.030
	SCAFFOLDGPT (w/o APO)	0.997 ± 0.001	0.673 ± 0.001	0.755 ± 0.001	-9.659 ± 0.023	0.764 ± 0.001	2.606 ± 0.007	3.481 ± 0.027	0.773 ± 0.003
	SCAFFOLDGPT	0.826 ± 0.100	**0.682** ± 0.004	**0.756** ± 0.003	**-9.757** ± 0.057	**0.765** ± 0.013	2.437 ± 0.059	**3.582** ± 0.043	0.747 ± 0.026

Table 1: **Main results.** A comparison of eight baselines including Original, six baselines from REINVENT {MMP, Similarity (≥ 0.5), Similarity $\in [0.5, 0.7)$, Similarity ≥ 0.7, Scaffold, Scaffold Generic}, DrugImprover, Molsearch, MIMOSA and different versions of SCAFFOLDGPT on various objectives based on 3CLPro and RTCB datasets. The top two results are highlighted as **1st** and 2nd. * represents the top-priority target objective. Results are reported for five experimental runs.

where $n \leq k$ denotes as top N candidate of next tokens with regard to BON function.

Remark 4.1. TOP-N differs from TOP-P, TOP-K, and TOP-PK in that it is measured based on maximum reward, whereas the others are measured based on maximum likelihood. TOP-N is also distinct from BON in that BON is optimized at the sequence level, while TOP-N is optimized at the token level.

5. EXPERIMENTS

5.1. Experimental configuration

The language model. We use GPT-2-like Transformers for causal language modeling, employing the standard 11M Drug-like Zinc dataset for training. Entries with empty scaffold SMILES are excluded, and we adopt a 90/10 split for training and validation, respectively. The training process is structured into three phases: pretraining, fine-tuning, and decoding optimization, as outlined in Algorithm SCAFFOLDGPT (See appendix for more details).

Baselines. We compare against baseline models: DrugImprover (Liu et al., 2023a), which utilizes an LSTM-based generator with APO fine-tuning; Molsearch (Sun et al., 2022), a Monte Carlo tree search (MCTS)-driven approach for molecular generation and optimization; MIMOSA (Fu et al., 2021), a sampling-based method leveraging graph-based molecular optimization; and DrugEx v3 (Liu et al., 2023b), which leverages transformer-based reinforcement learning for scaffold-guided drug optimization. Additionally, we incorporate the model proposed by He et al. (2021; 2022); Loeffler et al. (2024), which trains a transformer to adhere to the Matched Molecular Pair (MMP) guidelines (Kenny and Sadowski, 2005; 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. Here, $\mathcal{X} \times \mathcal{Z}$ denotes the input space, and $\mathcal{Y}$ denotes the target space. They defined six different 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. More specifically,

- MMP: there exists user-specified property changes between molecule X and Y.
- Similarity $\geq$ 0.5: tanimoto similarity between molecule X and Y is larger than 0.5.
- Similarity $\in [0.5, 0.7)$: the tanimoto similarity of pair (X, Y) is between 0.5 and 0.7.
- Similarity $\geq$ 0.7: tanimoto similarity between molecule X and Y is larger than 0.7.
- Scaffold: molecule X and Y share same scaffold.
- Scaffold generic: molecule X and Y share same generic scaffold.

All baseline models are fine-tuned on the cancer and COVID datasets following their respective fine-tuning methods.

Dataset. We employ, from the most recent Cancer and COVID dataset of Liu et al. (2023a), 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 this study, we evaluate the efficacy of SCAFFOLDGPT in generating molecules

