THE UNIVERSITY OF CHICAGO

# LEVERAGING TRANSFORMER MODELS FOR ACCELERATED DRUG DISCOVERY

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

SONGHAO JIANG

CHICAGO, ILLINOIS
AUGUST 2024

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Rick Stevens, for his unwavering support and patience throughout my PhD journey. His guidance and encouragement in this interdisciplinary project have been a constant source of inspiration, enabling me to pursue my research with confidence. This thesis would not have been possible without his invaluable support.

I am also profoundly grateful to my thesis committee members, Dr. Ian Foster and Dr. Fangfang Xia, for their insightful feedback and guidance. A special thank you to Dr. Fangfang Xia for always making time to meet with me, even when circumstances made it challenging. Your wise words, valuable suggestions, and patience with my rants have been a pillar of support throughout this journey. Thank you, Dr. Ian Foster, for your valuable suggestions on paper revisions. Your input has been instrumental in shaping the quality of my work.

I would also like to extend my heartfelt thanks to Xuefeng Liu. His constructive suggestions and support have helped me overcome many obstacles. Without his valuable ideas and assistance, the proposed projects would not have been completed as effectively and efficiently.

# ABSTRACT

In the realm of AI-accelerated drug discovery, particularly in de novo drug design, significant challenges include unpredictable drug responses in clinical trials, biases in predictive models, and the opaque nature of AI methodologies that complicate the understanding of a drug's mechanism of action. These issues have limited the progression of AI-discovered drugs into clinical trials and regulatory approval. Concurrently, the development of me-too drugs, which involve modifications of existing drugs within the same therapeutic class, presents a less risky and potentially more effective avenue. However, the potential of AI to enhance their development remains largely underexplored.

This dissertation aims to transform the development of me-too drugs through the application of AI, with a focus on transformer and large language models (LLMs). It introduces innovative frameworks that utilize the representation learning and generative capabilities of transformer models to refine and expedite the me-too drug development process. These methodologies, referred to as "drug optimization", seek to further accelerate the production of effective me-too drugs.

This work makes four significant contributions to the field: (1) It proposes two fusion methods that integrate transformer models with graph neural networks, enhancing the precision of binding affinity predictions. (2) It assembles a comprehensive dataset of 10 million binding affinity values across a diverse array of proteins and drugs, providing an invaluable resource for model training and validation. (3) It proposes two generative models for drug optimization, fine-tuned through reinforcement learning, with the goal of automating and expediting the creation of effective me-too drugs. (4) It introduces an innovative bidirectional GPT model for molecular textual sequences (SMILES), enabling precise generative mask infilling for targeted drug optimization. And by conducting comprehensive evaluations on real world viral and cancer target proteins, we demonstrate that the proposed drug optimization frameworks can consistently enhance existing molecules/drugs.

# CHAPTER 1

# INTRODUCTION

The journey from a novel drug concept to market entry is widely acknowledged as a lengthy, risky, and costly endeavor. Typically, this process spans approximately 14 years [71] and incurs costs ranging from 0.8 to 1.0 billion USD [70]. Despite significant investments in new drug development over recent decades, the output has not increased proportionally, primarily due to the low efficiency and high failure rates inherent in drug discovery [96]. Consequently, the pharmaceutical industry has explored various strategies to expedite the drug discovery process, aiming to reduce both the time and financial burdens, as well as the risk of failure.

In this context, computer-aided de novo drug design stands out as a particularly promising approach to achieve these objectives. It employs computational methods to generate novel molecular structures with desirable pharmacological and physicochemical properties from scratch. While many machine learning-based algorithms and biotech firms focus on this approach, it faces significant challenges. The most critical issue is the uncertain drug responses in humans, particularly regarding efficacy and safety. Although researchers have developed machine learning surrogate models for drug response prediction, these models often rely on in vitro and animal data, which may not accurately predict human responses [65]. Additionally, the black-box nature of AI complicates the understanding of a drug's mechanism of action, further challenging assessments of safety and efficacy. As a result, no AI-discovered drug has received regulatory approval despite some entering clinical trials[93]. Given these challenges, our focus shifts to an alternative approach: improving existing drugs, or me-too drug development [67].

Parallel to the exploration of entirely novel compounds, the pharmaceutical industry has also sought to develop "me-too" drugs. More specifically, me-too drugs are developed by modifying existing prototype drugs, belonging to the same therapeutic class and serving

| Reason | Examples |
|---|---|
| To improve specificity at the target, thus reducing the risks of off-target adverse reactions and drug–drug interactions | Atypical antipsychotic vs Typical antipsychotic |
| To reduce the risks of off-target adverse reactions and drug–drug interactions, without altering on-target specificity | Ranitidine vs Cimetidine |
| To develop drugs with similar structures but new targets | Acecainide vs Procainamide |
| To increase the chance of benefit, perhaps in a subset of patients | Ampicillin vs Benzylpenicillin |
| Incremental innovation | Beta-blockers |

Table 1.1: Selected therapeutic reasons for developing, marketing and using me-too drugs and their corresponding examples [4]

identical purposes. Building on the established profile of the prototype, me-too drugs are less risky and can offer incremental improvements in safety or efficacy rather than the development of entirely innovative drugs, which carries a higher risk of failure [113, 4]. Me-too drugs have gained considerable popularity within the pharmaceutical industry, with numerous companies actively developing these variations. As of 2020, the World Health Organization's list of essential medicines consists of over 370 drugs, of which 60% can be considered me-too drugs [4]. The rationale and benefits of developing me-too drugs, including improved target specificity and reduced adverse reactions, are summarized in Table 1.1, highlighting their significance in therapeutic advancement. However, despite its increasing importance, using computational methods for effective and efficient me-too drugs development is still much less studies comparing to de novo drug design. However, despite their popularity, AI and ML opportunities in me-too drug development are relatively less explored. In this thesis, we aim to utilize transformers, a rising star in NLP, to develop me-too drugs, thereby accelerating drug discovery.

Machine learning in molecular research can be roughly divided into two domains: representation learning, which involves learning representations from molecules for downstream tasks like property prediction, and generative models, which predict new structures with

desired properties. There are two popular ways to represent molecules: at the sequence level, using SMILES, and at the structure level, using graphs. Combining these two domains and two representations, we get four parts, as shown in Fig. 1.1. Numerous researches and models have been developed to address problems within these four parts.

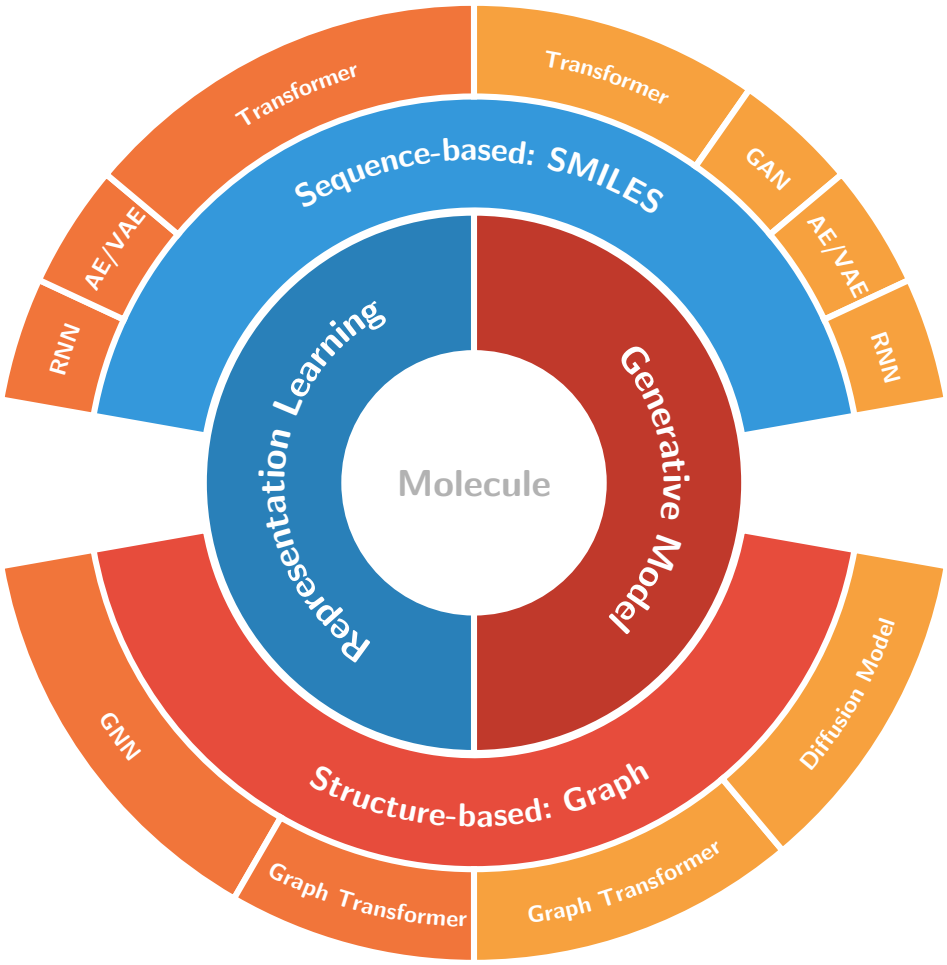



Figure 1.1: Machine learning in molecular research

At the sequence level, prior to transformers, researchers used models like traditional RNNs, LSTMs, autoencoders, VAEs, and GANs. However, transformers have demonstrated state-of-the-art performance in various NLP tasks and shown their effectiveness in drug discovery [35, 36] and protein design [26, 25], generating high-quality molecules and proteins. Recently, researchers have focused on applying graphs to transformers [125, 62] and diffusion

models [108, 41]. Despite some advances, these models face issues. For example, graph transformers struggle with low synthesizability due to uncommon rings when handling long-distance dependencies [62]. On the other hand, molecules generated by transformer models have shown decent synthesizability [35, 36]. Moreover, diffusion models on graphs have interpretability and conditional generation challenges [108]. As previously mentioned, models with decent interpretability can help us understand a drug's mechanism of action. Although the interpretability of transformers is still limited, researchers can use attention mechanisms and neurons for local and global explanations [135]. Moreover, in drug discovery, it is crucial to control the generated outputs for specific features, and transformers can achieve this by designing training corpus [35, 36] or reinforcement learning [40]. Given these drawbacks, we will focus on using transformers with sequences in generative models.

Similarly, transformers excel in representation learning for various downstream tasks compared to other sequence models like RNNs or autoencoders. Notably, transformers trained on sequences have successfully predicted 3D structures [60], a significant improvement over other sequence models. At the structure level, various GNNs [115, 133] and graph transformers [125] have shown competitive performance. However, sequence-level transformer models have produced comparable results to GNNs and graph transformers [134, 31]. Considering these points and the advantages of transformers in generative models, this thesis will focus on using transformers. This brings us to our research question.

## 1.1 Research Question and Formalization

*Can transformer models be utilized to improve desired properties of molecules, while maintaining structural similarity to their prototypes?*

Given a molecule $M_0$ with representation $r$ and a list of desired properties $\{p_1, p_2, \cdots, p_n\}$, where $p_i(M)$ quantifies molecule $M$, we aim to construct new molecule $\tilde{M}$ so that $R(\tilde{M}) - R(M_0) > 0$, where $R(M) = \sum_{i=1}^{n} a_i \cdot p_i(M) + \beta \cdot \text{Similarity}(M, M_0)$ and $\sum_{i=1}^{n} a_i + \beta = 1$

## 1.2 Contributions

To address previously mentioned challenges and the research question, the thesis proposes five novel contributions:

- We design two fusion methods that integrate the strengths of transformer models and graph neural networks, aiming to further improve the accuracy of binding affinity predictions.
- We simulate a new comprehensive dataset encompassing 10 million binding affinity values across 10,000 proteins and 1,000 drugs, providing a rich resource for model training and validation.
- We propose two drug optimization generative models, fine-tuned with reinforcement learning. These models aims to automate and expedite the development of effective me-too drugs.
- We present a bidirectional GPT on molecule textual sequences SMILES, a hybrid of causal and masked language models, and Seq2Seq that allows for bidirectional context control during generative mask infilling for guided drug optimization.
- We demonstrate that the proposed drug optimization models enhances existing molecules and drugs across all desired objectives, leading to improved drug candidate, by conducting comprehensive experiments on real world viral and cancer target proteins,

Through these contributions, this thesis aims to open up new possibilities for enhancing drug discovery and inspires future investigations into addressing challenges within the realm of me-too drugs development, thereby offering new pathways to accelerate the development of safer and more effective therapies.

## 1.3 Thesis Outline

Following this introduction, Chapter 2 will delve into the existing literature, with a focus on de novo drug design and binding affinity prediction, an important metric in computational drug discovery. Additionally, this chapter will explore the integration possibilities of transformer-based models in these domains.

Chapter 3 can be viewed as a forward design, which is a prediction of docking scores for a given molecule and target. This project introduces fusion methods, combining transformer and graph representation learning to improve binding affinity or docking score prediction. It aims to demonstrate that transformer-based models can enhance docking score prediction, which will be used as a surrogate model in the DrugImprover framework.

Chapter 4 is a transitional project, or the first naive version of DrugImprover, which combines transformer-predicted docking scores and other desired objectives. In addition, we use LSTM as the generator and apply reinforcement learning for optimization.

Following this are two DrugImprover frameworks using transformer-based large language models. In general, the DrugImproverLLM, as introduced in chapter 5, is a casual language model, which improves DrugImprover by replacing LSTM with GPT.

In addition, DrugImproverCM, which will be detailed in chapter 6 further improves DrugImproverLLM by combining benefits of casual and masked language model, and Seq2Seq, which enables user-directed sample generation via token conditioning.

# CHAPTER 2

# BACKGROUND AND RELATED WORKS

The journey of drug development is both time-consuming and costly, typically spanning 14 years and incurring expenses of up to $1 billion. This process is further complicated by low efficiency and high failure rates, prompting the pharmaceutical industry to explore various strategies to expedite the drug discovery process. Among these, computer-aided de novo drug design and the development of me-too drugs stand out. De novo drug design focuses on creating novel molecular entities from scratch with desired properties, while me-too drugs involve minor modifications to existing drugs to improve efficacy or reduce side effects. Despite the considerable popularity of me-too drugs, there is a noticeable gap in using computational methods compared for effective and efficient me-too drugs developemt compared to de novo drug design, highlighting an area ripe for further exploration.

The advent of transformer models, known for their groundbreaking achievements in natural language processing, presents a novel opportunity in this domain. Their advanced capabilities in representation learning and generative tasks position transformer models as a potentially transformative force in drug design. This approach, which we term drug optimization, aims to automate and expedite the development of me-too drugs, addressing the urgent need for rapid innovation in pharmaceutical research amidst emerging health threats.

This chapter aims to provide a comprehensive review of the literature on deep learning-based approaches in de novo drug design and the optimization of drug/drug-like molecules. It will also delve into binding affinity prediction, a crucial metric in computational drug discovery, to integrate this aspect into our drug optimization framework. Furthermore, we will explore the capabilities of transformers in representation learning and generation, underscoring their promising applicability in the realm of drug optimization, particularly for enhancing the development process of me-too drugs.

## 2.1 De Novo Drug Design

De novo design represents a revolutionary approach in the realm of drug discovery, offering a cost-effective and efficient pathway to synthesize novel molecules with desired properties from scratch, without the need for existing molecular frameworks. Central to this approach are two primary strategies: generative model-based methods and reinforcement learning (RL). This section delves into each of these strategies, examining their methodologies, benefits, and challenges within the context of de novo drug design.

### *2.1.1 Data-Driven Generative Models*

Generative models are at the forefront of de novo drug design, mapping points in a high-dimensional latent space to molecules. These models employ data-driven techniques to sample from a learned distribution of chemical structures, thereby bypassing the need for exhaustive screening of extensive databases. This capability leverages the model's ability to understand and replicate the distribution of chemical structures in a dataset, facilitating the generation of new, potentially effective molecules.

Data-driven generative models are roughly divided into four categories, including the models based on recurrent neural network (RNN) [33], autoencoder (AE) [58, 98], generative adversarial network (GAN) [48] and transformers [57]. In terms of representation, research in de novo drug design has predominantly utilized SMILES (Simplified Molecular-Input Line-Entry System) strings [48, 33, 58, 57] or graph-based representations [98].

Despite their successes, data-driven generative models face significant challenges. A significant hurdle is the requirement for pre-training on specific datasets. While pre-training enables the generation of molecules similar to those in the training set, it inherently limits exploration capabilities due to biases present in the training data [137]. Moreover, controlling the properties of generated molecules remains a complex issue. Most generative models produce molecules with random properties, and efforts to guide the generation process

towards desired properties have been met with mixed success. For instance, fine-tuning a pre-trained LSTM model with a subset of molecules possessing desired properties [33] can lead to overfitting, thereby constraining exploration to the biases of the selected subset. Similarly, attempts to control multiple molecular properties simultaneously by imposing constraints on the latent space [58] often face difficulties due to the high-dimensional and non-convex nature of the objective functions defined on this space [137].

### *2.1.2 Reinforcement Learning Models*

Transitioning from the inherent limitations of generative models, reinforcement learning (RL) offers a complementary approach by using a reward system to iteratively guide the generation of novel molecular structures towards desired properties. Through trial and error, RL algorithms learn to make decisions—such as adding or modifying parts of a molecule—that are evaluated against predefined criteria like solubility and synthetic accessibility. Successful outcomes reinforce the algorithm's decision-making process, steering it towards proposals that are more likely to exhibit the targeted characteristics.

More specifically, most RL algorithms are implemented in conjunction with data-driven generative models. For example, ReLeaSE employs RL techiniques on a stack-RNN generative model that has been trained using SMILES [80]. Similarly, ORGAN integrates RL with Monte Carlo sampling techniques on an LSTM-based GAN, also trained on SMILES [32]. Moreover, MolGAN represents an advancement of ORGAN by replacing the LSTM-based GAN with a graph-based GAN, which is pre-trained on molecular graphs [18].

However, these studies often focus on discovering new drugs, frequently overlooking molecular structure constraints during policy improvement. This oversight can result in low validity, uniqueness or complexity of generated molecules due to drastic changes in structure or functional groups. In contrast, our proposed drug optimization, concentrates on optimizing existing drugs while preserving their beneficial properties, rather than creating entirely new

ones from scratch.

In addition, while these RL-based models have shown success in generating molecules with desired properties, such as solubility and druglikeness, they often focus on a restricted set of drug properties and notably overlook docking score, a crucial metric for assessing structural compatibility with a target. And we aim to utilize the docking score/binding affinity as an additional objective in our proposed drug optimization.

## 2.2 Drug/Drug-like Molecule Optimization

In the realm of optimizing drug or drug-like molecules, both data-driven generative models and reinforcement learning (RL) methods have been utilized. Initially, researchers have explored training transformers to emulate the chemist's intuition, particularly through the lens of matched molecular pairs (MMPs), aiming to optimize promising molecules [35]. Specifically, the objective has been to generate molecules that not only possess the desired properties but also maintain structural resemblance to the starting molecule. Building on this, further research has broadened the scope, developing a methodology that allows for more extensive structural modifications beyond MMPs [36]. This involves training transformers on varied datasets, each comprising molecular pairs indicative of different transformation types. However, similar to challenges faced in de novo drug design, such pretraining often restricts exploratory breadth due to biases inherent in the training data [137].

Moreover, recent advancements have incorporated reinforcement learning to tackle molecule optimization. Zhou et al.'s work exemplifies this by formulating molecule modification as a Markov decision process (MDP) on molecular graph representation. This approach involves manipulating bonds and atoms in original molecules through MDP, employing multi-objective RL to choose actions at each state aimed at maximizing future rewards. Despite these advancements, a common limitation is the narrow focus on a select set of drug properties, often neglecting crucial metrics like the docking score, which is vital for assessing structural

compatibility with biological targets.