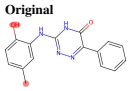
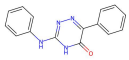
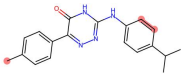
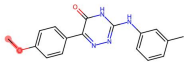
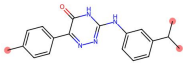
	Original	Scaffold	REINVENT	SCAFFOLDGPT 1	SCAFFOLDGPT 2	SCAFFOLDGPT 3
Molecule						
SMILE String	Cc1ccc(O)c(Nc2nnc (-c3ccccc3)c(=O)[nH]2)c1		CCOc1ccc(O)c(Nc2nnc (-c3ccccc3)c(=O)[nH]2)c1	Cc1ccc(-c2nnc(Nc3ccc (C(C)C)cc3)[nH]c2=O)cc1	COc1ccc(-c2nnc(Nc3cccc (C)c3)[nH]c2=O)cc1	Cc1ccc(-c2nnc(Nc3cccc (C(C)C)c3)[nH]c2=O)cc1
Scaffold	O=c1[nH]c(Nc2cc ccc2)nnc1-c1ccccc1		c1ccc(Nc2nnc(-c3 ccccc3)[nH]2)cc1	same as original	same as original	same as original
Similarity	-		0.878	0.826	0.911	0.866
Docking (↓)	-10.031		-9.258	-11.478 ✓	-11.474 ✓	-11.087 ✓
Druglikeness (↑)	0.646		0.624	0.762 ✓	0.774 ✓	0.762 ✓
Synthesizability (↓)	2.390		2.417	2.298 ✓	2.257 ✓	2.356 ✓
Solubility (↑)	2.590		2.680 ✓	4.007 ✓	2.893 ✓	4.007 ✓
Avg Norm Reward (↑)*	0.618		0.589	0.759 ✓	0.753 ✓	0.754 ✓

Table 2: One optimization example from cancer benchmark. Every generated molecules retains the scaffold, with all desired properties improved compared to the original. * represents the top-priority target objective. ✓ indicates improved property.

with desirable attributes within the context of pharmaceutical drug discovery. We leverage the RDKit (Landrum et al.) chemoinformatics package and employ various performance metrics as follows: **Validity** measures if the generated SMILES is valid in syntax. **Druglikeness** measures the likelihood of a molecule being a suitable candidate for drug development. **Solubility** assesses the likelihood of a molecule's ability to mix with water, commonly referred to as the water-octanol partition coefficient (LogP). **Synthetizability** quantifies the ease (score of 1) or difficulty (score of 10) associated with synthesizing a given molecule (Ertl and Schuffenhauer, 2009). **Docking Score** assesses the drug's potential to bind and inhibit the target site. To enable efficient computation, we employ a docking surrogate model (See Appendix A.4) to output this score. **Similarity:** We use Tanimoto similarity to evaluate the similarity between original SMILES and generated SMILES. **Average Top 10% Norm Reward** is the average of the normalized reward of the top 10% of molecules based on their average normalized reward. **Average Norm Reward** is the average of the normalized values of the docking score, druglikeness, synthesizability, solubility, and similarity across all valid molecules. This is the most important metric. Evaluations are based on a sample of 1,280 molecules.

Remark 5.1. A similarity score that is too high results in low structural diversity, while a score that is too low suggests the molecules have drifted too far from the original. Neither extreme is desirable. Our goal is to achieve a balanced level of similarity with meaningful variation. In this work, the primary optimization objective is the average normalized reward.

5.2. Main results.

Table 1 shows that SCAFFOLDGPT surpasses DrugImprover and six different versions of REINVENT4 in performance measures for both virus-related and cancer-related proteins. Moreover, SCAFFOLDGPT exceeds the performance of all baseline methods and also demonstrates a decent level of Tanimoto similarity to the original drug, indicating that it preserves the advantageous features of the original drugs while improving desired properties.

Several key factors contribute to this superior performance. Although DrugImprover established a strong foundation for the drug optimization field, including a workflow and a reinforcement learning algorithm to align the generative model with multiple pharmaceutical objectives, SCAFFOLDGPT outshines DrugImprover in all benchmarks. This is because SCAFFOLDGPT employs a GPT-2-like Transformer as the basis of its generative model, whereas DrugImprover relies solely on LSTM. Consequently, the GPT-2 Transformer grants SCAFFOLDGPT enhanced scalability, capacity, and contextual understanding compared to DrugImprover.

In contrast to the current state-of-the-art approach, REINVENT4, which pre-trains a Transformer with constraints on Tanimoto similarity, their method falls short in achieving drug optimization as it overlooks the optimization of multiple pharmaceutical properties. Therefore, Table 1 reveals that although REINVENT4 achieved high similarity, the generated molecules often failed to surpass the original ones. SCAFFOLDGPT, on the other hand, employs the APO reinforcement learning algorithm to fine-tune the pre-trained generative model and utilizes the TOP-N decoding optimization strategy. These approaches ensure improvements aligned with multiple pharmaceutical objectives and enable SCAFFOLDGPT to successfully enhance the original drug across various pharmaceutical properties while maintaining a high Tanimoto similarity.

5.3. Ablation studies.

Effectiveness of (two-phase) incremental training. In two-phase incremental pretraining, the intuition behind the first phase lies in training on critical keywords as knowledge pieces, reinforcing the memory of these key terms, particularly

1024 SMILES	Details
1024	> 100 length, > 50 scaffold
Validity	
One phase	0.57
Two phases	0.68

Figure 2: Ablation study of one-phase vs two-phases. Evaluations are based on five random seeds.

Model	Diversity ↑	Validity ↑	Avg Norm Reward ↑*	Avg Top 10% Norm Reward ↑	Docking ↓	Druglikeliness ↑	Synthesizability ↓	Solubility ↑	Similarity ↑
COVID									
ScaffoldGPT (w/o Top-N) Top-K	0.988	**0.947**	0.645	0.727	-8.726	**0.760**	2.418	3.499	0.697
ScaffoldGPT (w/o Top-N) Top-P	0.988	0.938	0.642	0.722	-8.653	0.756	2.420	3.505	0.696
ScaffoldGPT (w/o Top-N) Top-PK	**0.989**	0.941	0.643	0.724	-8.667	0.759	**2.407**	3.506	0.692
ScaffoldGPT (Top-N)	0.965	0.944	**0.675**	**0.740**	**-9.343**	0.746	2.453	**3.913**	**0.745**
CANCER									
ScaffoldGPT (w/o Top-N) Top-K	0.912	0.709	0.648	0.728	-8.944	0.756	2.456	3.258	0.730
ScaffoldGPT (w/o Top-N) Top-P	**0.931**	0.704	0.645	0.729	-8.907	0.756	2.466	3.226	0.723
ScaffoldGPT (w/o Top-N) Top-PK	0.926	0.719	0.645	0.727	-8.888	0.757	2.466	3.219	0.725
ScaffoldGPT (Top-N)	0.916	**0.826**	**0.682**	**0.756**	**-9.757**	**0.765**	**2.437**	**3.582**	**0.747**

Table 3: Ablation study of Top-N, Top-P, Top-K and Top-PK sampling strategies. The top result is highlighted as **1st**. ⋆ represents the top-priority target objective. Evaluations are based on five random seeds. Top-N outperforms others in most of the metrics.

in longer sequences. We conducted an ablation study comparing our novel two-phase incremental training with conventional one-phase training in Table 2. To ensure a fair comparison in terms of total training epochs, we trained for 10 epochs in the conventional single-phase setting, and for five epochs per phase in our two-phase setting. The results showed that the two-phase approach improves validity compared to one-phase training, demonstrating the effectiveness of the incremental training method.

Effectiveness of APO Finetuning. ScaffoldGPT adopts APO finetuning as the second step, following the completion of pretraining the GPT-based generator. Table 1 demonstrates the effectiveness of APO through two comparisons: ScaffoldGPT (w/o APO, Top-N) vs. ScaffoldGPT (w/o Top-N), which shows that after applying APO finetuning, performance improved on most properties. Additionally, ScaffoldGPT vs. ScaffoldGPT (w/o APO) validates the importance of the APO component. By applying APO on top of pretraining and Top-N decoding, performance improved. Both cases demonstrate the effectiveness of APO finetuning. Note that APO may compromise certain reward metrics, such as similarity or validity, if this trade-off leads to improved performance in the target weighted objective.

Effectiveness of Top-N decoding strategy. ScaffoldGPT adopts the Top-N decoding strategy as the final step followed by APO finetuning. Table 1 demonstrates the effectiveness of Top-N through two comparisons: ScaffoldGPT (w/o APO, Top-N) vs. ScaffoldGPT (w/o APO), showing that after applying the Top-N decoding strategy on top of pretrained GPT, performance improved across most properties. Moreover, ScaffoldGPT vs. ScaffoldGPT (w/o Top-N) illustrates that after applying APO on top of pretraining and RL, performance still improves on multiple attributes, surpassing all baselines. Furthermore, by comparing ScaffoldGPT (w/o APO) and ScaffoldGPT (w/o Top-N), we observe that applying Top-N decoding alone enhances performance more than applying APO alone.