## 2.3 Binding Affinity Prediction

In the rapidly advancing fields of de novo drug design and drug/drug-like molecule optimization, the importance of docking scores is often overlooked. These scores are essential for assessing the structural alignment between a molecular structure and a reference structure, playing a critical role in predicting the potential efficacy of drug candidates. However, the computational intensity required for accurately calculating docking scores through virtual screening tools, such as OEDocking [49] and Autodock Vina [107], presents a significant challenge. This challenge limits their widespread use and underscores the pressing need for the development of machine learning (ML) and deep learning (DL) based surrogate models. Such models could efficiently incorporate docking scores into our proposed drug optimization framework, overcoming existing computational barriers.

Traditionally, docking scores have also been employed to estimate protein–ligand binding affinity [130], a critical measure of the interaction strength between a protein and a ligand, typically determined through experimental methods. The focus of ML and DL research in this area has predominantly been on predicting this binding affinity. This emphasis reflects a broader scientific interest in accurately capturing the efficacy of real-world drug-ligand interactions, which is vital for the successful development of new therapeutic agents.

This section of the thesis aims to review existing DL-based methods for binding affinity prediction, making a distinction between interaction-based and interaction-free approaches. Furthermore, it will discuss how advancements in protein representation learning could enhance the accuracy of binding affinity predictions. Through this review, we seek to underscore the potential of DL not only in improving binding affinity prediction but also in its applicability to docking score prediction. The latter is particularly relevant given that the input format for both applications is fundamentally identical. This exploration intends

to bridge the gap between traditional docking score calculation methods and the innovative application of DL techniques, highlighting their potential to revolutionize the field of drug design and optimization.

### *2.3.1 Interaction-based vs. Interaction-free Methods*

In the domain of binding affinity prediction, the scientific community has primarily pursued two methodologies: interaction-based and interaction-free models. These methodologies are distinguished by their reliance on physical interactions to inform predictions.

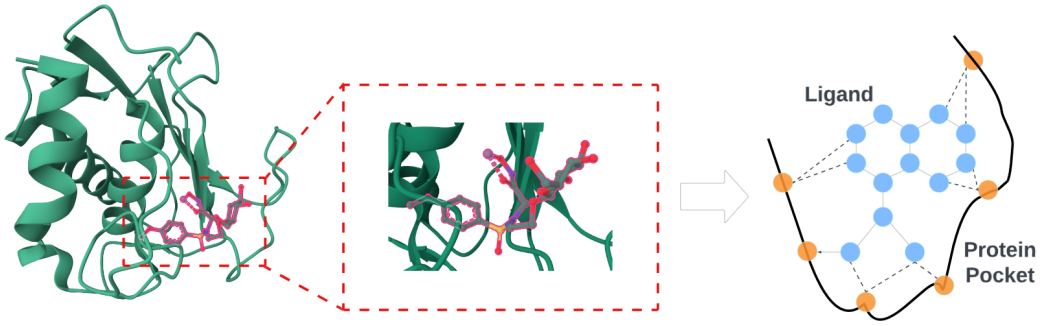



Figure 2.1: Interaction-based models make predictions based on the 3D structures of complexes and physical interactions of proteins and ligands. Only atoms surrounding the interaction/binding pocket are used to build graphs for prediction.

Interaction-based models operate on the premise that a detailed understanding of the

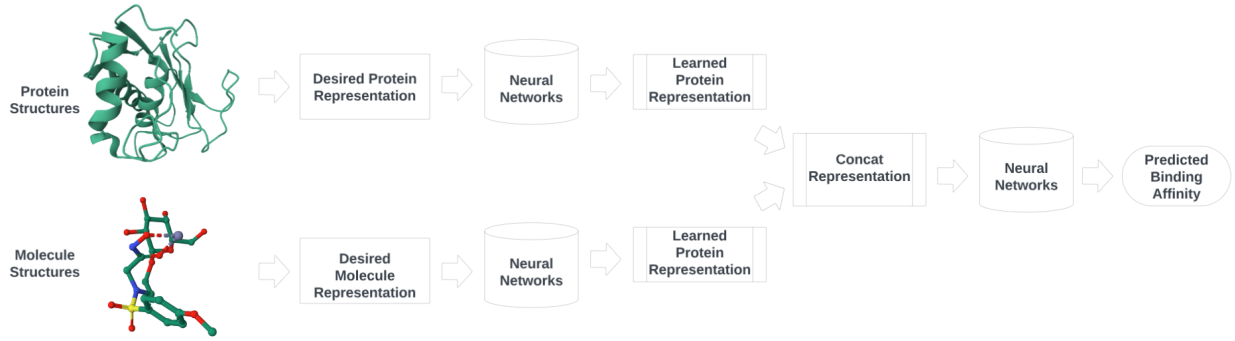



Figure 2.2: Interaction-free methods implicitly assume that ML models can predict molecular docking from data that do not reveal physical protein-ligand interactions.

3D structure of atomic-level interactions between protein-ligand complex can significantly enhance prediction accuracy. A prevalent method within this category involves representing these structures through graphs with atomic interaction information [59, 54, 124]. As illustrated in Figure 2.1, these models typically focus on the atoms within the binding pocket, converting them into graphs for GNN processing. More specifically, Yang et al. introduced a heterogeneous interaction layer that integrates both covalent and noncovalent interactions within the message passing phase, thereby enhancing node representation learning. This layer adheres to fundamental biological principles, such as invariance to the translation and rotation of complexes. Similarly, Li et al. developed a structure-aware interactive graph neural network (SIGN), which employs polar-inspired graph attention layers (PGAL) and pairwise interactive pooling (PiPool) to maintain distance and angle information among atoms and to capture global interactions, respectively. Despite their efficiency, these models often require extensive prior knowledge about the binding site and interaction dynamics, which may limit their applicability in scenarios where such information is not readily available [115]. Furthermore, their generalization and inference efficiency are constrained, as they necessitate experimentally or simulation-determined structures for protein-ligand pairs during inference.

Contrasting with the detailed structural reliance of interaction-based methods, interaction-free models infer molecular docking and binding affinity from data that abstract away explicit physical interactions. Typically, ligands are represented using the simplified molecular-input line-entry system (SMILES) or as 2D graphs, and proteins are depicted through sequences or 2D graphs, omitting atomic interactions for simplicity [77, 73, 115]. As depicted in Figure 2, this approach involves learning separate representations for proteins and molecules, which are then concatenated into a merged representation for affinity prediction. This approach, while avoiding the computational burden of modeling every potential interaction site on the protein surface, better accommodates the uncertainty inherent in exploring unknown interactions.

However, the abstraction from physical interactions necessitates sophisticated computational techniques to infer binding affinities accurately, posing its own set of challenges.

Given the distinct advantages and limitations of both methodologies, our research aims to propose a novel interaction-free binding affinity prediction model. This model seeks to leverage the strengths of interaction-free approaches while addressing their limitations, particularly in scenarios where optimized drugs lack experimentally or simulation-determined docking poses.

### *2.3.2 Protein Representation Learning in Binding Affinity Prediction*

As highlighted in the previous section, the pursuit of interaction-free methods for binding affinity prediction has underscored the necessity for advanced computational techniques that do not rely on physical interaction data. As indicated by figure 2.2, the interaction-free methods will require separate representations and neural networks for protein and molecules independently. Proteins, with their complex structures and dynamic functions, offer a richer, more complex canvas for computational exploration compared to small molecules. This section explores the evolution of protein representation methods, moving from basic sequence-based methods to more complex multimodal approaches, and outlines how these advancements have enhanced binding affinity predictions.

Initially, proteins were predominantly represented as sequences of amino acids, with 1D convolutional neural networks (CNNs) employed to learn these representations [77, 73]. In this approach, each amino acid in the protein sequence is encoded as an integer (e.g., 'C': 1, 'H': 2, 'N': 3, etc.), which is then often transformed into a one-hot encoded vector. Despite achieving a reasonable accuracy in predicting binding affinities, this method largely ignores the crucial 3D structural information of proteins, which plays a significant role in indentifying binding sites and pockets.

To address the limitations of sequence-based methods, recent efforts have focused on

incorporating 3D structural information into protein representations. Researchers have experimented with representing proteins as 3D voxels, where the protein's structure is mapped into a three-dimensional grid of fixed size, with each voxel encoding atomic features [87, 46]. This representation is then processed using 3D CNNs to learn the protein's features. Although voxel-based methods marked a significant improvement over sequence-based approaches by incorporating spatial information, they faced challenges related to sparsity of resulted voxel girds, which often resulted in many empty voxels, hence increasing computational cost and reducing performance at lower resolution [123].

The advent of graph representation learning offered a solution to the drawbacks of voxel-based methods. By representing proteins as graphs, where nodes correspond to amino acids or atoms and edges represent spatial or sequential relationships, researchers have managed to not only enhance performance but also address the computational inefficiencies of voxelization [38, 133]. Graph-based models have demonstrated promising results in binding affinity prediction, achieving SOTA outcomes by capturing the complex structures and interactions of proteins more naturally and efficiently [54, 124, 115].

Despite the successes of graph-based representations, it is increasingly recognized that no single modality fully capture the complicated nature of protein structure and function. Consequently, this has led to increasing interest in multimodal approaches that integrate various types of information to provide a more comprehensive view of proteins. For instance, GearNet [133] combines graph representations with sequential data from protein sequences, improving the model's understanding of protein structures. Furthermore, integrating embeddings from protein language models, such as LSTM, TAPE, and ESM [37, 86, 60], with geometric learning frameworks [44, 117, 134], has also shown to improve predictions by capturing more details about protein structures and functions.

Despite these advancements, the field is still in the early stages of exploring how to best use multimodal approaches for predicting binding affinities. Most studies have simply combined

sequence and graph representations via concatenation without fully taking advantage of the strengths of each method. This highlights a gap in current methods and points to the need for more sophisticated ways to bring together different types of information. Specifically, we aim to explore how transformer-based models for sequence representation could improve predictions over unimodal graph neural networks. This could be a key part of our proposed drug optimization, which enables docking score as part of the framework.

## 2.4 Transformer-based Language Model

The advent of transformer models and large language models (LLMs), known for their exceptional performance in natural language processing (NLP), opens new avenues for innovation in drug optimization. These models, with their advanced capabilities in representation learning and generative tasks, hold the potential to significantly revolutionize the field of drug design. In this section, we will delve into LLMs, covering their training objectives and generation/decoding strategies. Furthermore, we will explore the application of LLMs in bioinformatics, highlighting their potential to transform this critical area.

### *2.4.1 Training Objectives*

This subsection introduces popular training objectives in LLMs, including Masked Language Models (MLM), Causal Language Models (CLM), and the Causal Masked Model (CMM), providing a foundation for understanding how these models learn and evolve.

**Masked Language Models (MLM):** MLM is a pre-training technique used in NLP that enables models to better understand the context and meaning of words within sentences. By randomly masking a certain percentage of the input tokens, the model is tasked with predicting the original identity of these masked tokens, leveraging the context provided by the other, non-masked tokens. This approach, popularized by BERT (Bidirectional Encoder Representations from Transformers) [20], allows the model to learn from bidirectional contexts,

thereby gaining a deeper understanding of language syntax and semantics. BERT has shown strong performance gains using self-supervised training that masks individual words or subword units. However, many NLP tasks involve reasoning about relationships between two or more spans of text. To address more complex language structures, Span-MLM and Span-Corruption-MLM have been developed.

Span-MLM, introduced by SpanBERT [47], extends the basic MLM concept by masking contiguous spans of tokens, compelling the model to predict entire phrases or segments of text. This method enhances the model's grasp of syntactic and semantic structures, proving beneficial for tasks requiring an understanding of complex language constructs. Span-Corruption-MLM, implemented in the T5 model [84], frames the prediction task as a text-to-text problem, where the model generates an output sequence to correct or fill in the corrupted parts of the input. This flexible approach allows for comprehensive training across a wide range of NLP tasks by framing them as text-to-text problems, showcasing the model's versatility and its ability to generate human-like text across different contexts. We provide examples of the inputs and targets for these three methods in Table 2.1

| Objective | Inputs | Targets |
|---|---|---|
| Bert-style [20] | Today <M> <M> beautiful <M> to go <M> a walk in the park | *(original text)* |
| SpanBert-stype [47] | Today <M> <M> <M> <M> to go for a walk in the park | *(original text)* |
| T5-style [84] | Today <X> beautiful day to <Y> for a walk in the park | <X> is a <Y> go <Z> |

Table 2.1: Examples of inputs and targets produced by some MLM objectives we consider using the input text "Today is a beautiful day to go for a walk in the park". We write *(original text)* as a target to denote that the model is tasked with reconstructing the entire input text. The symbol <M> represents a common mask token, while <X>, <Y>, and <Z> are used for sentinel tokens, each with a distinct token ID.

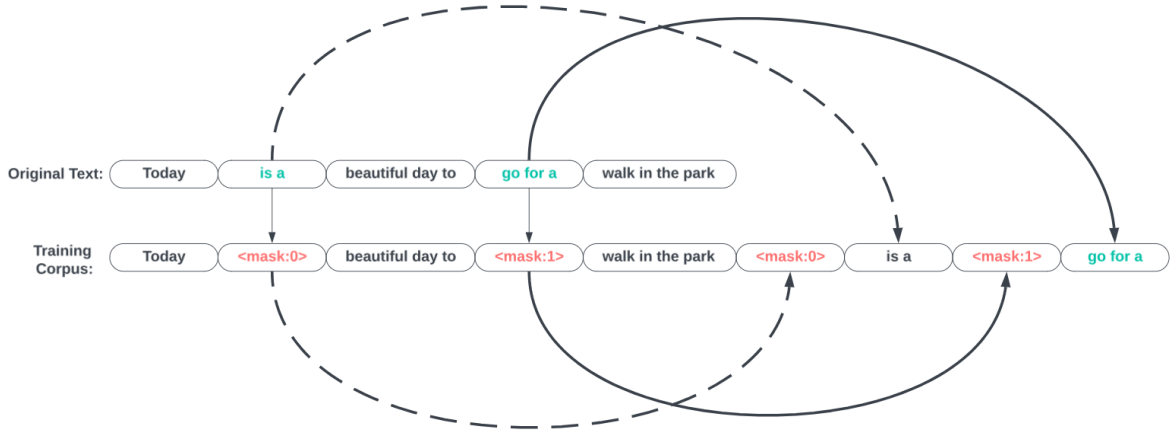



Figure 2.3: A visual representation of causal masked objective with two masks. The training corpus will then be trained using the same manner as CLM.

**Causal Language Models (CLM):** Causal Language Modeling (CLM), also known as autoregressive language modeling, focuses on predicting the next word in a sequence given the words that precede it. Unlike MLM, which predicts masked tokens anywhere in a sentence using bidirectional context, CLM operates in a unidirectional manner. This approach mimics the way humans naturally generate language, making it particularly suitable for tasks involving text generation, such as story generation, machine translation, and auto-completion. Models like GPT and GPT2 [82, 83] exemplify CLM's ability to generate coherent and contextually relevant text by learning the underlying structure and patterns of the text

**Causal Masked Model (CMM):** In an effort to get most of the best of both CLM and MLM, Aghajanyan et al. introduce a novel objective, causal masked objective that combines the benefit of per-token generation with optional bi-directionality specifically tailored to prompting. More specifically, causally masked model generates tokens left to right, just like CLM, but also mask out a small number of long token spans, which are then generated at the end of the string instead of their original positions. This provides a new hybrid of causal and masked language models, enabling full generative modeling with bidirectional context. Figure 2.3 shows the an example of such process.

### *2.4.2 Generation Strategies*

Besides the improved transformer architecture trained with different objectives, better decoding methods have also played an important role. These strategies balance between randomness and determinism to produce coherent and diverse outputs. Here's an overview of some commonly used methods:

**Greedy search:** Greedy search is the simplest form of text generation where, at each step, the model selects the word with the highest probability as its next output

$$w_t = argmax_w P(w|w_{1:t\_1}) \tag{2.1}$$

This approach ensures that the model always chooses what it considers the most likely next word. However, this method can lead to repetitive and less diverse text because it always opts for the safest, most probable option without exploring potentially more interesting alternatives. The major drawback of greedy search though is that it misses high probability words hidden behind a low probability word, which is alleviated by beam search.

**Beam search:** Beam search [101] is a more sophisticated approach that expands on greedy search by considering multiple possible sequences at each step. It keeps track of a fixed number of the best (highest probability) sequences seen so far, known as the beam width. At each step, it expands each sequence in the beam by one word, calculates the probabilities of all possible next words, and keeps only the top-scoring sequences. This method balances between breadth and depth, aiming to find a more optimal sequence than greedy search. However, it can still suffer from a lack of diversity and may miss high-quality sequences outside of the narrow beam.

**Multinomial sampling:** Unlike greedy and beam search, multinomial (or categorical) sampling introduces randomness into the text generation process by selecting the next word based on its probability distribution. This means that higher probability words are more

likely to be chosen, but lower probability words still have a chance. This method can generate more diverse and interesting text but may sometimes produce less coherent or relevant output due to its stochastic nature.

**Top-K Sampling:** Top-K sampling [24] introduces a balance between randomness and relevance by only considering the top K most likely next words at each step and sampling from this subset according to their probability distribution:

$$w_t \sim P(w|w_{1:t\_1}) \tag{2.2}$$

This method reduces the chance of selecting very unlikely words, thus maintaining coherence while still allowing for some variability and creativity in the generated text. The choice of K is crucial; too small a value might constrain creativity, while too large a value might include too many low-probability, irrelevant options.

**Top-P (Nucleus) Sampling:** Top-P sampling [39], also known as nucleus sampling, takes a dynamic approach by choosing a variable number of words to consider at each step, based on a cumulative probability threshold $p$. It selects the smallest set of $V$ words whose cumulative probability exceeds the threshold $p$ and samples from this set.

$$\sum_{w \in V} P(w|w_{1:t-1}) \geq p \tag{2.3}$$

where $V$ is the smallest possible set of tokens.

This method aims to balance diversity and coherence by dynamically adjusting the range of considered words based on their relevance. Top-P sampling can effectively prevent the generation of implausible text while allowing for creative and contextually appropriate outputs.

### *2.4.3 Generative Models in Drug Discovery*

The advent of generative models, particularly transformer-based language models, has marked a significant milestone in the field of drug discovery. These models have been adeptly applied to mimic and augment the chemist's intuition, leading to notable advancements in the identification and optimization of therapeutic molecules.

Initially, researchers have explored training transformers to emulate the chemist's intuition, particularly through the lens of matched molecular pairs (MMPs), aiming to optimize promising molecules [35]. Specifically, the objective has been to generate molecules that not only possess the desired properties but also maintain structural resemblance to the starting molecule. Building on this, further research has broadened the scope, developing a methodology that allows for more extensive structural modifications beyond MMPs [36]. This involves training transformers on varied datasets, each comprising molecular pairs indicative of different transformation types.

In a parallel development, DrugGPT [57] introduced a ligand design strategy leveraging the autoregressive capabilities of the GPT model. This strategy focuses on exploring the chemical space extensively to discover ligands that can specifically bind to target proteins. DrugGPT's approach involves training the model on a vast dataset of protein-ligand binding information, aiming to generate novel molecules capable of interacting with specific proteins effectively.

Moreover, ChemGPT [27] has explored the neural-scaling behavior in large chemical models by adjusting the model and dataset sizes across several orders of magnitude. This investigation into models with over one billion parameters, trained on datasets comprising up to ten million data points, sheds light on the scalability and efficiency of transformer-based models in processing and generating chemical data.

These pioneering efforts underscore the transformative potential of transformer-based language models in drug discovery. However, these data-driven pretraining models often

restricts exploratory breadth due to biases inherent in the training data [137]. In addition, generation of molecules with desired properties are often overlook.

Despite these advancements, the reliance on data-driven pretraining models introduces inherent biases from the training data, which can limit the exploratory breadth [137] and overlook the generation of molecules with desired properties. These limitations underscore the need for continuous refinement of these models to mitigate biases and enhance their predictive and generative capabilities.

### *2.4.4 Representation Learning in Bioinformatics*

Transformers have revolutionized the field of natural language processing (NLP) with their generative capabilities and have extended their influence to representation learning, a crucial aspect of machine learning. Representation learning transforms raw data into a format that can be effectively utilized for prediction or classification tasks. Transformers, especially those trained with the Masked Language Model (MLM) objective, have demonstrated exceptional proficiency in encoding complex data into meaningful representations.

A key study by Ma et al. explores the linguistic information captured by BERT, a transformer model, through its layer-wise activations. Their research demonstrates that BERT's sentence embeddings are highly effective not only in question answering but also in SentEval tasks. This proficiency in generating robust embeddings extends to applications beyond NLP, such as time series prediction, as shown by [129].

The bioinformatics field, in particular, has also witnessed transformative applications of these advanced representation learning techniques. For instance, ESM2 [60] is a transformer protein language models trained to predict the identity of amino acids that have been randomly masked out of protein sequences. The representations learned by ESM2 have further been applied in ESMFold, significantly advancing the prediction of protein structures. Similarly, ProteinBERT [10] leverages deep language model representations to enhance gene ontology

annotation predictions. Beyond protein sequences, the ChemBERTa model [13] applies transformer technology to predict molecular properties, achieving competitive performance on the MoleculeNet benchmark tasks.

These advancements in bioinformatics highlight the transformative potential of representation learning techniques. By accurately encoding biological sequences into meaningful representations, transformer-based models have shown potential in various scientific researches, including binding affinity prediction.

# CHAPTER 3

# IMPROVING INTERACTION-FREE BINDING AFFINITY PREDICTION VIA PROTEIN TRANSFORMER AND GRAPH NEURAL NETWORK FUSION

In the rapidly advancing fields of de novo drug design [32, 18, 80] and drug/drug-like molecule optimization [137, 35, 36], docking scores play a pivotal role yet are often overlooked. These scores, crucial for assessing the structural alignment between a molecular structure and a reference structure, are instrumental in predicting the potential efficacy of drug candidates. However, accurately calculating docking scores through virtual screening tools, such as OEDocking [49] and Autodock Vina [107], requires significant computational resources, presenting a substantial challenge that limits their widespread application. This challenge underscores the urgent need for the development of machine learning (ML) and deep learning (DL) based surrogate models that can efficiently incorporate docking scores into our proposed drug optimization framework, thereby overcoming existing computational barriers.

Docking scores have traditionally been used to estimate protein–ligand binding affinity [130], a measure of the interaction strength between a protein and a ligand that is typically determined through experimental methods. The focus of ML and DL research in this area has predominantly been on predicting this binding affinity, reflecting a broader scientific interest in accurately capturing the efficacy of real-world drug-ligand interactions. Such accuracy is vital for the successful development of new therapeutic agents.

As discussed in section 2.3, existing DL-based methods for binding affinity prediction face several challenges: (1) Interaction-free vs. interaction-based methods: Interaction-based methods [54, 124], which rely on data with physical interaction information, suffer from limited generalization and efficiency during inference. This limitation arises because only protein-ligand pairs with experimentally or simulation-determined structures can be utilized.

In contrast, interaction-free models [77, 73] infer molecular docking and binding affinity from data that abstract away explicit physical interactions, accommodating the uncertainty inherent in exploring unknown interactions more effectively. (2) Unimodal vs. multimodal surrogate models: While unimodal networks often show limited performance compared to multimodal models [44, 117, 133, 134], the field is still exploring how to best leverage multimodal approaches for predicting binding affinities. (3) Limited diversity and volume of training data: The benchmark datasets [118, 17, 104] contain only experimentally determined physical interaction pairs and binding affinities. (4) Sparsity of dataset: Typically, proteins in the database are documented to interact with a single molecule and vice versa.

To address these challenges, our project proposes the following contributions: (1) We introduce two fusion methods to integrate representations from transformer with those from SOTA unimodal graph neural network. This integration aims to enhance protein representation learning, thereby improving the accuracy of binding affinity predictions. Furthermore, we train the model in an interaction-free manner to ensure better generalizability and efficiency during inference. (2) We simulate a new comprehensive dataset comprising 10 million docking scores across 10,000 proteins and 1,000 drugs. This dataset provides a rich resource for model training and validation, addressing the issue of limited data diversity and volume. (3) Through comprehensive experiments on the benchmark dataset PDBbind [118] and our simulated datasets, we demonstrate the efficacy of transformers in docking score and binding affinity prediction.

This project illustrates that transformer-based models have the potential to improve binding affinity and docking score predictions. Such findings highlight the effectiveness of transformer-based representation learning in the proposed drug optimization process.

## 3.1 Preliminaries

**3D Graphs.** Many real-world data can be modeled as 3D graphs. A 3D graph can be represented as $G = (\mathcal{V}, \mathcal{E}, \mathcal{P})$. Here, $\mathcal{V} = \{\mathbf{v}_i\}_{i=1,...,n}$ is the set of node features, where each $\mathbf{v}_i \in \mathbb{R}^{d_v}$ denotes the feature vector for node $i$. $\mathcal{E} = \left\{e_{ij}\right\}_{i,j=1,\cdots,n}$ is the set of edge features, where $e_{ij} \in \mathbb{R}$ denotes the edge feature vector for edge $ij$. $\mathcal{P} = \{P_i\}_{i=1,\cdots,n}$ is the set of position matrices, where $P_i \in \mathbb{R}^{k_i \times 3}$ denotes the position matrix for node $i$. $k_i$ can be different for different applications. For example, if we treat each atom in a molecule as a node, then $k_i = 1$ for each node $i$. For a protein, if we treat each amino acid as a node, then $k_i$ is the number of atoms in amino acid $i$. In our method, we represent proteins as 3D graphs and learn hierarchical representations of protein.

**Complete Message Passing Scheme.** By incorporating complete geometric representations to the commonly-used message passing framework, we achieve a complete message passing scheme as

$\mathbf{v}_i^{l+1} = \text{UPDATE}\left(\mathbf{v}_i^l, \sum_{j \in N_i} \text{MESSAGE}\left(\mathbf{v}_j^l, \mathbf{e}_{ji}, \mathcal{F}\left(G\right)\right)\right)$, where $\mathcal{N}_i$ denotes the set of node $i$'s neighbors, and UPDATE and MASSAGE functions are usually implemented by neural networks or mathematical operations.

**Amino Acid Level Representation.** Specifically, ProNet designs geometric representation at the amino acid level as

$$\{(d_{ij}, \theta_{ij}, \phi_{ij}, \tau_{ij})\}_{i=1,\cdots,n, j \in N}$$

In this context, $(d_{ij}, \theta_{ij}, \phi_{ij})$ denotes the spherical coordinates of node $j$ relative to the local coordinate system of node $i$. These coordinates determine the relative position of node $j$, where $d$, $\theta$, and $\phi$ represent the radial distance, polar angle, and azimuthal angle, respectively. Additionally, $\tau_{ij}$ captures the rotation angle of the edge $ji$, accounting for the remaining degree of freedom. Using this representation along with a complete message passing scheme,

a detailed and comprehensive representation of an entire 3D protein graph is achieved.

**Complete Geometric Representations.** Specifically, a geometric transformation $F(\cdot)$ is complete if for two 3D graphs $G^1 = (V, E, P^1)$ and $G^2 = (V, E, P^2)$, the geometric representations

$$F(G^1) = F(G^2) \Leftrightarrow \exists R \in SE(3), \text{ where } i = 1, \cdots, n, P_i^1 = R(P_i^2)$$

Here, SE(3) encompasses all possible rotations and translations in a 3D space, introduced to maintain the 3D conformation of a graph despite any rotations and translations, thereby preserving the inherent structure of the graph. In line with the settings used in ProNet, HoloProt is employed as the ligand network for a fair comparison.

Our primary goal is to demonstrate that the integration of transformer models with advanced fusion methods can significantly enhance protein representation learning, thereby improving the accuracy of binding affinity predictions.

Below we present two frameworks, serial fusion and adaptive fusion. These frameworks aim to harness the complementary strengths of each representation type, enhancing the overall predictive power while mitigating their individual limitations.

## 3.2 Leveraging Protein Language Models and GNNs for Binding Affinity Prediction

In this section, we propose two frameworks designed to integrate sequence and structural representations of proteins. These frameworks aim to harness the complementary strengths of each representation type, enhancing the overall predictive power while mitigating their individual limitations.

### *3.2.1 Serial Fusion: Using Sequence Representations as Protein Residue Features in Graph Neural Networks*

Integrating diverse modalities of data representation offers a promising avenue to enrich protein analysis. A notable approach in this context is serial fusion, exemplified by the ESM-GearNet model [134]. This model innovatively incorporates the output of the ESM into GearNet [133], by substituting GearNet's node features with those derived from ESM. The resultant representation benefits from the deep evolutionary insights encoded by PLMs, demonstrating the potential of combining PLMs with GNNs for advanced protein representation. However, the reliance of ESM-GearNet on self-supervised learning for pre-training poses questions about its adaptability and efficacy in supervised learning contexts.

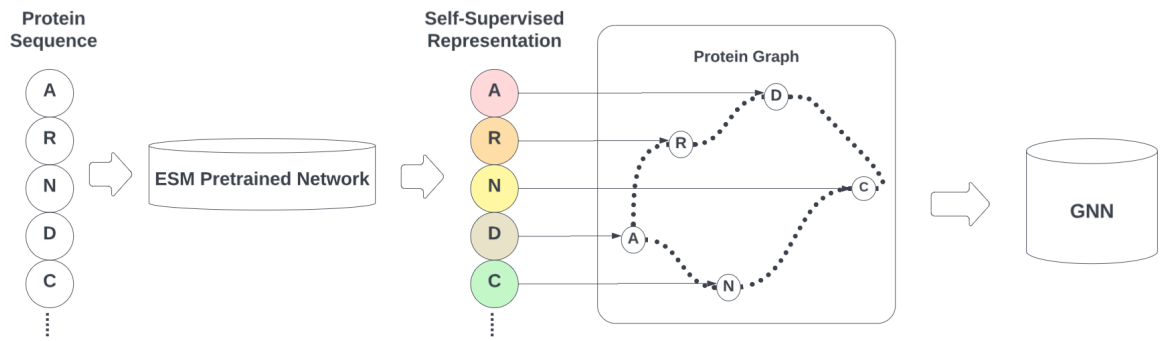



Figure 3.1: Overview of the Serial Fusion framework. The protein sequence is processed through the pre-trained protein language model, ESM, to generate per-residue representations. These representations are then employed as node features within 3D protein graphs for subsequent analysis by the baseline GNN, ProNet.

### 3.2.2 *Adaptive Fusion: Adaptively Merging Sequence ans Graph Representation Multiple and Mutual Information Interactions*

Inspired by the success of RPVNet [123] in fusing voxel, point cloud, and range image data for LiDAR analysis, we explore the potential of a similar multimodal fusion strategy for protein representation. Proteins, like LiDAR datasets, exhibit complex, multi-faceted structures that can benefit from an integrated analysis approach.

As depicted in Figure 3.2, our framework integrates sequence and graph representations, leveraging multiple and mutual information interactions. This method aims to capture a complete picture of the protein from various perspectives, with each representation providing unique insights to enhance the model's predictive accuracy.

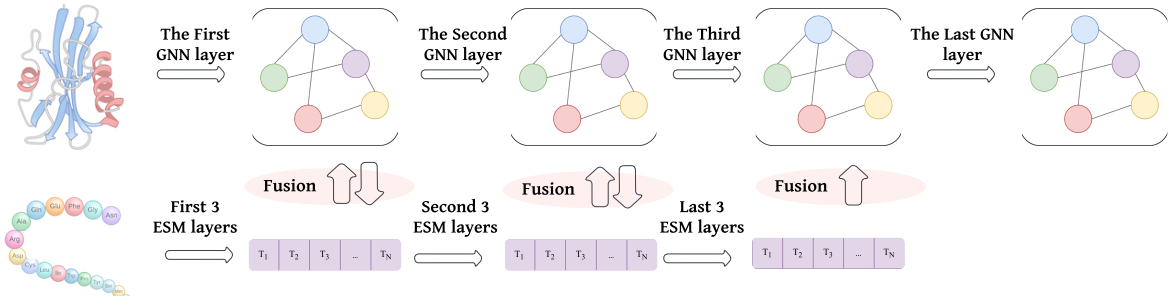



Figure 3.2: Overview of the framework. The proposed structure is a two-branch network, characterized by intricate interactions among its branches. Specifically, the sequence-branch leverages ESM, and the graph-branch employs the selected baseline GNN, ProNet.

To merge features from these representations effectively, we employ a gating mechanism [14, 103], as detailed in Figure 3.3. This approach dynamically weights the contributions from each modality, enhancing the integration process and illuminating the impact of individual modalities on the final prediction.

More specifically, given 2 feature vectors $X_i \in \mathbb{R}^{N_i \times C_i}$ from two different view branches, where $N_i, C_i$ are the number of points and channels of the ith feature vector respectively. The gated fusion layer is designed based on the normal addition-based fusion by filtering

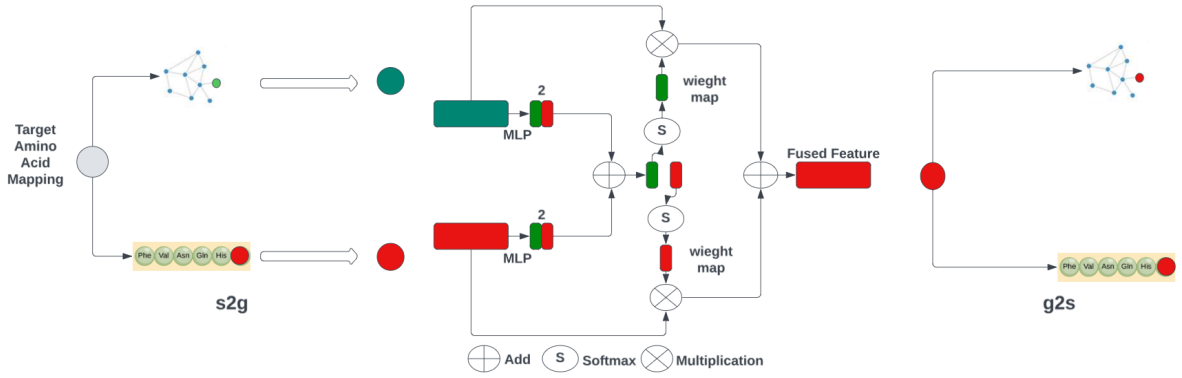



Figure 3.3: Details of fusion. In left block: Given an amino acid, we need to find the corresponding character in sequence and graph node. In the central block: When presented with features of an amino acid from varied representations, we merge them adaptively with gating mechanism. In the right block: Once the features are fused, the next step is to map them back to their respective representations.

information flow with gates, following RPVNet. The feature weight votes on each channel are superimposed by summation and converted into probability weights by softmax. Finally, the consequences on the corresponding channels are separated to weight the input features. The gated fusion is formally defined as:

$$\tilde{X} = split[softmax(\sum_{i}^{2} G_i)]_1 \cdot X_1 + split[softmax(\sum_{i}^{2} G_i)]_2 \cdot X_2 \tag{3.1}$$

where $G_i = w_i X_i + b_i, w_i \in \mathbb{R}^{2 \times C_i}, b_i \in \mathbb{R}^2$

$G_i \in [0,1]^{N_i \times 2}$

$\cdot$ denotes element-wise multiplication

In addition, this methodology not only facilitates the seamless integration of sequence and graph data but also enhances the model's interpretability by highlighting the contributions of different modalities to the final predictions.

### *3.2.3 Simulated Interaction-free Binding Affinity Dataset*

The limited volume and sparsity of experimentally determined protein-ligand complexes pose significant challenges in the field of DL. To address these challenges, we have simulated a comprehensive new dataset comprising 10 million docking scores across 10,000 proteins and 1,000 drugs. This dataset serves as a rich resource for model training and validation, effectively addressing the issues of limited data diversity and volume, as well as the sparsity of binding affinity values.

More specifically, we randomly selected 10,000 proteomes from human organisms available in AlphaFoldDB [110] and 1,000 drugs/molecules from DrugBank [121]. Our focus is on the effectiveness of real-world drugs on human organisms. Additionally, since our proposed fusion methods primarily focus on the protein side, we opted for a larger number of proteins relative to molecules. Consequently, we have generated 10 million protein-ligand complexes. To determine the docking scores, we first utilized Fpocket [53] to predict binding pockets within the proteins, selecting the most promising pocket based on druggability scores generated by Fpocket. With these identified pockets, we then employed Vina-GPU [21] to simulate the poses of molecules and their corresponding docking scores, selecting the best docking score for each complex.

In addition to training and evaluating our model's capacity to generalize to previously unseen data, we adopted a splitting strategy that improves upon the one used by Ong et al.. Specifically, instead of splitting the dataset by inhibitor/molecule alone, we split it on both the protein and molecule sides. This approach resulted in the dataset split depicted in Figure 3.4. We selected 15% of the ligands and 15% of the proteins, moving all datapoints corresponding to these molecules and proteins into the test set. This method yields three testing datasets: (unseen protein, seen molecule), (seen protein, unseen molecule), and (unseen protein, unseen molecule). Furthermore, from the training datapoints, we split an additional 10% of ligands and proteins into a fourth testing dataset (seen protein, seen molecule). This strategy allows

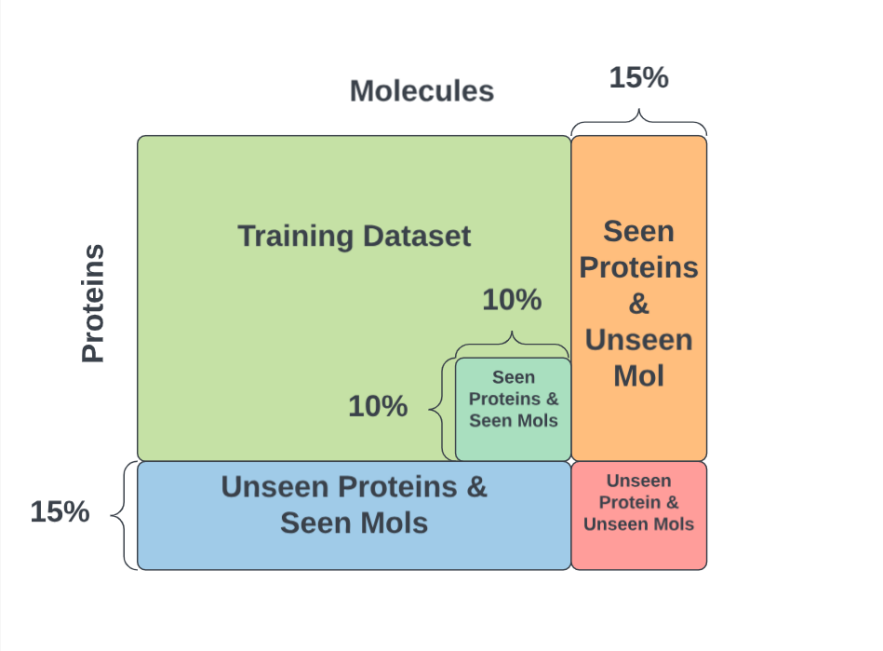



Figure 3.4: This figure illustrates the dataset splitting strategy employed to create distinct training and testing subsets. The cube represents the entire dataset, segmented into five parts: 1) Unseen protein, seen molecule; 2) Seen protein, unseen molecule; 3) Unseen protein, unseen molecule; 4) Seen protein, seen molecule, which constitutes the fourth testing dataset; and 5) The remaining data designated as the training dataset. Each segment's color coding facilitates an intuitive understanding of how the data is divided, ensuring a comprehensive evaluation of the model's generalization capabilities across both familiar and novel proteins and molecules.

us to evaluate how well a model generalizes to both proteins and molecules, whether they were seen or unseen during training.