Ablation study of Top-N vs Top-P, Top-K and Top-PK strategies. Following the setup described in §5.1, we perform an ablation study on sampling strategies. When removing Top-N component, we employ multinomial generation, where multinomial sampling randomly selects the next token from the entire vocabulary based on the model's probability distribution. In the ablation study detailed in Table 3, we examined Top-N over Top-P, Top-K, and Top-PK with K=20 and P=0.95. The results indicate that Top-N surpasses the other strategies in most metrics.

Drug optimization illustration. Finally, we provide three examples illustrating the effectiveness of ScaffoldGPT in improving upon the original molecule on the cancer benchmark, as shown in Table 2 (Refer to Appendix A.5 for the COVID benchmark). The results in Table 2 demonstrate that the drugs generated by ScaffoldGPT outperform the original drugs across all desired properties. Additionally, the comparison figure in Table 2 illustrates that the improved molecules preserve the original drug to a significant extent, with only minor changes highlighted in red. The results indicate that ScaffoldGPT effectively optimizes desired properties while preserving the beneficial properties of the original drug.

6. Conclusion

We have introduced ScaffoldGPT, a novel framework for drug optimization. This framework incorporates a unique scaffold-based GPT design, a three-step optimization process, a two-phase incremental training method, and a novel Top-N decoding strategy that facilitates controlled reward-guided generation using GPT. To showcase the superior performance of ScaffoldGPT, we conduct evaluations on real-world viral and cancer-related datasets to compare it against eight competing baselines. Our results demonstrate that ScaffoldGPT surpasses all baselines across the majority of performance metrics, underscoring its efficacy. Our work highlights ScaffoldGPT's effectiveness in drug optimization, as evidenced by enhancements in various pharmaceutical properties. Currently, ScaffoldGPT is limited to handling molecules in SMILES format. We are working to expand ScaffoldGPT's capabilities to accommodate a broader range of drug representation formats as a future direction.

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

Amira Alakhdar, Barnabas Poczos, and Newell Washburn. Diffusion models in de novo drug design. *Journal of Chemical Information and Modeling*, 64(19):7238–7256, 2024. 3

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. 3

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

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

Antoine Chaffin, Vincent Claveau, and Ewa Kijak. PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding. *arXiv preprint arXiv:2109.13582*, 2021. 2

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

Daiki Erikawa, Nobuaki Yasuo, and Masakazu Sekijima. Mermaid: an open source automated hit-to-lead method based on deep reinforcement learning. *Journal of Cheminformatics*, 13:1–10, 2021. 3

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. 7

Argonne Leadership Computing Facility. https://www.alcf.anl.gov/polaris, last accessed on 10-2-2023. 13

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. 2, 3

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. 3, 6

Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994. 4

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023. 5

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

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

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. 2, 3, 6

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. 2, 3, 6

Peter W Kenny and Jens Sadowski. Structure modification in chemical databases. *Chemoinformatics in drug discovery*, pages 271–285, 2005. 6

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. 3

Greg Landrum et al. RDkit: Open-source cheminformatics software. https://www.rdkit.org. Accessed Oct 2023. 7

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis

Antonoglou, Karen Simonyan, and Oriol Vinyals. Machine translation decoding beyond beam search. *arXiv preprint arXiv:2104.05336*, 2021. 2

Yuesen Li, Chengyi Gao, Xin Song, Xiangyu Wang, Yungang Xu, and Suxia Han. Druggpt: A gpt-based strategy for designing potential ligands targeting specific proteins. *bioRxiv*, pages 2023–06, 2023. 2

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Conversational drug editing using retrieval and domain feedback. In *The twelfth international conference on learning representations*, 2024a. 2

Xuefeng Liu, Songhao Jiang, Archit Vasan, Alexander Brace, Ozan Gokdemir, Thomas Brettin, Fangfang Xia, Ian Foster, and Rick Stevens. 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. 1, 2, 3, 5, 6, 12

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*, 2024b. 2, 5, 13

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, 2023b. 3, 6

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024c. 1

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. 2, 3, 6

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

Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024. 3

Kusuri Murakumo, Naruki Yoshikawa, Kentaro Rikimaru, Shogo Nakamura, Kairi Furui, Takamasa Suzuki, Hiroyuki Yamasaki, Yuki Nishigaya, Yuzo Takagi, and Masahito Ohue. LLM drug discovery challenge: A contest as a feasibility study on the utilization of large language models in medicinal chemistry. In *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*, 2023. 3