## 3.3 Experiments

### *3.3.1 Experiment Setup*

**Baseline and proposed fusion methods** To evaluate the effectiveness of our fusion methods, we establish ProNet [115] as our primary benchmark. ProNet is a cutting-edge model designed for protein representation learning, emphasizing the incorporation of 3D

structural information. It has achieved outstanding results in several downstream tasks, particularly in binding affinity prediction. For a balanced comparison, especially from the perspective of molecule representation, we employ the same ligand network, HoloProt [100], used alongside ProNet. This approach ensures a fair evaluation framework by maintaining consistency in the molecular component of our experiments. In our experimental setup, we adhere to the default parameters specified by the authors of ProNet and HoloProt to maintain methodological integrity. Our exploration into fusion methods involves experimenting with various sizes of the ESM, in conjunction with the default configurations of ProNet and HoloProt. And the model with best performance is reported.

**Dataset.** Our study evaluates the proposed methods across two datasets: the PDBbind dataset [118] and the simulated dataset detailed in Section 3.2.3. For the PDBbind dataset, we follow the train/test split as defined by ProNet [115], which employs 30% and 60% sequence identity thresholds. Sequence identity refers to the percentage of amino acids that are identical when two protein sequences are aligned, providing a measure of their similarity. By using these thresholds, we aim to test the generalization capability of our models on unseen proteins that vary in similarity to those in the training set. However, to accommodate our model's limitations, we introduce a modification: proteins with sequence lengths exceeding 1024 amino acids are excluded. This is because our model, based on the ESM framework, truncates sequences longer than this threshold, potentially omitting critical information for our analysis. This adjustment ensures the inclusion of only those sequences that our models can fully process, maintaining the integrity of our evaluation. Conversely, the simulated dataset is partitioned following the specific guidelines outlined in Section 3.2.3. These guidelines are designed to challenge our models under controlled conditions, providing a distinct perspective on their capabilities. This dual-dataset approach allows us to comprehensively assess the performance and applicability of our methods across both real-world and theoretical scenarios.

**Evaluation metric.** To rigorously assess the performance of our model in predicting binding affinity, we employ three distinct metrics same as ProNet [115]. These metrics are:

1. **Root Mean Square Error (RMSE):** RMSE is a widely used measure of the differences between values predicted by a model and the values actually observed. It is particularly useful in quantitatively assessing the magnitude of prediction error. The formula for RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where $y_i$ represents the observed values, $\hat{y}_i$ represents the predicted values, and $n$ is the number of observations. A lower RMSE value indicates a better fit to the data.

2. **Pearson Correlation Coefficient (Pearson):** The Pearson correlation coefficient measures the linear correlation between two sets of data. It assesses the linear relationship between the predicted and observed affinities. The coefficient ranges from -1 to 1, where 1 means a perfect positive linear relationship, -1 means a perfect negative linear relationship, and 0 indicates no linear relationship. The Pearson correlation coefficient is calculated as:

$$Pearson = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the mean values of the observed and predicted affinities, respectively. A higher Pearson coefficient indicates a stronger linear relationship between the predicted and actual values.

3. **Spearman's Rank Correlation Coefficient (Spearman):** Spearman's coefficient measures the strength and direction of the monotonic relationship between two datasets. By comparing the rank order of the data points, Spearman's correlation is less sensitive to outliers than Pearson. This makes it particularly useful for non-linear data or when

the assumption of normality is not met. Spearman's coefficient is calculated as:

$$Spearman = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where $d_i$ is the difference between the ranks of corresponding values in the two data sets, and $n$ is the number of observations. A Spearman coefficient close to 1 or -1 indicates a strong monotonic relationship, while a coefficient around 0 suggests no such relationship.

Together, these three metrics provide a comprehensive evaluation of our model's performance in predicting binding affinities. RMSE offers insight into the average error magnitude, Pearson assesses the linear correlation between predicted and observed values, and Spearman evaluates the monotonic relationship, offering a nuanced view of the model's predictive accuracy and reliability.

### *3.3.2 Experimental Results*

| Split | Method | RMSE ↓ | Pearson ↑ | Spearman ↑ |
|---|---|---|---|---|
| **60% Sequence Identity** | **ProNet** | 1.477 | 0.703 | 0.704 |
| | **Serial Fusion** | <u>1.350</u> | **0.769** | **0.759** |
| | **Adaptive Fusion** | **1.337** | <u>0.760</u> | <u>0.757</u> |
| **30% Sequence Identity** | **ProNet** | 1.544 | 0.499 | 0.488 |
| | **Serial Fusion** | <u>1.478</u> | **0.573** | **0.554** |
| | **Adaptive Fusion** | **1.439** | <u>0.547</u> | <u>0.543</u> |

Table 3.1: Binding affinity experimental results on PDBbind at 60% and 30% sequence identity levels. The top two results are highlighted as **1st** and <u>2nd</u>.

The experimental results on the PDBbind dataset, as shown in table 3.1, reveal significant differences in the performance of the three methods when dealing with varying levels of sequence identity. At 60% sequence identity, Adaptive Fusion outperforms the other methods in terms of RMSE, indicating its superior accuracy in predicting binding affinity with a minimal

error margin. Although Serial Fusion shows the highest Pearson and Spearman correlation coefficients, the margin between Serial and Adaptive Fusion is relatively narrow, suggesting that both methods are competitive in capturing the linear and rank-order relationships between predicted and actual binding affinities. The disparity in method performance becomes more pronounced at 30% sequence identity. Here, Adaptive Fusion demonstrates a remarkable improvement in RMSE, significantly outperforming both ProNet and Serial Fusion. This improvement underscores Adaptive Fusion's robustness and adaptability to challenging conditions with low sequence similarity. Furthermore, while Serial Fusion still leads in Pearson and Spearman metrics, the gap between Serial and Adaptive Fusion narrows, highlighting Adaptive Fusion's consistent performance across different evaluation metrics.

| **Split with Unseen Protein** | **Method** | RMSE ↓ | Pearson ↑ | Spearman ↑ |
|---|---|---|---|---|
| **Seen Molecule** | **ProNet** | 1.209 | 0.838 | 0.840 |
| | **Serial Fusion** | <u>1.176</u> | <u>0.849</u> | <u>0.854</u> |
| | **Adaptive Fusion** | **1.121** | **0.863** | **0.865** |
| **Unseen Molecule** | **ProNet** | 1.245 | 0.833 | 0.845 |
| | **Serial Fusion** | <u>1.199</u> | <u>0.847</u> | <u>0.860</u> |
| | **Adaptive Fusion** | **1.140** | **0.862** | **0.871** |

Table 3.2: Binding affinity experimental results on Simulated dataset - Unseen Protein. The table presents RMSE, Pearson, and Spearman for scenarios with seen and unseen molecules. The top two results are highlighted as **1st** and <u>2nd</u>

| **Split with Seen Protein** | **Method** | RMSE ↓ | Pearson ↑ | Spearman ↑ |
|---|---|---|---|---|
| **Seen Molecule** | **ProNet** | 0.419 | 0.983 | 0.984 |
| | **Serial Fusion** | 0.398 | 0.985 | 0.986 |
| | **Adaptive Fusion** | 0.426 | 0.983 | 0.984 |
| **Unseen Molecule** | **ProNet** | 0.500 | 0.976 | 0.977 |
| | **Serial Fusion** | **0.481** | **0.978** | **0.979** |
| | **Adaptive Fusion** | <u>0.490</u> | <u>0.977</u> | <u>0.978</u> |

Table 3.3: Binding affinity experimental results on Simulated dataset - Seen Protein. The table presents RMSE, Pearson, and Spearman for scenarios with seen and unseen molecules. The top two results are highlighted as **1st** and <u>2nd</u>

The analysis of results from the simulated dataset for unseen proteins, as shown in table

3.2. shows that all methods generally perform better when predicting binding affinities for seen molecules compared to unseen molecules. This is expected as seen molecules provide a familiarity advantage that models can leverage for more accurate predictions. Notably, Adaptive Fusion again stands out, especially in the unseen molecule scenario, achieving the lowest RMSE and the highest Pearson and Spearman correlation coefficients. This indicates its exceptional ability to generalize from known to unknown molecular interactions, a crucial attribute for practical applications in drug discovery where novel targets are frequent.

When evaluating the performance on seen proteins with both seen and unseen molecules, as shown in table 3.3, it's evident that the performance trends are consistent with those observed in the unseen protein scenario. However, the overall metrics are significantly better across all methods, which can be attributed to the advantage of training models on data that include the same proteins as in the test set. Despite this, the relative performance of Adaptive Fusion in the context of unseen molecules suggests a slight decrease compared to its performance on seen molecules, indicating the challenges inherent in generalizing to completely novel molecular entities.

The comparative analysis across different experimental setups and sequence identities highlights the strengths and limitations of ProNet, Serial Fusion, and Adaptive Fusion in predicting binding affinities. Adaptive Fusion consistently demonstrates superior or competitive performance, especially under challenging conditions of low sequence identity and when dealing with unseen molecules. This suggests that Adaptive Fusion's approach to integrating diverse data sources and adapting to the complexity of the task makes it a promising method for accurate binding affinity prediction in drug discovery.

## 3.4 Conclusion

In order to maximize the benefit of both PLM and GNN for representing proteins for prediction tasks, we design two novel frameworks in protein representation, serial fusion

and the dynamic adaptive fusion. By experimentation, we demonstrated that both variants greatly improve upon the previous state-of-the-art system, proving the benefit of dynamically merging representations from two views, the view of PLM and that of GNN.

We believe the framework is versatile and adaptable to any future GNN and PLM, and can benefit other GNNs and PLMs for other downstream tasks involving proteins. One constraint of our framework is that it requires that the PLM and GNN somehow represent nodes of the graph at the same level of, and does not yet have a way to utilize hierarchical structures of systems with multi-scale representations. We leave this extension to future work.

# CHAPTER 4

# DRUG OPTIMIZATION WITH TRANSFORMER-PREDICTED DOCKING SCORES IN A MULTI-OBJECTIVE REINFORCEMENT LEARNING FRAMEWORK

The journey of drug development is a daunting one, often spanning 14 years and incurring costs of up to $1 billion. This process is challenged by inefficiencies and high failure rates, driving the pharmaceutical industry to seek innovative strategies to accelerate drug discovery. Among these strategies, computer-aided de novo drug design and the development of me-too drugs are particularly noteworthy. De novo drug design leverages computational methods to create novel molecular entities from scratch, aiming for specific desired properties. In contrast, me-too drugs are developed through minor modifications to existing drugs to enhance efficacy or reduce side effects. Despite the popularity of me-too drugs [4], there is a noticeable gap in the application of computational methods for their development compared to de novo drug design, highlighting a substantial opportunity for progress in the field.

The urgency for more rapid drug development is further amplified by the emergence of rapidly evolving virus variants [34], such as those associated with SARS-CoV-2 [128], and drug-resistant cancer cells [68]. These challenges underscore the need for swift adaptation in drug design, where the integration of artificial intelligence (AI) and machine learning technologies could play a pivotal role. Specifically, these technologies hold promise for enhancing the development of me-too drugs, offering a pathway to quickly adapt existing drugs to combat fast-evolving threats.

This work introduces DrugImprover, a deep learning-based drug optimization framework designed to automate and expedite the creation of effective me-too drugs. DrugImprover aims to adapt existing drugs to combat fast-evolving virus variants and cancer cells efficiently. We aim to improve various properties of an original drug in a robust and efficient manner. To

achieve this, we address the significant challenges encountered in current de novo drug design and molecule optimization methods, as detailed in sections 2.1 and 2.2: (1) Data-driven generative models vs. RL-based models: Generated molecules must fulfill multiple criteria, including solubility and synthesizability. Data-driven generative models, which are pre-trained on specific datasets to sample from a learned distribution, faces limitations in exploration due to biases present in the training data and constrained ability to control the properties of the generated molecules [33, 58, 35, 36]. In contrast, RL-based works offers a complementary approach by using a reward system to iteratively guide the generation of novel molecular structures towards desired properties [32, 18, 137]. However, these models often overlook docking score, a crucial metric for assessing structural compatibility with a target, mainly due to the computational cost of calculating it through virtual screening. (2) Search space complexity: An RL algorithm for drug discovery needs to demonstrate both sample and computational efficiency. However, the overwhelming complexity of the search space [79] renders RL incapable of adequately exploring potential effective actions and states required for policy learning. (3) Sparse rewards: In contrast to the continuous reward environment found in popular environments like DeepMind Control Suite [105] or Meta-World [127], drug generation operates within a sparse reward environment where rewards are only obtainable upon a complete molecule. (4) Preservation of original beneficial properties: As drugs with similar chemical structures should exhibit similar biological/chemical effects [8], it is crucial to strike a balance between optimizing the drug and preserving the original drug's beneficial properties.

To address these challenges, DrugImprover incorporates the Advantage-alignment Policy Optimization (APO) algorithm, which utilizes the advantage preference to perform direct policy improvement under the guidance of multiple critics. This approach tackles the outlined challenges by: (1) Multiple objectives: APO employs multiple critics, each of which serves as an evaluator with domain-specific expertise, such as knowledge related to solubility,

synthesizability, druglikeness and docking score. In particular, as demonstrated by project 3 where transformers show their power in binding affinity and docking score prediction, we adopt a transformer-based surrogate model [112] to obtain docking scores more efficiently. These multiple critics guide the exploration in the drug refinement process toward the improved properties. (2) Sample complexity and sparsity: Because of the sparse reward nature of the drug design, pure RL often finds it challenging to learn a good policy due to the complexity of the search space. To reduce this complexity, APO employs an imitation-learning-based approach to initialize a generator policy with desirable behavior based on prior experience of designing drug SMILES strings. APO also addresses the problem of reward sparsity by adapting Monte-Carlo sampling to obtain estimated rewards for intermediate steps. (3) Property preserving: To preserve the original drug's beneficial properties, throughout the optimization process, it is crucial to balance the preservation of the original drug's beneficial properties with the optimization of other chemical attributes. To achieve this, we use Tanimoto similarity as a critic to maximize the Tanimoto similarity between the original and generated drugs.

In summary, our contributions are (1) We introduce DrugImprover, a framework tailored for efficient drug optimization. Within DrugImprover, we incorporate APO algorithm that performs advantage-alignment policy optimization with multi-critic guided exploration. In addition, we use a transformer-based surrogate model to obtain docking scores more efficiently and Tanimoto similarity rewards to maintain similar structures with original molecules. (2) By conducting comprehensive experiments on real world viral and cancer target proteins, we illustrate that APO consistently enhances existing molecules/drugs across all desired objectives, leading to improved drug candidates.

The DrugImprover framework, powered by the APO algorithm, transformer-based docking score surrogate model and Tanimoto similarity rewards, presents a novel solution to the challenges of me-too drug development. By directly leveraging the advantage preference

of generated drugs over the original based on multiple objectives, this approach not only streamlines the optimization process but also ensures that the original drug's beneficial properties are retained.This research opens up new possibilities for enhancing drug optimization and inspires future investigations into addressing challenges within the realm of drug optimization.

## 4.1 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 $d_t^\pi$ as the distribution over states at time $t$ under policy $\pi$. The goal is to train a policy to maximize the expected long-term reward. The quality of the policy can be measured by the $Q$-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as:

$$Q^\pi(s, a) := \mathbb{E}^\pi \left[ \sum_{t=0}^{T} R(s_t, a_t) \,|s_0 = s, a_0 = a \right], \tag{4.1}$$

where the expectation is taken over the trajectory following $\pi$, and the value function is as follows:

$$V^\pi(s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s, a)]. \tag{4.2}$$

**Drug generation process.** We formalize the drug generation problem within the framework of Markov Decision Processes (MDP). Given a dataset consisting of real-world structured sequences represented as SMILES [119] strings, our objective is to train a generative policy $\pi_\theta^G$ to generate a high-quality sequence denoted as $Y_{1:T} = (y_1, \ldots, y_t, \ldots, y_T), y_t \in \mathcal{Y}$. Here, $\mathcal{Y}$

represents the vocabulary of potential SMILES tokens, constituting the action space denoted as $\mathcal{A}$. The length of the sequence, denoted as $T$, represents the planning horizon. At time step $t$, the state $s_{t-1}$ comprises the currently generated tokens $(y_1, \ldots, y_{t-1})$, and the action $a$ corresponds to the next token $y_t$ to be selected. While the policy model $\pi_\theta^G(y_t|Y_{1:t-1})$ operates in a stochastic manner, the state transition function $\mathcal{P}$ becomes deterministic once an action has been chosen. The primary objective of the generator policy $\pi_\theta^G$ is to initiate the generation process from an initial state $Y_1$ and maximize the expected final reward at the end of the sequence:

$$J(\theta) = \mathbb{E}_{Y_1 \sim d_0^{\pi_\theta^G}}[r_T|\theta], \tag{4.3}$$

where $r_T$ represents the reward associated with a fully generated sequence. To estimate the $Q$ value, we reference the REINFORCE algorithm [120], which we define as follows:

$$Q(s = Y_{1:T-1}, a = y_T) = R(Y_{1:T}). \tag{4.4}$$

Nonetheless, the reward function only supports a reward value for a completed sequence. In our case, we aim to compute the $Q$ for partial sequences at intermediate time steps, accounting for the expected future reward upon sequence completion. To achieve this, we employ a Monte Carlo search approach and Roll-in-Roll-out (RIRO) [12, 62, 89] scheduling, utilizing a roll-out policy denoted as $\pi_\beta$ to sample the unknown last $T - t$ tokens. We represent an N-time Monte Carlo search as follows:

$$\left\{Y_{1:T}^1, ..., Y_{1:T}^N\right\} = MC^{\pi_\beta}(Y_{1:t}; N), \tag{4.5}$$

where $Y_{t+1:T}^N$ is sampled based on the roll-out policy $\pi_\beta$ and the current state $Y_{1:t}^n$ is stochastically sampled via the roll-in policy $\pi_\theta^G$. In our experiment, we set $\pi_\beta$ to be identical to the learner policy $\pi_\theta^G$, although it can alternatively be an oracle policy if one is accessible. To enhance the precision of expected $Q$ value assessment, we execute the roll-out policy from

the current state to the end of the sequence N times and estimate its averaged rewards on a batch of complete samples. Thus:

$$Q\left(s=Y_{1:t-1},a=y_t\right)=\begin{cases}\frac{1}{N}\sum_{n=1}^{N}R\left(Y_{1:T}^{n}\right),\text{where }Y_{1:T}^{n}\in MC^{\pi_\theta^G}\left(Y_{1:t};N\right),\text{if t}<T,\\ R\left(Y_{1:t}\right),\text{if t}=T.\end{cases} \tag{4.6}$$

Here, $Q^{\pi_\theta^G}(s,a)$ stands for the action-value function, which represents the expected reward at state $s$ of taking action $a\sim\pi_\theta^G(s)$ and following the current policy $\pi_\theta^G$ to complete the sequence. Policy gradient optimizes a parameterized policy to maximize the expected total reward by repeatedly estimating the gradient $g:=\nabla_\theta J(\theta)$. There is a general form for the policy gradient [94, 102]:

$$g=\sum_{t=1}^{T}\mathbb{E}_{y_t\sim\pi_\theta^G(y_t|Y_{1:t-1})}\left[\nabla_\theta\log\pi_\theta^G\left(y_t|Y_{1:t-1}\right)\cdot\Phi_t\right], \tag{4.7}$$

where $\Phi_t$ could be in several forms. One common choice for $\Phi_t$ in previous drug discovery work is $Q\left(Y_{1:t-1},y_t\right)$ [32, 126].

**Limitations of previous work.** 1) Prior studies concentrated primarily on the discovery of new drugs from the ground up [6, 80, 132]. In contrast, we focus on the relatively less explored, yet highly practical and significant, issue of drug optimization. In drug optimization, the goal is to enhance an existing drug according to multiple objectives while preserving a similar chemical structure. 2) Earlier research employed $Q$ [32, 126] in gradient calculations, which can introduce high variance and potentially lead to divergence. Our advantage-alignment policy gradient approach avoids this problem.

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

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

Initialize $\pi_\theta^G$ with random weight $\theta$.
Pre-train $\pi_\theta^G$ usng MLE on $\mathcal{B}$.
$\beta \leftarrow \theta$.
**for** $n = 1, \ldots, N$ **do**
    $s_0 \sim \rho_0$, where $\rho_0 \in \Delta(\mathcal{B})$.
    Generate a sequence $Y_{1:T} = (y_t, \ldots, y_T) \sim \pi_\theta^G(\cdot|s_0)$.
    Compute advantage preference $R^{\text{Advantage-Preference}}$ by (4.6)(4.9)(4.13)(4.14).
    Update generator parameters via policy gradient by (4.16)(4.17).
    $\beta \leftarrow \theta$.

## 4.2 DrugImprover Framework for Drug Optimization

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

### *4.2.1 Advantage-alignment policy optimization with multi-critic guidance algorithm*

**Multi-critic guidance.** Given an ensemble of critics

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

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

$$\mathbf{W} = \left[W^{\text{Druglikeness}}, W^{\text{Solubility}}, W^{\text{Synthesizability}}, -1 \cdot W^{\text{Docking}}, W^{\text{Tanimoto}}\right]. \tag{4.8}$$

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

$$R_c\left(Y_{1:T}\right) := R_c\left(Y_{1:T}|s_0\right) = \sum_{t=1}^{T}\sum_{n=1}^{N} \text{Norm}\left(\mathbf{C}\left(s_0, Y_{1:T}^n\right)\right) \cdot \mathbf{W}, \quad Y_{1:T}^n \in MC^{\pi_\theta^G}\left(Y_{1:t}; N\right). \tag{4.9}$$

We use Norm[1] to normalize different attributes onto the same scale. In this study, we employ the Tanimoto similarity calculation $C^{\text{Tani-Similarity}}$ to quantify the chemical similarity between the generated compound and the original drug. Essentially, this calculation involves first computing Morgan Fingerprints [88] for each molecule and then measuring the Jaccard distance [43] (i.e., intersection over union) between the two fingerprints.

**Advantage-alignment policy gradient.** The return, denoted as $Q^\pi$, often exhibits significant variance across multiple episodes. One approach to mitigate this issue is to subtract a baseline $b\left(s\right)$ from each $Q$. The baseline function can be any function, provided that it remains invariant with respect to $a$. For a generator policy $\pi_\theta^G$, the advantage function [102] is defined as follows:

$$\boldsymbol{A}^{\pi_\theta^G}\left(s,a\right) = Q^{\pi_\theta^G}\left(s,a\right) - b\left(s\right) \tag{4.10}$$

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

$$V\left(s\right) = \mathbb{E}_{a\sim\pi_\theta^G(s)}[Q(s,a)] = \mathbb{E}_{y_t\sim\pi_\theta^G(Y_{1:t-1})}[Q(Y_{1:t-1}, y_t)] \tag{4.11}$$

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

Thus, we have advantage function as

$$\boldsymbol{A}^{\pi_\theta^G}(s,a) = \boldsymbol{A}^{\pi_\theta^G}(Y_{1:t-1}, y_t) = Q^{\pi_\theta^G}(Y_{1:t-1}, y_t) - V^{\pi_\theta^G}(Y_{1:t-1}). \tag{4.12}$$

*Remark* 4.2.1. When compared to $Q^{\pi_\theta^G}$ as described in (4.7), the selection of $\boldsymbol{A}^{\pi_\theta^G}$ tends to result in potentially lower variance. This assertion can be intuitively justified by considering the interpretation of the policy gradient: the direction of a step in the policy gradient should increase the probability of actions better than the average and decrease the probability of actions worse than the average. The advantage function essentially gauges whether an action is superior or inferior to the policy's default behavior. Consequently, we opt to designate $\Phi_t$ as the advantage function $\boldsymbol{A}^{\pi_\theta^G}$. This choice ensures that the gradient term $\Phi_t \nabla_\theta \log \pi_\theta^G(a_t|s_t)$ aligns with an increase in $\pi_\theta^G(a_t|s_t)$ only when $\boldsymbol{A}^{\pi_\theta^G}(a,s) > 0$. This is contrast with previous method using $Q$. For a more thorough examination of the variance of policy gradient estimators and the impact of employing baseline, please refer to [30].

**Drug optimization.** In the drug optimization problem, the primary objective is different from objective of drug generation problem in (4.3). In this work, we employ the one-step RL [11, 78] method and regard the drug optimization method as a sequence to sequence language generation task. Rather than treating each token as an individual action, we treat the entire sequence $Y_{1:T}$ as a single action generated by the policy $\pi_\theta^G$. Subsequently, we receive rewards from critics, and the episode concludes. This leads to the formulation of our advantage function as follows:

$$\boldsymbol{A}^{\pi_\theta^G}(s,a) = Q^{\pi_\theta^G}(Y_{1:t-1}, y_t)\,|_{t=T} - V^{\pi_\theta^G}(s_0) = R_c(Y_{1:T}) - R_c(s_0), \tag{4.13}$$

where $s_0$ is the initial state sequence drawn from the distribution $\rho_0$, which corresponds to our buffer known as $\mathcal{B}$ containing selected SMILES strings. Thus, by applying amplifier $\gamma$,

the advantage preference of the generated versus the original drug is

$$R^{\text{Advantage-Preference}}\left(s_0, Y_{1:T}\right) = \gamma_n \left(R_c\left(Y_{1:T}\right) - R_c\left(s_0\right)\right), \tag{4.14}$$

where $\gamma_n \in \mathbb{R}^+$ represents an amplifier of advantage preference at n-th episode, $n \in [N]$, that controls the aggressiveness or conservatism in performing policy gradient updates. The advantage preference of (4.14) will be employed directly in the policy gradient (4.16) to finetune the generator policy $\pi_\theta^G$. The rationale behind the advantage preference is to produce a sequence that surpasses the initial state sequence $s_0$ in every objective. In this work, our objective is to maximize the expected final advantage preference compared to the original drug $s_0$ at the end of the sequence as follows

$$J\left(\theta\right) = \mathbb{E}_{s_0 \sim \rho_0}\left[R_T^{\text{Advantage-Preference}} | s_0, \theta\right], \tag{4.15}$$

Thus, we have gradient as follows:

$$g = \mathbb{E}_{Y_{1:T} \sim \pi_\theta^G(\cdot|s_0), s_0 \sim \rho_0}\left[\nabla_\theta \log \pi_\theta^G\left(Y_{1:T}|s_0\right) \cdot R^{\text{Advantage-Preference}}\left(s_0, Y_{1:T}\right)\right], \tag{4.16}$$

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

$$\theta \leftarrow \theta + \alpha_n g, \tag{4.17}$$

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

### *4.2.2 DrugImprover Framework*

**Step 1: Pre-training a generator policy $\pi_\theta^G$.** First we use ZINC dataset [42] to pre-train the $\pi_\theta^G$ policy using a self-supervised imitation learning approach.

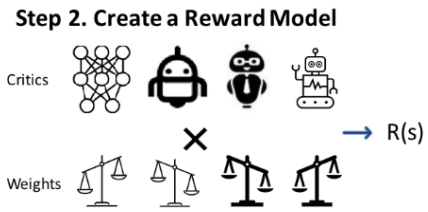



Figure 4.1: We define our reward model $R_c$ based on selected objectives, which are further balanced assigning different weight to each objective.

**Step 2: Creating a reward model $R_c$.** Next we define our reward model on the following objectives: (1) The docking score calculated by transformer-based surrogate model [112]. (2) Solubility. (3) Synthetizability. (4) Tanimoto similarity to the initial molecule. Each objective serves as a domain-specific critic, with each critic individually specializing in and optimizing for a specific molecular property. We balance these objectives by assigning different weight to each critic.

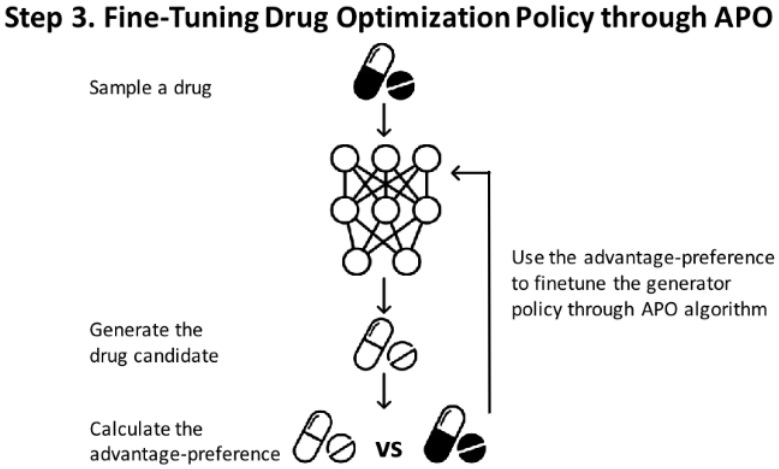



Figure 4.2: Step 3 of DrugImprover Framework. We use Advantage-alignment Policy Optimization with multi-critic guided exploration algorithm (APO) for drug optimization.

**Step 3: Performing objective-oriented policy finetuning:** Finally, we fine tune the original drug based generator on the reward model $R_c$. These reward critics are tailored to

optimize the learner policy $\pi_\theta^G$ with reward signals that align with their respective specialties. Subsequently, we apply the proposed Algorithm 1 to finetune the learner policy $\pi_\theta^G$ for improving the drug optimization process.

## 4.3 Experiments

### *4.3.1 Experiment Setup*

**Baselines and sequence generative model.** In our experimental setup, we compare our approach against three representative baselines: Maximum Likelihood Estimation (MLE) and ORGAN [32] and Naive RL. For the MLE baseline, we utilize the pretrained LSTM-based generator $\pi_\theta^G$, without proceeding further to finetune the model. ORGAN is a RL-based method for drug discovery, utilizing policy gradient based on the Q-value and employing a combination of a discriminator and domain objectives as rewards. Naive RL is using the same architecture as ORGAN, except that it gives zero weight to the discriminator. Appendix (B.3, B.5, B.7) provides further details.

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

**Datasets.** The dataset used for training the surrogate models is built with a similar scheme as in an earlier virtual screening on SARS-CoV-2 targets [7, 15]. Each datapoint has an input SMILES string representing the molecule and an output docking score. The receptors used are prepared with the OEDOCK application and FPocket [53] is used if the protein

active site is unknown. The score for each molecule is determined by inputting the molecular structure and receptor to OEDOCK and computing the minimum Chemgauss4 score over the ensemble of poses in the docking simulation. A set of 1 million orderable compounds within the ZINC15 dataset were docked to the 3CLPro (PDBID: 7BQY) SARS-CoV-2 protein and the RTCB (PDBID: 4DWQ) cancer protein. The resulting datasets are used for training two separate surrogate models for each protein.

**Critics and evaluation metric.** In this study, we evaluate the efficacy of APO in generating molecules with desirable attributes within the context of pharmaceutical drug discovery. We leverage the RDKit [52] chemoinformatics package and employ various performance metrics as follows: **Druglikeness:** The druglikeness measure the likelihood of a molecule being suitable candidate for a drug. **Solubility:** This metric assesses the likelihood of a molecule's ability to mix with water, commonly referred to as the water-octanol partition coefficient (LogP). Calculation is performed using RDKit's Crippen function. **Synthetizability:** This parameter quantifies the ease (score of 1) or difficulty (score of 0) associated with synthesizing a given molecule [22]. **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 B.4) to output this score.

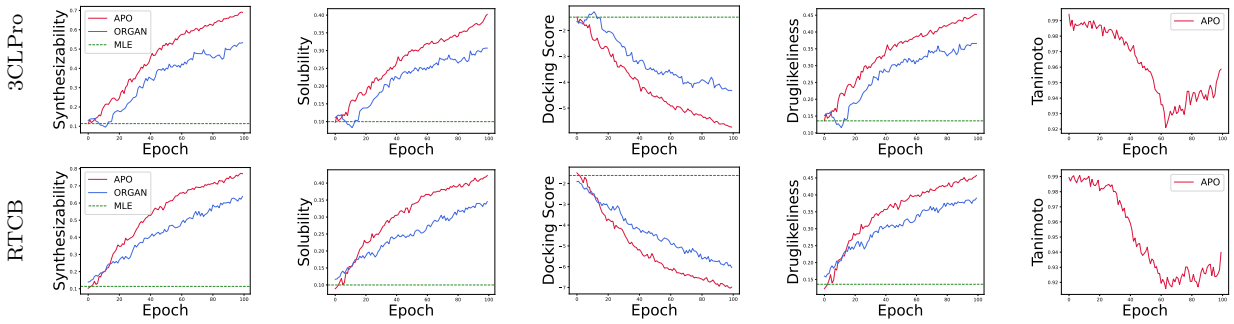
### *4.3.2 Experimental results*

Table 4.3 and Fig. 4.1 demonstrate that APO outperforms both MLE and the RL-based drug discovery baseline, ORGAN and Naive RL, across all the performance metrics for both viral and cancer-related proteins. Furthermore, APO not only surpasses all the baseline methods but also achieves a high Tanimoto similarity compared to the original drug. This suggests that it retains the beneficial properties of the original drugs while enhancing others.

Two main factors contribute to APO outperforming ORGAN. The first factor is the

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

Table 4.1: **Main results.** A comparison between three baselines {MLE, ORGAN, Naive RL} with DrugImprover/APO on objectives {druglikeness, synthesizability, solubility, docking score, Tanimoto similarity} based on 3CLPro and RTCB datasets. The presented values represent the mean values of generated molecules and Tanimoto similarity is measured on valid molecules. Values displayed in bold indicate notable improvements, and the percentage of improvement over the MLE baselines is enclosed in parentheses.




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

Figure 4.3: Visualize the performance curve associated with Table 4.1, featuring DrugImprover/APO (in red), and the baselines: MLE (in green), ORGAN (in blue), Naive RL (in yellow).

APO algorithm employs advantage-alignment, which increases the probability of generating the sequence only when it exhibits a positive preference advantage and decreases it when the advantage is negative. In contrast, ORGAN consistently increases the probability of sampled actions for positive rewards, leading to faster convergence of the APO algorithm. Additionally, APO employs the Tanimoto similarity constraint, which enables the generator policy to explore a nearby molecular domain in relation to the original one. This increases the probability of preserving chemical scaffolds and functional groups that are beneficial for binding to target proteins and dissolving in solvents. Note that the performance curve of Tanimoto similarity in Fig. 4.3e initially decreases and then increases. This trend aligns

ideally with the RL-based molecule generation improvement process. The initial decrease occurs because RL reduces the complexity of the original molecule to enhance the validity of the generator policy while improving the generated molecules' diversified properties. This causes the molecule to deviate from its original structure, leading to a decrease in Tanimoto similarity. Subsequently, there is a gradual increase in the trend as the generated molecules reach a decent level of diverse properties and begin optimizing their structure towards that of the original molecule, resulting in an increasing trend in Tanimoto similarity. Finally, the generated molecules not only improve the desired properties but also achieve a high Tanimoto similarity to the original drug. This reduces the likelihood of drastic structural changes that might result in unsynthesizable compounds. This process demonstrates that APO achieves a balance between optimizing desired properties and preserving the beneficial properties of the original drug.

## 4.4 Conclusion

We present DrugImprover, a practical and effective framework for drug optimization. Within the framework, we introduce APO, an advantage-alignment policy gradient algorithm with multi-critic guided exploration. This algorithm aims to align the generator policy with objectives from multiple critics and performs policy gradient updates based on the advantage preference. APO seeks to achieve maximal improvement based on the original drug while maintaining its necessary properties. Finally, we evaluate the docking score of our optimized compounds to two proteins, 3CLPro and RTCB, which are target proteins of SARS-CoV-2 and human cancer, respective. Our results reveal that our optimized compounds exhibit significantly stronger binding affinity to both proteins compared to compounds generated using baseline methods. Moreover, our compounds outperform those from the baseline method across all performance metrics, including solubility and synthesizability. Our research opens up new possibilities for enhancing drug optimization and inspires future investigations

into addressing challenges within the realm of drug optimization. This includes exploring areas like the integration of graph information, a facet that our current work does not tackle.

# CHAPTER 5

# SCAFFOLD-BASED DRUG OPTIMIZATION USING GPT AND MULTI-OBJECTIVE REINFORCEMENT LEARNING

The quest for drug improvement is increasingly critical yet remains underexplored. The DrugImprover framework, as detailed in Section 4, pioneers in defining the drug optimization challenge through the lens of Tanimoto similarity [61]. More specifically, the DrugImprover framework contributes to the drug optmization domain in two key aspects: 1) a detailed workflow for drug optimization, 2) it introduced an Advantage-alignment Optimization (APO) reinforcement learning (RL) algorithm to enhance the multi-objective generative model for drug optmization. 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. This limitation is notable, especially when considering drug optimization as analogous to machine translation in NLP, where a text is translated from one language to another. For drug optimization, an input starting molecule is translated into a target molecule with optimized properties based on the SMILES representation.

Despite its promising outcomes, DrugImprover's reliance on the simpler LSTM network architecture curtails its scalability and depth in contextual understanding. This limitation is notable, especially when considering drug optimization as analogous to machine translation in NLP, where the goal is to 'translate' a molecule to a version with optimized properties using the SMILES representation. Transformers, known for their SOTA performance in machine translation, outperform LSTM-based models [51], suggesting a potential avenue for enhancing drug optimization models.

Moreover, the initial state selection for DrugImprover's LSTM generator is random, leading to the generation of molecules significantly different from the original targets. This highlights the need for a minimum similarity threshold to maintain the essence of the original drugs while optimizing their properties.

Recent attempts to utilize LLMs in drug discovery [36, 27, 57], despite making initial strides, have not yet achieved performance comparable to their successes in other domains. The field eagerly awaits breakthroughs akin to those LLMs have achieved in natural language understanding [122, 76], text-to-video conversion [64], and code generation [131, 72].

Several challenges have impeded the influence of LLMs on drug design: (1) The first challenge is that generated molecules must meet multiple criteria, including solubility and synthesizability, while also achieving a high docking score when targeting a specific site. Current drug discovery LLMs merely perform pretraining on molecules without focusing on improving multiple properties [35, 36, 57]. (2) Secondly, considering that drugs with akin chemical structures are expected to produce similar biological or chemical effects [8], it is vital to balance the optimization of the drug while retaining the positive attributes of the original compound. (3) Lastly, the approach of current drug discovery LLMs predominantly centers on optimizing for maximum likelihood during the decoding phase, without tailoring the optimization to specific objectives.