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

Nhan Nguyen and Sarah Nadi. An empirical evaluation of github copilot's code suggestions. In *Proceedings of the 19th International Conference on Mining Software Repositories*, pages 1–5, 2022. 1

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. 2

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 1

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018. 2

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

Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 3

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. 3

Thomas Scialom, Paul-Alexis Dray, Jacopo Staiano, Sylvain Lamprier, and Benjamin Piwowarski. To beam or not to beam: That is a question of cooperation for language GANs. *Advances in Neural Information Processing Systems*, 34:26585–26597, 2021. 2

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 4

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. 2

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. 3, 6

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. 2

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. 2

Christian Tyrchan and Emma Evertsson. Matched molecular pair analysis in short: algorithms, applications and limitations. *Computational and structural biotechnology journal*, 15:86–90, 2017. 6

Archit Vasan, Rick Stevens, Arvind Ramanathan, and Vishwanath Venkatram. Benchmarking language-based docking models. 2023. 12

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, 2022a. 2

Wenlu Wang, Ye Wang, Honggang Zhao, and Simone Sciabola. A transformer-based generative model for de novo molecular design. *arXiv preprint arXiv:2210.08749*, 2022b. 3

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. 3, 4

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10 (5):1122–1136, 2023. 1

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

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B Tenenbaum, and Chuang Gan. Planning with large language models for code generation. *arXiv preprint arXiv:2303.05510*, 2023a. 1, 2

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, 2023b. 2

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. 2, 3

A. Appendix

A.1. Pre-training and finetuning dataset

For pretrainig, We used the ZINC dataset, filtering for Standard, In-Stock, and Drug-Like molecules, resulting in approximately 11 million molecules. For preprocessing, we perform a few straightforward steps:
1. We first canonicalize smiles Chem.MolToSmiles(Chem.MolFromSmiles(mol),True).
2. We filter out molecules whose scaffold SMILES is an empty string. These preprocessing steps are also included in the huggingface data repo.

For finetuning, we utilize 1 million compounds from the ZINC15 dataset, docked to the 3CLPro protein (PDB ID: 7BQY) linked to SARS-CoV-2 and the RTCB protein (PDB ID: 4DWQ) associated with human cancer, as sourced from the latest Cancer and COVID dataset by Liu et al. (2023a), across all baselines.

A.2. Generation with finetuned model

The top five epochs with the highest historical average normalized reward (as detailed in Section 5.1) are selected. From these five epochs, the epoch with the highest product of validity and average normalized reward is chosen as the final model for generation.

With this epoch and corresponding weights, we apply the proposed decoding method (as described in section 4) for generation.

A.3. BPE Tokenization

The Byte Pair Encoding (BPE) algorithm involves the following steps:

1. **Initialize the Vocabulary:** Start with a base vocabulary consisting of all individual characters in the text corpus.
2. **Count Frequencies:** Count the frequency of all character pairs in the text.
3. **Merge Most Frequent Pair:** Identify the most frequent pair of characters and merge them into a single token. Add this new token to the vocabulary.
4. **Update Text:** Replace all occurrences of the most frequent pair with the new token in the text.
5. **Repeat:** Repeat the process of counting frequencies, merging pairs, and updating the text until the desired vocabulary size is reached or no more merges are possible.

BPE constructs a robust vocabulary by iteratively merging the most frequent token pairs, effectively capturing common subword units for more efficient and flexible text representation. The resulting vocabulary comprises 3,152 tokens and includes special tokens as well. For instance, the sequence <L> is tokenized into three separate tokens: $<$, L, and $>$. The tokenizer was trained on 10 million molecules from the ZINC dataset, ensuring comprehensive coverage of chemical elements.

A.4. Surrogate model

The surrogate model (Vasan et al., 2023) is a simplified variant of a BERT-like transformer, extensively utilized in natural language processing. In this model, tokenized SMILES strings are inputted and then embedded with positional information. The resulting outputs are subsequently fed into a series of five transformer blocks, each comprising a multi-head attention layer (21 heads), a dropout layer, layer normalization with residual connection, and a feedforward network. This feedforward network consists 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.