Figure 5.1: Scaffold is a transformer-based language model that maps scaffold to molecule

To address these challenge, in this work we propose ScaffoldGPT, a novel large language model-based, three-stage workflow for drug improvement. This workflow is designed to enhance existing drugs to rapidly evolving virus variants and cancer cells, addressing the previously mentioned limitations of drug discovery and repurposing, as well as overcoming the limitation of earlier drug improvement efforts. The workflow of ScaffoldGPT comprises

three parts: 1) A novel pretraining framework for scaffold-based large language models. This initial step involves pretraining a drug improvement Large Language Model by learning the inference between a complete molecule and its scaffold. By initiating generation from a scaffold, we ensure the generated molecules will always share a same substructure as the original ones, thus establishing a minimum similarity threshold 2) RL alignment with the desired objective. This part involves fine-tuning the drug improvement large language model with Reinforcement Learning towards multiple objective optimization targets, using Tanimoto Similarity as a critic to preserve the beneficial properties of the original drug and adopting various criteria as critics. 3) Optimized controllable decoding. The final step involves refining the LLM decoder to address the challenge of generating controllable outcomes aligned with the targeted objectives.

In summary, our contributions are: (1) A framework for drug improvement using LLMs that involves a three-step optimization process. (2) A decoding strategy that enables controlled, reward-guided generation using pretrained LLMs. (3) Through extensive testing on real-world viral and cancer-related proteins, we demonstrate that ScaffoldGPT reliably improves upon existing molecules/drugs in terms of all targeted objectives, resulting in superior drug candidates.

## 5.1 Preliminaries

In the following sections, we detail MDP, LLM 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) [81] $\mathcal{M}_0 = \langle \mathcal{S}, \mathcal{A}, T, \mathcal{P}, R \rangle$. 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:

$$Q^\pi(s, a) := \mathbb{E}^\pi \left[ \sum_{t=0}^{T} R(s_t, a_t) \mid s_0 = s, a_0 = a \right], \tag{5.1}$$

where the expectation is based on the trajectory determined by the policy $\pi$. The associated value function is given by:

$$V^\pi(s) := \mathbb{E}_{a \sim \pi(\cdot|s)} \left[ Q^\pi(s, a) \right]. \tag{5.2}$$

**LLM.** 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 [119] representation string, and a termination action [EOS]. We define the set of complete molecules as

$$\mathcal{Y}_T := \{ [\text{BOS}] \circ \mathbf{v} \circ [\text{EOS}] \mid \mathbf{v} \in \mathcal{V}^* \}, \tag{5.3}$$

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 LLM policy $\pi_\theta$ to acquire prior knowledge for generating valid molecules based on a given set of molecules $\mathcal{B}$. We define the generator policy $\pi_\theta$, with learned weights $\theta$, 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 $\mathbf{y}_{<1} = y_0 := [\text{BOS}]$. 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 $\pi_\theta$, the goal in drug optimization is to find the optimal policy $\pi_{\theta^*}$ that maximize the following objective:

$$\pi_{\theta^*} = \arg\max_{\pi_\theta} \mathbb{E}_{X \sim d_0}[R(Y) - R(X) \,|\theta, X], \tag{5.4}$$

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 Large Language Models. The current state of the art, REINVENT, employs the Transformer architecture; however, it 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 DrugImproverLLM.

## 5.2 The DrugImproverLLM Algorithm

In this section, we present DrugImproverLLM, a GPT model focused on scaffold-based molecule optimization. Initially, we detail the design of ScaffoldGPT, which serves as the basis for DrugImproverLLM. We then elaborate on the APO and decoding algorithms that follow ScaffoldGPT, forming a three-stage drug optimization strategy encompassing pretraining, fine-tuning, and decoding optimization.

### *5.2.1 ScaffoldGPT: Drug Optimization by Preserving Chemical Scaffolds*

**Stage 1. Pretrain a LLM-based generator.** Let us note the LLM-based generator policy as $\pi_\theta$, which computes the probability $p$ of the occurrence of the $t^{th}$ token in a target

**Algorithm 2** DrugImproverLLM

**Require:** LLM-based generator policy $\pi_\theta^G$; critics $\mathbf{C}$.

Initialize $\pi_\theta^G$ with GPT2-like Transformer with random weight $\theta$.

▷ /* Stage 1: Pretrain LLM-based generator */

Build the training corpus $C_{\text{Phase 1}}$ (5.6), $C_{\text{Phase 2}}$ (5.7).

Pre-train BPE tokenizer and $\pi_\theta^G$ on $C_{\text{Phase 1}}$ via CLM objective (5.5).

Pre-train $\pi_\theta^G$ 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 $Y_{1:T} = (y_t, \ldots, y_T) \sim \pi_\theta^G(\cdot|S)$.

    Compute advantage preference $R^{\text{AP}}$ by (5.12)(5.13).

    Update generator $\theta$ via policy gradient by (5.14).

▷ /* Stage 3: Token-level Decoding Optimization */

Optimize the generation of $\pi_\theta^G$ via Top-N (5.15) decoding strategy.

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 $\theta$ of the generator policy $\pi_\theta$ 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:

$$NLL = -\log P(Y|S) = -\sum_{t=1}^{T} \log P(y_t|y_{t-1}, ..., y_1, S) = -\sum_{t=1}^{T} \log \pi_\theta(y_t|y_{1:t-1}, S), \quad (5.5)$$

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) [119] 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 [28, 95]. Building on the pre-trained BPE tokenizer, we propose a two-phase incremental training approach, as illustrated in Fig. 5.2, 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 large language model (LLM) 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:

$$C_{\text{Phase 1}} = \left\{ [BOS], \quad \underbrace{y_1, \cdots, y_T}_{\text{target molecule Y}} \quad , [EOS] \right\}, \tag{5.6}$$

In the second phase, building upon the success of the LLM 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 is designed as follows:

$$C_{\text{Phase 2}} = \left\{ [BOS], \langle S \rangle, \quad \underbrace{s_1, \cdots, s_T}_{\text{source scaffold S}} \quad , \langle L \rangle, \quad \underbrace{y_1, \cdots, y_T}_{\text{target molecule Y}} \quad , [EOS] \right\}, \tag{5.7}$$

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 LLM-based generator policy, $\pi_\theta$, to target specific outcomes by applying reinforcement learning finetuning in the next stage.

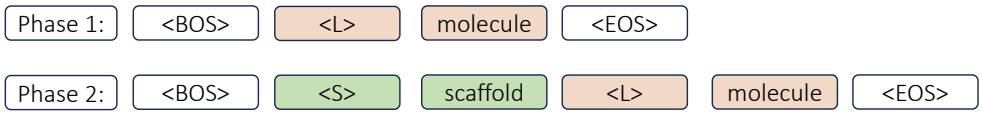



Figure 5.2: Two-phase incremental training of DrugImproverLLM.

### *5.2.2 Policy Improvement via Advantage-alignment Policy Optimization*

**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) [61] algorithm to fine-tune the pretrained LLM-based generator policy $\pi_\theta$. 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. [63], and regarded each pharmaceutical property as a critic $C$ and got an ensemble critics as follows:

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

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:

$$R^{\text{norm}}(Y) := R^{\text{norm}}(Y|S) = \lambda \cdot \text{Norm}\left(C^{\text{Tanimoto}}(S, Y)\right) + \sum_{i=0}^{|\mathbf{C}|-1} \lambda \cdot \text{Norm}\left(C_i(Y)\right), \quad (5.8)$$

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.

In this work, we use *BON* [29] (Best of N) search to estimate the reward for a prompt

(partial molecule). $BON$ can be formulated as the following:

$$BON\left(\mathbf{y}_{<i}, N, R\right)|_{S,p,k} = \max_{\mathbf{Y}_j \in \{\mathbf{Y}_1, \cdots, \mathbf{Y}_N\}} R\left(\mathbf{Y}_j\right), \tag{5.9}$$

$$\text{where } Y_j = \left[\mathbf{y}_{<i}, y_i, \cdots, y_T\right]_j, \text{and } y_i \sim \textsc{Top-PK}\left(\mathbf{y}_{<i}, p, k\right)|_S. \tag{5.10}$$

where Top-PK [63] defined as follows:

$$\textsc{Top-PK}\left(\mathbf{y}_{<i}, p, k\right)|_S = \mathcal{A}_{\mathbf{y}_{<i}}, \text{ where } \mathcal{A}_{\mathbf{y}_{<i}} = \left\{y_1, \ldots, y_j\right\}, y_i \in \mathcal{V}, \tag{5.11}$$

$$j = \min\left\{\arg\min_{j'} \sum_{i=1}^{j'} \pi_\theta\left(y_i|S, \mathbf{y}_{<i}\right) \geq p, k\right\}, \text{and } \pi_\theta\left(y_g|S, \mathbf{y}_{<i}\right) > \pi_\theta\left(y_h|S, \mathbf{y}_{<i}\right), \text{if } g < h,$$

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 (5.11) and $BON$ (5.9) method to select the best one (the one with the highest normalized reward) to serve as the foundation for the final reward calculation.

$$R_c\left(Y|S\right) = R^{\text{norm}}(Y)|_S, \quad Y \in BON\left(y_0, N, R\right)|_{S,p,k}. \tag{5.12}$$

APO makes policy gradient based on the advantage preference [61], which is defined as

$$R^{\text{AP}}\left(Y_{1:T}, S\right) = R_c\left(Y_{1:T}\right) - R_c\left(S\right), \tag{5.13}$$

and perform APO policy gradient with follows:

$$g = \mathbb{E}_{S \sim \rho_0, Y_{1:T} \sim \pi_\theta^G(\cdot|S)}\left[\nabla_\theta \log \pi_\theta^G\left(Y_{1:T}|S\right) \cdot R^{\texttt{AP}}\left(S, Y_{1:T}\right)\right], \tag{5.14}$$

### 5.2.3 Multi-objective Reward-Driven Token-level Generation Strategy

**Stage 3. Token-level Controllable decoding generation.** Ultimately, the current LLM-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:

$$
\begin{aligned}
&Y^{\star} \sim \{[\mathbf{y}_{<i}, y_i, \cdots, y_T]\}, \text{ where } y_i \sim \textsc{Top-N}\left(\mathbf{y}_{<i}, p, k, n\right)|_S \\
&\textsc{Top-N}\left(\mathbf{y}_{<i}, p, k, n\right)|_S = \mathcal{A}_{\mathbf{y}_{<i}}, \text{ where } \mathcal{A}_{\mathbf{y}_{<i}} = \{y_1, \ldots, y_n\}, y_i \in \mathcal{V}, |\mathcal{A}| \leq k, \\
&\text{and } R\left(BON\left(\mathbf{y}_{<i} \circ y_g, N, R\right)|_{S,p,k}\right) > R\left(BON\left(\mathbf{y}_{<i} \circ y_h, N, R\right)|_{S,p,k}\right), \forall g < h,
\end{aligned}
\tag{5.15}
$$

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

*Remark* 5.2.1. Top-N differs from Top-P, Top-K, and Top-K 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.

## 5.3 Experiments

### 5.3.1 Experiment Setup

**The language model.** We utilize 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 DrugImproverLLM (See appendix for more details).

**Baselines.** In this work, we employ baseline models including DrugImprover [61], which utilizes an LSTM-based generator with APO fine-tuning. Additionally, we incorporate the model proposed by He et al. [35, 36], which trains a transformer to adhere to the Matched Molecular Pair (MMP) guidelines [50, 109]. 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 desirable property changes between molecule $X$ and $Y$.
- Similarity $\geq 0.5$: the 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 $\geq 0.7$: the 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.

**Dataset.** We employ, from the most recent Cancer and COVID dataset of Liu et al. [61], 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 DrugImprover-LLM in generating molecules with desirable attributes within the context of pharmaceutical drug discovery. We leverage the RDKit [52] chemoinformatics package and employ various performance metrics as follows: **Validity**: It measures if the generated SMILES is valid in syntax. **Druglikeness:** The druglikeness metric measures the likelihood of a molecule being a suitable candidate for drug development. **Solubility:** This metric assesses the likelihood of a molecule's ability to mix with water, commonly referred to as the water-octanol partition coefficient (LogP). **Synthetizability:** This parameter quantifies the ease (score of 1) or difficulty (score of 10) associated with synthesizing a given molecule [22]. **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 C.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:** It is the average of the normalized reward of the top 10% of molecules based on their average normalized reward. **Average Norm Reward:** It 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.

### *5.3.2 Experimental Results*

Table 5.1 shows that DrugImproverLLM surpasses DrugImprover and six different versions of REINVENT in performance measures for both virus-related and cancer-related proteins. Moreover, DrugImproverLLM 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

| 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 [36]** | 0.995 ± 0.001 | 0.629 ± 0.001 | 0.717 ± 0.001 | -8.241 ± 0.015 | 0.687 ± 0.003 | 2.683 ± 0.005 | 3.144 ± 0.028 | 0.870 ± 0.003 |
| 7BQY) | **Similarity (≥ 0.5) [36]** | 0.995 ± 0.001 | 0.617 ± 0.001 | 0.706 ± 0.001 | -8.222 ± 0.022 | 0.690 ± 0.003 | 2.664 ± 0.005 | 3.162 ± 0.014 | 0.803 ± 0.002 |
| | **Similarity ([0.5, 0.7)]) [36]** | 0.995 ± 0.001 | 0.611 ± 0.001 | 0.699 ± 0.001 | -8.195 ± 0.027 | 0.688 ± 0.002 | 2.660 ± 0.009 | 3.196 ± 0.022 | 0.775 ± 0.003 |
| | **Similarity (≥ 0.7) [36]** | 0.995 ± 0.001 | 0.630 ± 0.001 | 0.717 ± 0.001 | -8.218 ± 0.007 | 0.694 ± 0.001 | 2.719 ± 0.006 | 3.058 ± 0.021 | **0.890** ± 0.003 |
| | **Scaffold [36]** | 0.995 ± 0.001 | 0.607 ± 0.001 | 0.704 ± 0.002 | -8.113 ± 0.015 | 0.700 ± 0.002 | 2.702 ± 0.006 | 2.961 ± 0.014 | 0.789 ± 0.002 |
| | **Scaffold Generic [36]** | 0.994 ± 0.001 | 0.617 ± 0.001 | 0.710 ± 0.002 | -8.185 ± 0.017 | 0.698 ± 0.002 | 2.663 ± 0.007 | 3.070 ± 0.020 | 0.808 ± 0.002 |
| | **DrugImprover [61]** | 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 | 0.087 ± 0.002 |
| | **DrugImproverLLM (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 |
| | **DrugImproverLLM (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 |
| | **DrugImproverLLM (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 |
| | **DrugImproverLLM** | 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 [36]** | 0.998 ± 0.001 | 0.636 ± 0.001 | 0.731 ± 0.002 | -8.422 ± 0.022 | 0.712 ± 0.03 | 2.601 ± 0.003 | 2.987 ± 0.025 | 0.851 ± 0.002 |
| 4DWQ) | **Similarity (≥ 0.5) [36]** | 0.999 ± 0.001 | 0.626 ± 0.001 | 0.723 ± 0.001 | -8.452 ± 0.037 | 0.712 ± 0.003 | 2.579 ± 0.006 | 3.013 ± 0.018 | 0.785 ± 0.003 |
| | **Similarity ([0.5, 0.7)]) [36]** | 0.999 ± 0.001 | 0.622 ± 0.002 | 0.718 ± 0.001 | -8.428 ± 0.016 | 0.709 ± 0.002 | 2.558 ± 0.006 | 3.079 ± 0.029 | 0.757 ± 0.003 |
| | **Similarity (≥ 0.7) [36]** | 0.999 ± 0.001 | 0.640 ± 0.001 | 0.733 ± 0.002 | -8.445 ± 0.023 | 0.718 ± 0.002 | 2.629 ± 0.004 | 2.880 ± 0.012 | **0.880** ± 0.003 |
| | **Scaffold [36]** | 0.998 ± 0.001 | 0.615 ± 0.003 | 0.720 ± 0.002 | -8.512 ± 0.038 | 0.719 ± 0.001 | 2.587 ± 0.005 | 2.764 ± 0.014 | 0.748 ± 0.002 |
| | **Scaffold Generic [36]** | 0.998 ± 0.001 | 0.624 ± 0.001 | 0.723 ± 0.001 | -8.497 ± 0.023 | 0.722 ± 0.002 | 2.562 ± 0.006 | 2.877 ± 0.019 | 0.771 ± 0.002 |
| | **DrugImprover [61]** | 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 | 0.077 ± 0.001 |
| | **DrugImproverLLM (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 |
| | **DrugImproverLLM (w/o Top-N)** | 0.611 ± 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 |
| | **DrugImproverLLM (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 |
| | **DrugImproverLLM** | 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 5.1: **Main results.** A comparison of eight baselines including Original, six baselines from REINVENT {MMP, Similarity ($\geq 0.5$), Similarity $\in [0.5, 0.7)$, Similarity $\geq 0.7$, Scaffold, Scaffold Generic}, DrugImprover and different versions of DrugImproverLLM on multiple objectives based on 3CLPro and RTCB datasets. The top two results are highlighted as **1st** and 2nd. Results are reported for 5 experimental runs.

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, DrugImproverLLM outshines DrugImprover in all benchmarks. This is because DrugImproverLLM 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 DrugImproverLLM enhanced scalability, capacity, and contextual understanding compared to DrugImprover.

In contrast to the current state-of-the-art approach, REINVENT, which pre-trains a Transformer with constraints on Tanimoto similarity, their method falls short in achieving drug improvement as it overlooks the optimization of multiple pharmaceutical properties. Therefore, Table 5.1 reveals that although REINVENT achieved high similarity, the generated

molecules often failed to surpass the original ones. DrugImproverLLM, 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 DrugImprover-LLM to successfully enhance the original drug across various pharmaceutical properties while maintaining a high Tanimoto similarity.

### *5.3.3 Ablation Studies*

In this section, we present ablation studies that underscore the necessity and effectiveness of each component of DrugImproverLLM. These components complement each other, substantially enhancing overall performance.

**Effectiveness of APO Finetuning.** DrugImproverLLM adopts APO finetuning as the second step, following the completion of pretraining the LLM-based generator. Table 5.1 demonstrates the effectiveness of APO through two comparisons: DrugImproverLLM (w/o APO, Top-N) vs. DrugImproverLLM (w/o Top-N), which shows that after applying APO finetuning, performance improved on most properties. Additionally, DrugImproverLLM vs. DrugImproverLLM (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.

**Effectiveness of Top-N decoding strategy.** DrugImproverLLM adopts the Top-N decoding strategy as the final step followed by APO finetuning. Table 5.1 demonstrates the effectiveness of Top-N through two comparisons: DrugImproverLLM (w/o APO, Top-N) vs. DrugImproverLLM (w/o APO), showing that after applying the Top-N decoding strategy on top of pretrained LLM, performance improved across most properties. Moreover, DrugImproverLLM vs. DrugImproverLLM (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 DrugImproverLLM (w/o APO) and DrugImproverLLM (w/o Top-N), we observe that applying Top-N decoding alone enhances performance more than applying APO alone.

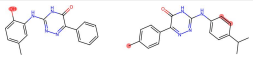
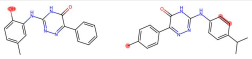
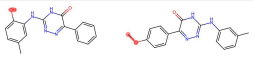
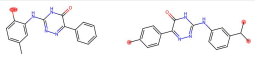
| | **Original** | **improved 1** | **improved 2** | **improved 3** |
|---|---|---|---|---|
| **Molecule** |  |  |  |  |
| **SMILE String** | Cc1ccc(O)c(Nc2nnc (-c3ccccc3)c(=O)[nH]2)c1 | Cc1ccc(-c2nnc(Nc3ccc (C(C)C)cc3)[nH]c2=O)cc1 | COc1ccc(-c2nnc(Nc3cccc (C)c3)[nH]c2=O)cc1 | Cc1ccc(-c2nnc(Nc3cccc (C(C)C)c3)[nH]c2=O)cc1 |
| **Scaffold** | O=c1[nH]c(Nc2ccccc2) nnc1-c1ccccc1 | - | - | - |
| **Docking** (↓) | -10.031 | -11.478 | -11.474 | -11.087 |
| **Druglikeness** (↑) | 0.646 | 0.762 | 0.774 | 0.762 |
| **Synthesizability** (↓) | 2.390 | 2.298 | 2.257 | 2.356 |
| **Solubility** (↑) | 2.590 | 4.007 | 2.893 | 4.007 |
| **Similarity** (↑) | - | 0.826 | 0.911 | 0.866 |
| **Avg Norm Reward** (↑) | 0.618 | 0.759 | 0.753 | 0.754 |

Table 5.2: One optimization example from cancer benchmark. The scaffold is slightly different from original, and every generated molecules contains scaffold.

**Drug optimization illustration.** Finally, we provide three examples illustrating the effectiveness of DrugImproverLLM in improving upon the original molecule on the cancer benchmark, as shown in Table 5.2 (Refer to Appendix C.5 for the COVID benchmark). The results in Table 5.2 demonstrate that the drugs generated by DrugImproverLLM outperform the original drugs across all desired properties. Additionally, the comparison figure in Table 5.2 illustrates that the improved molecules preserve the original drug to a significant extent, with only minor changes highlighted in red. The results indicate that DrugImproverLLM effectively optimizes desired properties while preserving the beneficial properties of the original drug.

## 5.4 Conclusion

In this study, we introduce DrugImproverLLM, a novel framework named designed for drug optimization. It incorporates a unique scaffold-based LLM design, a three-step

optimization process, a two-phase incremental training method, and a novel Top-N decoding strategy, facilitating controlled reward-guided generation using pretrained LLMs. To showcase the superior performance of DrugImproverLLM, we conduct evaluations on real-world viral and cancer-related datasets, comparing it against eight competing baselines, including the current state-of-the-art approach. Our results demonstrate that DrugImproverLLM surpasses all baselines, including the state-of-the-art, across the majority of performance metrics, underscoring its efficacy. Our work highlights DrugImproverLLM's effectiveness in drug optimization, as evidenced by enhancements in various pharmaceutical properties. As a future direction, we encourage the utilization of DrugImproverLLM in domains beyond the scope of our current research.

# CHAPTER 6

# CAUSAL MASKED SEQ2SEQ BIDIRECTIONAL GPT FOR GUIDED DRUG OPTIMIZATION

The search for enhancing drugs is becoming more urgent but is still not fully explored. The DrugImprover framework, introduced in Section 4, leads the way in tackling the challenge of drug optimization using Tanimoto similarity. Furthermore, DrugImproverLLM, discussed in section 5, enhances outcomes by utilizing the advanced generative capabilities of transformers and LLMs. Specifically, DrugImproverLLM aids the field of drug optimization in three main areas: (1) It introduces a new pretraining approach that focuses on a scaffold-based LLM, aimed at understanding the connection between a molecule and its scaffold, setting the stage for further optimization efforts. (2) It applies RL to fine-tune the model towards achieving multiple objective optimization goals, using Tanimoto Similarity as a critic to preserve the beneficial properties of the original drug and adopting various criteria as critics. (3) It utilizes a decoding strategy that enables controlled, reward-guided generation using pretrained LLMs

Although DrugImproverLLM has further improved the performance and demonstrated promising outcomes, the effectiveness is still limited due to the 'black box' nature of LLMs generation process, where LLMs optimize for either maximum likelihood or specific objectives set by users during the decoding phase. More specifically, users cannot specify the generation locations or the quantity of tokens to be generated. Such limitation is further intensified in drug optimization and DrugImproverLLM: (1) If an important substructure exists within the original molecule, the scaffold-based DrugImproverLLM might miss it and not include it in the generated ones, even though retaining that substructure could be beneficial. (2) If the scaffold has associated side effects, drugs derived from it might inherit these issues, which contradicts the goal of optimization. (3) Examples of drug improvement in the real world have shown that minor modifications to the original drugs, such as adding a simple NH2 functional group, can significantly enhance their effectiveness and reduce side effects, as

shown with Ampicillin improving upon Penicillin, illustrated in figure 6.1.

(a) Penicillin (b) Ampicillin

Figure 6.1: Penicillin in drug optimization. With adding a simple functional group NH2 (in red), Ampicillin has resolved the rash side effect bring about by Penicillin.

To overcome these challenges, this work introduces DrugImproverCMS, a Causal Masked Seq2Seq (CMS) Bidirectional GPT designed for drug optimization. This model's training objective mimics biological growth and evolution in SMILES sequences, allowing for modifications at any point in the sequence. DrugImproverCMS consists of two main components: (1) It presents a new type of causally masked generative models trained on a vast collection of molecular SMILES. These models generate tokens sequentially but can also mask and later generate long token spans at different positions, offering a blend of causal and masked language modeling for comprehensive generative modeling with bidirectional context. This feature enables targeted modifications without altering the molecule's overall structure. (2) It incorporates size hints, enabling users to specify the number of new tokens to be generated at a particular position in the original molecule, guiding the model during sample generation.

In summary, our contributions are: (1) We present the causally masked objective, merging causal and masked language modeling for bidirectional control over SMILES sequence generation. (2) Through extensive testing on real-world viral and cancer-related proteins, we demonstrate the DrugImproverCMS has potential to improve upon existing molecules/drugs in terms of all targeted objectives, resulting in superior drug candidates

## 6.1 Preliminaries

**LLM.** Let $\mathbf{X} = [x_1, x_2, \cdots, x_n]$ be a sequence of tokens representing an input sentence (prompt), where each $x_i$ is a token from a vocabulary $\mathcal{Y}$. Let $\mathbf{Y} = [y_1, y_2, \cdots, y_T], y_i \in \mathcal{Y}$ be the output sequence of tokens with vocabulary $\mathcal{Y}$. $\mathcal{Y}$ and $\mathcal{Y}$ are potentially different

vocabularies. Note that $\mathbf{y}_{<t} = [y_1, \cdots, y_{t-1}], \mathbf{y}_T := Y$. $T$ represents the length of sequence. Each training corpus begins with a start token [BOS], follows with a sequence of tokens $\mathbf{y}$ where each $y_i$ belongs to $\mathcal{Y}$, and concludes with a termination action [EOS]. Each molecule is depicted using a sequence of tokens $\mathbf{y}$ to assemble a SMILES string, applicable to both incomplete and complete molecular structures. Let us denote $\circ$ as string concatenation, and let $\mathcal{V}^*$ represent the Kleene closure of $\mathcal{V}$. The set of training corpus $C$ is defined as: $C := \{[\text{BOS}] \circ \mathbf{v} \circ [\text{EOS}] \mid \mathbf{v} \in \mathcal{V}^*\}$. The LLM generator policy $\pi_\theta$, which is parameterized by a deep neural network (DNN) with learned weights $\theta$, 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 $y_t$. The text generation decoding process is designed to select the most probable hypothesis from all possible candidates by addressing the following optimization problem: $\mathbf{y}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}_T} \log \pi_\theta(\mathbf{y}|\mathbf{x})$.

**CLM.** CLM is a variant of language modeling where the model is trained to estimate the probability of $x_i$ conditioned on the preceding tokens $\mathbf{X}_{<i}$, where $\mathbf{X}_{<i} = x_1, x_2, \cdots, x_{i-1}$, in typically an autoregressively manner. The objective of CLM is to maximize the log likelihood of observing the correct next token $x_i$ given all the previous tokens in the sequence $\mathbf{X}_{<i}$, which could be formulated as $\max_\theta \sum_{i=1}^{n} \log P(x_i|\mathbf{X}_{<i}; \theta)$, where $P(x_i|\mathbf{X}_{<i}; \theta)$ is the conditional probability of observing token $x_i$ given all the preceding tokens $\mathbf{X}_{<i}$. Causal Language Modeling is particularly powerful for generating text, as it conditions on all previous tokens, ensuring that each generated word is based on the full history of the text generated so far.

**MLM.** In MLM, a subset (around 15%) of the tokens in $\mathbf{X}$ is randomly selected and replaced with a special token [MASK]. Let us denote this masked sequence as $\mathbf{M}$ and unmasked sequence as $\mathbf{S}$, $\mathbf{S} = \{x_i\}, x_i \in \mathbf{X}$ and $x_i \notin \mathbf{M}$. The objective of the MLM is to predict the original tokens of the masked positions based solely on the unmasked context $\mathbf{S}$, which can be represented as maximizing the likelihood: $\mathcal{L}_{MLM} = \prod_{i \in \mathbf{M}} P(x_i|\mathbf{S}; \theta)$, where

$P(x_i|\mathbf{S};\theta)$ represents the conditional probability of observing token $x_i$ given the context provided by the unmasked tokens in $\mathbf{S}$. $\theta$ represents the parameters of the model. The parameters $\theta$ of the model are optimized to maximize the likelihood of the correct tokens at the masked positions. During training, the model learns to utilize the surrounding context to predict the masked tokens, which helps it develop a deep understanding of language structure and usage. MLM has proven effective for pre-training language models that are later fine-tuned for various downstream tasks.

**Seq2Seq.** Sequence-to-sequence (seq2seq) modeling is a framework in natural language processing designed to convert sequences from input sequence to output sequence. Seq2seq models typically consist of two main components: an encoder and a decoder, with model parameter $\theta_{enc}$ and $\theta_{dec}$ respectively. The encoder processes the input sequence $\mathbf{X}$ to a fixed-dimensional vector representation $\mathbf{c}$ to capture the semantic or contextual information. The decoder's objective is to generate the target sequence $\mathbf{Y}$ given the encoded representation $\mathbf{c}$. The objective in training seq2seq models is typically to maximize the log likelihood of the correct output sequence $\mathbf{Y}$ given the input sequence $\mathbf{X}$ across a dataset of paired sequences: $\max_{\theta_{enc},\theta_{dec}} \sum_{(\mathbf{X},\mathbf{Y})} \log P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{P}$ is product of the conditional probabilities of each output token and $P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{n} P\left(y_j|\mathbf{Y}_{<j}, \mathbf{c}; \theta_{dec}\right)$. Training involves adjusting both the encoder and decoder parameters to optimize this objective. Seq2seq models are powerful because they can handle variable-length input and output sequences and are capable of learning complex transformations between different types of sequence data.

**Limitation.** The existing models - CLM, MLM, and seq2seq - have limitations in controllable generation, which is especially important for drug optimization tasks that need to preserve specific structures and allow expansion, shrinking, or mutation at specific positions. The current state of the art in drug optimization, REINVENT, although it incorporates various similarity metrics in building the training corpus for pre-training the transformer

model, still does not yield ideal results due to a lack of controllability in generation. The beneficial structure of the original drug often fails to preserve. In this work, we propose DrugImproverCMS, which effectively addresses the above limitations of current GPT models in controllable drug optimization.

## 6.2 The DrugImproverCMS Algorithm

In this section, we propose DrugImproverCMS, a ground-up designed GPT model for molecule optimization. We first introduce the novel Causally Masked Seq2seq (CMS) Objective as the foundation of DrugImproverCMS. Then, we discuss the design of GPT, including designing the training corpus, a pre-training strategy, and a generation process.

### *6.2.1 Causally Masked Seq2seq (CMS) Objective*

Masked, causal, and seq2seq language modeling each offer unique benefits and limitations. Masked models encode bi-directional contexts but only decode about 15% of the tokens during training. Causal models, being decoder-only, process every token but are restricted to left-to-right contexts. Seq2seq models are versatile yet often lack bidirectional context and precise generation control. To combine the strengths of MLM, CLM, and seq2seq models and draw inspiration from biological molecule evolution using SMILES representation—which allows for molecular expansion, shrinking, and mutation—we introduce the Causally Masked Seq2seq (CMS) Objective. The CMS objective enables per-token generation, incorporating optional bidirectional and seq2seq functionality for greater adaptability. It allows for precise control over specific positions and spans within sequences, supporting the expansion, contraction, or mutation of segments while maintaining the integrity of designated areas. The construction of the CMS objective involves the following steps:

**Designing the corpus.** Our methodology for developing the CMS objective to a SMILES [119] string of length $\mathcal{L}$ begins with the most basic corpus suitable for the CLM objective as

**Blending the MLM objective.** We then build the MLM objective on top of CLM. It involves a probability $p$ to determine the total number of tokens to mask as $\lfloor \mathcal{L} \cdot p \rfloor$. Let us denote $N \in \mathbb{R}^+$ as the number of span of mask in the source document.

$$[BOS], x_1, \cdots, \underbrace{x_{idx_1}, \cdots, x_{idx_1+\lfloor \mathcal{L} \cdot p \rfloor}}_{<mask\_n:\mathcal{L}\cdot p>}, \cdots, x_T, [EOS], \tag{6.1}$$

For $N = 1$, let us choose a random starting index $idx_1 \sim [0, \mathcal{L} - \lfloor \mathcal{L} \cdot p \rfloor - 1]$, and proceed to mask tokens in range $[idx_1, idx_1 + \lfloor \mathcal{L} \cdot p \rfloor]$. For $N = 2$, we divide $\lfloor \mathcal{L} \cdot p \rfloor$ into two segments, $m1$ and $m2$, ensuring $m1 + m2 = \lfloor \mathcal{L} \cdot p \rfloor$ and that each segment's length is uniformly selected from the range $[1, \lfloor \mathcal{L} \cdot p \rfloor]$. We then identify a starting point $idx_1$ within $[0, \mathcal{L} - \lfloor \mathcal{L} \cdot p \rfloor - 1]$ for the first mask span and a second starting point $idx_2$ from the range $[idx_1 + m1 + 1, \mathcal{L} - m2 - 1]$ for the second mask span, ensuring that the two masked segments are non-overlapping and sequentially ordered in the SMILES string. Following the same strategy for selection and masking of these segments, we could reach for any $N$. For the $n_{th}$ span of mask, we replace the span by the token $<mask\_n : \mathcal{L} \cdot p>$, where $n$ and $\mathcal{L} \cdot p$ represents for the $n_{th}$ masked segment with size hint length $\mathcal{L} \cdot p$, which specify the desired length of text to generate for replacing the mask conditioning on tokens length. Finally, we reposition the masked spans to the end of the SMILES string, maintaining their sequence order as illustrated in Fig. 6.2. In this work, we embed the size hint within the mask token as $<mask\_i : n>$ to avoid the ambiguity seen in prior works Aghajanyan et al. that use $<mask\_i>n$. This format prevents misinterpretation by models, as numerical values in chemical structures can indicate ring closures or chain lengths.

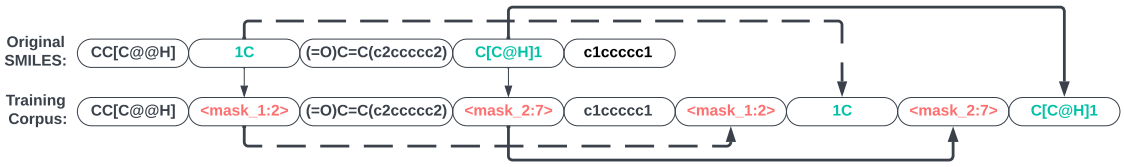



Figure 6.2: The visual representation of our causal masked objective on a molecule features two mask spans ($n = 2$), each with a specific size hint. The first span, $<mask\_1:2>$, covers two tokens, and the second, $<mask\_2:7>$, covers seven tokens.

**Blending the seq2seq objective.** Finally, we establish CMS objective by applying seq2seq objective on top of MLM and CLM. Initially, we train a GPT model using the MLM and CLM objectives, denoted as $\pi_{\text{CM}}$. Given a smile string, we randomly mask a seq2seq span starting at position $s_1$ and of length $\mathcal{L}$, ensuring it does not overlap with previously masked spans, while regard the remaining tokens as $\mathbf{Z}$. Our goal is to transform this s2s span $\left[x_{s_1}, \cdots, x_{s_1+\mathcal{L}}\right]$ into a target span with desired length $\mathcal{L}^t$. To create this training corpus, we utilize $\pi_{\text{CM}}$ to generate $\mathcal{L}^t$ tokens $\left[m_1, \cdots, m_{\mathcal{L}^t}\right]$ with regarded to $\mathbf{Z}$. We then construct the training corpus by mapping the s2s span to the subsequence generated by $\pi_{\text{MLM}}$.

$$\underbrace{x_1, \cdots, x_{s_1-1}, \left\langle \text{mask\_1} : \mathcal{L}^t \right\rangle, x_{s_1+\mathcal{L}+1}, \cdots, x_T, \left\langle \text{mask\_1} : \mathcal{L}^t \right\rangle}_{\text{Prompt based on the pretrained model in previous step}} \rightarrow [m_1, \cdots, m_{\mathcal{L}^t}]$$

$$\underbrace{x_1, \cdots, \left\langle \text{s2s\_}i\text{\_}\mathcal{L}^t : x_{s_1}, \cdots, x_{s_1+\mathcal{L}} \right\rangle, \cdots, x_T, \left\langle \text{s2s\_}i\text{\_}\mathcal{L}^t : x_{s_1}, \cdots, x_{s_1+\mathcal{L}} \right\rangle, [m_1, \cdots, m_{\mathcal{L}^t}]}_{\text{Training corpus for seq2seq objective}}$$

where $\left\langle \text{s2s\_}i\text{\_}\mathcal{L}^t : x_{s_1}, \cdots, x_{s_1+\mathcal{L}} \right\rangle$ denotes the seq2seq objective conditioned on a specific subsequence $x_{s_1}, \cdots, x_{s_1+\mathcal{L}}$ and its bidirectional unmasked tokens. The index $i$ indicates the $i$-th span, and $\mathcal{L}^t$ represents the target length of the generated subsequence. Unlike conventional sequence-to-sequence models, our work on seq2seq is also conditioned on and benefits from the bidirectional context surrounding the seq2seq span. This approach allows for the incorporation of task-specific length priors into prompts, resulting in outputs that are more precise and controlled.

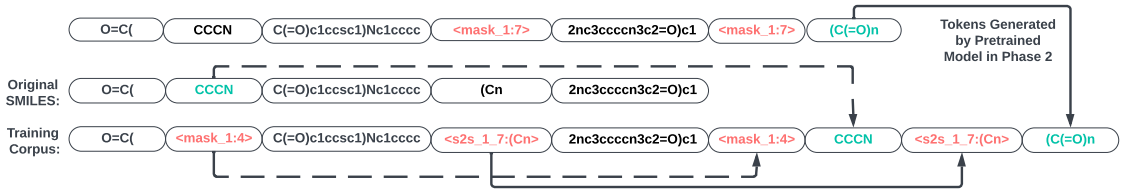



Figure 6.3: The visual representation of building the training corpus with both masked and seq2seq spans for seq2seq causal masked objective.

### *6.2.2 The Design of the DrugImproverCMS*

**Pretraining.** In this work, we propose a novel three-phase training approach to train a GPT model under CMS objective.

*In the initial phase.* Our objective is to train a LLM specifically designed for understanding molecules. This training employs a CLM approach. CLM is an autoregressive technique where the model learns to predict the next token in a sequence based solely on the preceding tokens. This creates a unidirectional context model, which means it only considers past information and ignores any future context when making predictions. For this phase, the model is trained on a dataset comprised of texts about ligands. This dataset enables the model to accurately learn the representation of compounds, including their chemical structures and properties.

*In the second phase.* Building on the success of the LLM developed in Phase 1, which demonstrated high accuracy in generating molecular structures, we proceed to refine the model's training. This phase employs a causally masked objective with multiple mask tokens, each with a size hint, as illustrated in Figure 6.2. In this phase, the model, denoted as $\pi_{CM}$, benefits from both Causal Language Modeling (CLM) and Masked Language Modeling (MLM), which enhance CLM's performance by utilizing bidirectional context. $\pi_{CM}$ is capable of generating molecules in a controlled manner, specifying both the target length and the position for expansion.