A.5. Drug Optimization illustration on COVID benchmark

This is another example illustrating the effectiveness of SCAFFOLDGPT in enhancing the original molecule on the COVID benchmark. The results in Table 4 show that the drugs generated by SCAFFOLDGPT outperform the original drugs across all desired properties. Even though the original scaffold is altered and not present in the generated molecules, the similarity still demonstrates a decent level.

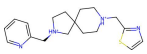
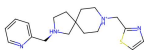
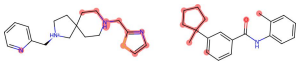
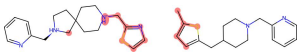
	Original	Scaffold	improved 1	improved 2
Molecule				
SMILE String	c1ccc(C[N@H+]2CCC3(CC[NH+](Cc4nccs4)CC3)C2)nc1		CC1(c2cccc(C(=O)Nc3ccccc3C)c2)CCCC1	Cc1ccc(CC2CCN(Cc3ccccn3)CC2)s1
Scaffold	c1ccc(C[N@H+]2CCC3(CC[NH+](Cc4nccs4)CC3)C2)nc1		-	-
Docking (↓)	-9.748		-10.184 ✓	-10.187 ✓
Druglikeness (↑)	0.839		0.840 ✓	0.847 ✓
Synthesizability (↓)	5.631		1.983 ✓	2.199 ✓
Solubility (↑)	0.192		5.079 ✓	3.906 ✓
Similarity	-		0.335	0.563
Avg Norm Reward (↑)	0.398		0.688 ✓	0.694 ✓

Table 4: One molecule example from 3CLPro dataset, where scaffold and original are same. In this case the model tries to modify the scaffold, and the generated molecules does not contain scaffold. ✓ indicates improved property.

A.6. Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of the supercomputer where each node contains CPUs (64 cores) and 4 A100 GPU nodes (Facility). The training time for each model was approximately 3 hours.

We conducted other experiments on a cluster that includes CPU nodes (approximately 280 cores) and GPU nodes (approximately 110 Nvidia GPUs, ranging from Titan X to A6000, set up mostly in 4- and 8-GPU configurations).

The pretraining process utilizes 8 GPUs, while APO and generation employs a single GPU. Both processes use either V100 or A100 GPUs. Based on the computing infrastructure, we obtained the wall-time comparison in Table 5 as follows.

Methods	Total Run Time
Pretraining	24h
APO	27h
TOP-N (One Generation)	17-20s

Table 5: Wall-time comparison between different methods.

A.7. Hyperparameters and architectures

Table 6 and 7 provides a list of hyperparameter settings we used for our experiments.

For APO finetuning and experimentation, 1280 molecules were selected from each of the RTCB and 3CLPro datasets, with docking scores ranging from -14 to -6. This range is based on (Liu et al., 2024b).

Moreover, when computing the average normalized reward for the original molecule, in the absence of similarity considerations, we use weights of 0.25 for docking, drug-likeness, synthesizability, and solubility, respectively.

Moreover, when the generated SMILES is invalid, indicating that the reward R_c cannot be calculated, we have two options: either directly subtract the reward of the original SMILES (i.e., $-R_c(X)$), or consider the advantage preference as zero instead.

Parameter	Value
Pretraining	
Learning rate	$5 \times e^{-5}$
Batch size	24
Optimizer	Adam
# of Epochs for Training First Phase	10
# of Epochs for Training Second Phase	10
Model # of Params	$124M$
Generation	
N (Top-N)	1
K (Number of possible next token)	16
TopK	$[10, 15, 20]$
TopP	$[0.85, 0.9, 0.95]$

Table 6: Hyperparameters for pretraining and generation.

Parameter	Value
Shared	
# of Molecules Optimized	1280
Learning rate	1×10^{-4}
Optimizer	Adam
# of Epochs for Training	100
Batch size	64
Best-of-N	$[4, 6, 8]$
TopK	$[10, 15, 20]$
TopP	$[0.85, 0.9, 0.95]$
APO Objective Weight	
Docking Score	0.2
Druglikeliness	0.2
Synthesizability	0.2
Solubility	0.2
Tamimoto Similarity	0.2
APO Other	
Fingerprint Size	1024
Normalize Min/Max	$[-10, 10]$
Advantage preference with invalid generated SMILES	
3CLPro	$[0, -R_c(X)]$
RTCB	$[0, -R_c(X)]$

Table 7: Hyperparameters for APO.