*In the third phase.* Ultimately, we achieve building the GPT under CMS objective by further refining the causally masked model,$\pi_{CM}$, through the integration of a sequence-to-sequence objective. We trained our model, denoted as $\pi_{CMS}$, using the training corpus

outlined in Fig. 6.3 to refine the causally masked model $\pi_{CM}$ developed in Phase 2. This advancement aims to enhance the model's controllable generation in terms of both contraction and mutation. It mimics the mutation behavior in biological sequences. Thus, our $\pi_{SCM}$ achieves controllable generation in expansion, contraction, and mutation at specific positions or ranges, in either a random or specified length.

*Loss function.* Instead of altering the standard cross-entropy loss to consider the loss from predicting masked tokens negligible, we treat masked tokens like regular tokens, subject to the usual loss calculations. This method is used because our training data may contain multiple masked tokens, each with size hint information indicating the number of tokens to generate in place of the mask. Thus, it's crucial to accurately predict both the presence of these masked tokens and their corresponding size hints.

**Generation Process.** The prompt, output string, and generated SMILES for DrugImproverCMS can be viewed in figure 6.4 and figure 6.5. More specifically, in the process of generating new molecular structures, DrugImproverCMS employs a method that either modifies existing molecules or adds new elements to them without altering the original essential structure, showcasing the flexibility and precision of the model in generating novel molecular designs. This is illustrated through two examples:

*Modifying the Original Molecule:* Initially, two segments of the original molecule's SMILES string are identified and replaced with mask and seq2seq token respectively, which are placeholders indicating where and how long the new segments should be. These mask tokens are then processed by the model, which generates new segments in their place. The generated segments, highlighted in green, are manually repositioned to replace the original masked segments, effectively changing the molecule's structure and construct a new molecule. This process is depicted in Fig. 6.4, where the mask and seq2seq token are shown in red and the newly generated segments in green. The caption for Fig. 6.4 explains this process in detail, emphasizing the manual reintegration of generated tokens.

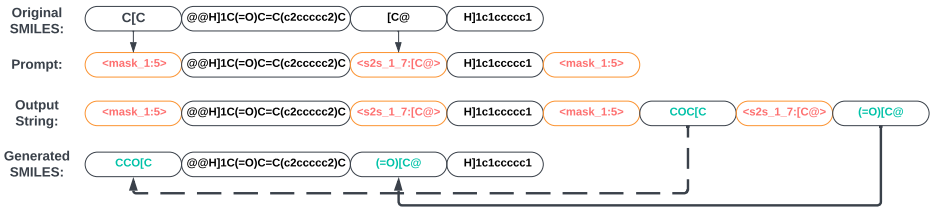



Figure 6.4: Modification of an original molecule. This figure illustrates the process of altering a molecule's structure. Key steps include replacing original segments with masked and sequence-to-sequence tokens (highlighted in red), generating new molecular segments (in green) by the model, and manually reintegrating these segments into the molecule.

*Adding to the Original Molecule Without Modification:* In this scenario, instead of replacing parts of the SMILES string, one mask token and one seq2seq token are inserted at random positions within the string. These tokens serve as prompts for the model to generate new molecular segments that are then manually inserted into the specified positions, expanding the original molecule without altering its existing structure. This approach is visualized in Fig. 6.5, with the mask tokens again represented in green and the generated segments in red. The caption for Fig. 6.5 provides a clear explanation of this additive process.

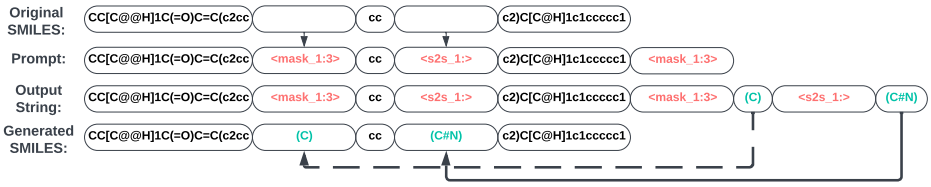



Figure 6.5: Expansion of an original molecule: Mask tokens (in red) are inserted into the SMILES string, prompting the generation of new segments (in green). These segments are then manually added to the molecule, showcasing the model's capability to expand molecular structures both creatively and precisely.

## 6.3 Experiments

### *6.3.1 Experiment Setup*

**The language model.** We employ the Byte Pair Encoding (BPE) method [28, 95] to initially pre-train our tokenizer using raw SMILES strings, and GPT-2-like Transformers

for causal language modeling. We use the standard 11M Drug-like Zinc dataset for training, excluding entries with empty scaffold SMILES. The dataset is divided into a 90/10 split for training and validation, respectively. (For more details, see the appendix D.1).

**Dataset.** We employ, from the most recent Cancer and COVID dataset of Liu et al. [61], 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.

**Baselines.** In this work, we adopt eight baselines including DrugImprover [61], which utilizes an LSTM-based generator with APO fine-tuning. Additionally, we incorporate the current state of art model, REINVENT, proposed by He et al. [35, 36], which trains a transformer to follow the Matched Molecular Pair (MMP) [50, 109] guidelines. Specifically, given a set $\{(X, Y, Z)\}$, where $X$ represents source molecule, $Y$ the target molecule, and $Z$ 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 defined six different kinds of property change $Z$, including MMP for user-specified changes, different similarity thresholds, and scaffold-based alterations, where molecules share the same scaffold or generic scaffold.

**Critics and evaluation metric.** We evaluate seven key attributes for pharmaceutical drug discovery: 1) *Average normalized reward* is the average of the normalized values of the docking score, drug-likeness, synthesizability, solubility, and similarity across all valid molecules. This is regarded as the most crucial metric.; 2) *Average top 10% normalized reward* is the average of the normalized reward of the top 10% of molecules based on their average normalized reward; 3) *Docking score* (generated, for efficient calculation, with a surrogate docking model: see Appendix D.5) evaluates the potential of a drug to inhibit the target site. 4) *Druglikeness* assesses the probability of a molecule being a suitable drug candidate; 5) *Synthesizability* measures the synthesizability of a molecule, assigning a score of 1 for easy synthesis and a score of 10 for difficult synthesis [22]; 6) *Solubility* a molecule's

water-octanol partition coefficient (LogP), indicating how well it can dissolve in water; and 7) *Similarity* evaluates the similarity between original SMILES and generated SMILES using Tonimoto similarity.

### *6.3.2 Experimental Results*

| Target | Algorithm | Avg Norm Reward ↑ | Avg Top 10 % Norm Reward ↑ | Docking ↓ | Druglikeliness ↑ | Synthesizability ↓ | Solubility ↑ | Similarity ↑ |
|---|---|---|---|---|---|---|---|---|
| **3CLPro** | **Original** | 0.532 | 0.689 | -8.698 | 0.682 | 3.920 | 2.471 | - |
| (PDBID: | **MMP [36]** | 0.629 ± 0.001 | 0.717 ± 0.001 | -8.241 ± 0.015 | 0.687 ± 0.003 | 2.683 ± 0.005 | 3.144 ± 0.028 | 0.870 ± 0.003 |
| 7BQY) | **Similarity (≥ 0.5) [36]** | 0.617 ± 0.001 | 0.706 ± 0.001 | -8.222 ± 0.022 | 0.690 ± 0.003 | 2.664 ± 0.005 | 3.162 ± 0.014 | 0.803 ± 0.002 |
| | **Similarity ([0.5, 0.7)]) [36]** | 0.611 ± 0.001 | 0.699 ± 0.001 | -8.195 ± 0.027 | 0.688 ± 0.002 | 2.660 ± 0.009 | 3.196 ± 0.022 | 0.775 ± 0.003 |
| | **Similarity (≥ 0.7) [36]** | 0.630 ± 0.001 | 0.717 ± 0.001 | -8.218 ± 0.007 | 0.694 ± 0.001 | 2.719 ± 0.006 | 3.058 ± 0.021 | 0.890 ± 0.003 |
| | **Scaffold [36]** | 0.607 ± 0.001 | 0.704 ± 0.002 | -8.113 ± 0.015 | 0.700 ± 0.002 | 2.702 ± 0.006 | 2.961 ± 0.014 | 0.789 ± 0.002 |
| | **Scaffold Generic [36]** | 0.617 ± 0.001 | 0.710 ± 0.002 | -8.185 ± 0.017 | 0.698 ± 0.002 | 2.663 ± 0.007 | 3.070 ± 0.020 | 0.808 ± 0.002 |
| | **DrugImprover [61]** | 0.432 ± 0.002 | 0.493 ± 0.005 | -6.726 ± 0.007 | 0.506 ± 0.002 | **1.306** ± 0.010 | 2.057 ± 0.011 | 0.087 ± 0.002 |
| | **DrugImproverCMS (masks only)** | 0.668 ± 0.001 | 0.743 ± 0.001 | -9.083 ± 0.003 | **0.718** ± 0.001 | 2.750 ± 0.001 | 3.630 ± 0.005 | 0.889 ± 0.001 |
| | **DrugImproverCMS (mask + s2s)** | **0.671** ± 0.001 | **0.743** ± 0.001 | **-9.150** ± 0.001 | 0.714 ± 0.001 | 2.763 ± 0.002 | **3.672** ± 0.003 | **0.895** ± 0.001 |
| **RTCB** | **Original** | 0.536 | 0.698 | -8.572 | 0.709 | 3.005 | 2.299 | - |
| (PDBID: | **MMP [36]** | 0.636 ± 0.001 | 0.731 ± 0.002 | -8.422 ± 0.022 | 0.712 ± 0.03 | 2.601 ± 0.003 | 2.987 ± 0.025 | 0.851 ± 0.002 |
| 4DWQ) | **Similarity (≥ 0.5) [36]** | 0.626 ± 0.001 | 0.723 ± 0.001 | -8.452 ± 0.037 | 0.712 ± 0.003 | 2.579 ± 0.006 | 3.013 ± 0.018 | 0.785 ± 0.003 |
| | **Similarity ([0.5, 0.7)]) [36]** | 0.622 ± 0.002 | 0.718 ± 0.001 | -8.428 ± 0.016 | 0.709 ± 0.002 | 2.558 ± 0.006 | 3.079 ± 0.029 | 0.757 ± 0.003 |
| | **Similarity (≥ 0.7) [36]** | 0.640 ± 0.001 | 0.733 ± 0.002 | -8.445 ± 0.023 | 0.718 ± 0.002 | 2.629 ± 0.004 | 2.880 ± 0.012 | 0.880 ± 0.003 |
| | **Scaffold [36]** | 0.615 ± 0.003 | 0.720 ± 0.002 | -8.512 ± 0.038 | 0.719 ± 0.001 | 2.587 ± 0.005 | 2.764 ± 0.014 | 0.748 ± 0.002 |
| | **Scaffold Generic [36]** | 0.624 ± 0.001 | 0.723 ± 0.001 | -8.497 ± 0.023 | 0.722 ± 0.002 | 2.562 ± 0.006 | 2.877 ± 0.019 | 0.771 ± 0.002 |
| | **DrugImprover [61]** | 0.478 ± 0.001 | 0.618 ± 0.002 | -8.701 ± 0.037 | 0.486 ± 0.002 | **1.181** ± 0.010 | 2.026 ± 0.013 | 0.077 ± 0.001 |
| | **DrugImproverCMS (masks only)** | 0.675 ± 0.001 | 0.753 ± 0.001 | -9.318 ± 0.002 | **0.752** ± 0.001 | 2.674 ± 0.001 | 3.292 ± 0.002 | 0.883 ± 0.001 |
| | **DrugImproverCMS (mask + s2s)** | **0.678** ± 0.001 | **0.755** ± 0.001 | **-9.377** ± 0.003 | 0.751 ± 0.001 | 2.688 ± 0.001 | **3.328** ± 0.005 | **0.890** ± 0.001 |

Table 6.1: **Main results.** A comparison of eight baselines including Original, six baselines from REINVENT {MMP, Similarity ($\geq 0.5$), Similarity $\in [0.5, 0.7)$, Similarity $\geq 0.7$, Scaffold, Scaffold Generic}, DrugImprover and DrugImproverCMS on multiple objectives based on 3CLPro and RTCB datasets. The top two results are highlighted as **1st** and 2nd. Results are reported for 5 experimental runs.

Table 6.1 illustrates the performance comparison between DrugImproverCMS and the baseline methods. The results indicate that DrugImproverCMS outperforms the all baselines across all the metrics except for synthesizability. Notably, DrugImproverCMS achieves the highest Tanimoto similarity score, surpassing both the current state-of-the-art, REINVENT, and its six variants. This implies that molecules optimized by DrugImprover-CMS not only exhibit structures more similar to the original drug compared to existing methods but also demonstrate improved properties across various metrics. Additionally, when compared to the original baseline, the drugs generated by DrugImproverCMS significantly enhance the original drug across all desired aspects. These results underscore the superiority

and effectiveness of DrugImproverCMS in controllable optimization of original drugs, preserving beneficial structures while optimizing diverse properties. In addition, DrugImproverCMS with both masked and seq2seq tokens outperforms the masked token only, which demonstrate that the GPT build under our causally masked seq2seq (CMS) objective outperforms the causally masked modeling.

### *6.3.3 Ablation Studies*

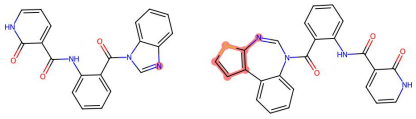
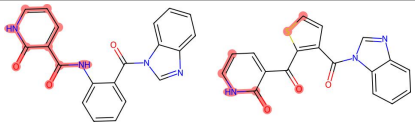
| **Description** | Adding to the Original Molecule Without Modification | Modifying the Original Molecule |
|---|---|---|
| **Molecule** |  |  |
| **Original SMILES** | O=C(Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O | O=C(Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O |
| **Prompt** | O=C(Nc1ccccc1C(=O)n1cnc2<mask_1:7>ccccc21)c1ccc[nH]c1=O<mask_1:7> | O=C(<mask_1:3>ccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O<mask_1:3> |
| **Masked token and length** | None, 0 | Nc1cc, 5 |
| **Generated token and length** | sccc2c2, 7 | c1s, 3 |
| **Generated SMILES** | O=C(Nc1ccccc1C(=O)n1cnc2sccc2c2ccccc21)c1ccc[nH]c1=O | O=C(c1sccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O |

Table 6.2: Examples using masking and size hints for controllable generation.

**Adding to the original molecule without modification.** Table 6.2 (Left) visualizes the addition to the original molecule while preserving the complete original structure. In this experiment, a given original molecule with the SMILES representation O=C(Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O serves as the basis. Our objective is to extend the ring in the molecule. We designed the prompt by adding a mask token $< mask_1 : 7 >$ to the specific position adjacent to the ring in the SMILES. Finally, we obtained the generated molecule with the desired features (additional ring in red) while maintaining the completeness of the original molecule structure. This study demonstrates the ability of DrugImproverCMS to extend at specific positions with a specific length.

**Modifying the Original Molecule.** In this experiment, our goal is to alter a portion of the original molecule by modifying bonds and atoms connecting the two rings. For this purpose, we construct the prompt by substituting the original structure *Nc1cc* with a

masked token $< mask_1 : 3 >$. Table 6.2 (Right) illustrates the modification of the original molecule by removing the ring and introducing a few atoms, while retaining the majority of the structure. This demonstrate the ability of DrugImproverCMS by modifying partial of molecule and random generated in specific length.

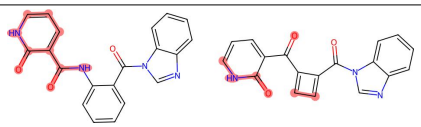
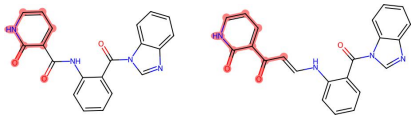
| **Description** | Modifying to the Original Molecule: Simplification | Modifying the Original Molecule: Expansion |
|---|---|---|
| **Molecule** |  |  |
| **Original SMILES** | O=C(Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O | O=C(Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O |
| **Prompt** | O=C(<s2s_1_2:Nc1cc>ccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O<s2s_1_2:Nc1cc> | O=C(<s2s_1_10:Nc1cc>ccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O<s2s_1_10:Nc1cc> |
| **Masked token and length** | Nc1cc, 5 | Nc1cc, 5 |
| **Generated token and length** | c1, 2 | /C=C/Nc1cc, 10 |
| **Generated SMILES** | O=C(c1ccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O | O=C(/C=C/Nc1ccccc1C(=O)n1cnc2ccccc21)c1ccc[nH]c1=O |

Table 6.3: Examples using Seq2Seq and size hints for controllable generation.

**Conditional Modifying to the Original Molecule: contraction and Expansion.** This experiment aims to showcase conditional modifications to the original molecule. Unlike §6.3.3, where the focus is on modifications and expansion in a random manner, here we concentrate on generating subsequences conditioned on a partial molecule. We undertake two tasks: expanding and shrinking partial molecules based on a given subsequence. For the simplification task, we successfully reduce a length 5 subsequence, $Nc1cc$, to a length 2 token using the token $< s2s\_1\_2 : Nc1cc >$. Conversely, for the expansion task, we extend the subsequence to a length of 10 tokens using the token $< s2s\_1\_10 : Nc1cc >$. Both tasks yield the desired molecules, as depicted in Table 6.3. This demonstrates that DrugImproverCMS is capable of generating molecules controllably for contraction and expansion, conditioned on specific segments of the molecule, to target specific lengths of subsequences.

## 6.4 Conclusion

In this study, we introduce the Causally Masked Seq2Seq (CMS) objective and DrugImproverCMS, which is built under CMS objective. DrugImproverCMS is capable of managing precise control over specific areas within sequences, enabling expansion, reduction, or mutation while maintaining the integrity of designated regions. To validate its effectiveness, we conducted extensive experiments on Covid and Cancer drug optimization benchmarks, comparing it against eight baselines, including state-of-the-art methods. DrugImproverCMS not only outperforms these methods across various metrics but also optimizes drugs by preserving the original structure with high Tanimoto similarity and enhancing all desired pharmaceutical properties. We also demonstrate the controllability of DrugImproverCMS through various ablation studies. This work presents a novel objective and training strategy for controlled generation, highlighting DrugImproverCMS's success in drug optimization by improving pharmaceutical properties while retaining beneficial structures. For future directions, we encourage applying DrugImproverCMS in fields beyond our current research scope.

# CHAPTER 7

# CONCLUSION

The quest for developing new drugs is a long, expensive, and high-risk journey, often spanning over a decade and costing billions of dollars. Despite substantial investments in novel drug development, the pharmaceutical industry's output has not increased proportionally due to the inherent inefficiencies and high failure rates in drug discovery. This thesis addresses these challenges by exploring the potential of transformer models in improving the drug discovery process, specifically focusing on me-too drugs, which offer incremental improvements over existing therapies.

In response to the central research question—*"Can transformer models be utilized to improve desired properties of molecules, while maintaining structural similarity to their prototypes?"*—this thesis presents several significant contributions.

Firstly, we designed two fusion methods that integrate the strengths of transformer models and graph neural networks, enhancing the accuracy of binding affinity predictions. Secondly, we simulated a comprehensive dataset encompassing 10 million binding affinity values across 10,000 proteins and 1,000 drugs, providing a robust resource for model training and validation. Thirdly, we proposed two drug optimization generative models, fine-tuned with reinforcement learning, to automate and expedite the development of effective me-too drugs. Additionally, we introduced a bidirectional GPT on molecule textual sequences (SMILES), a hybrid of causal and masked language models, and Seq2Seq that allows for bidirectional context control during generative mask infilling for guided drug optimization. Lastly, we demonstrated that the proposed drug optimization models enhance existing molecules and drugs across all desired objectives through comprehensive experiments on real-world viral and cancer target proteins.

Throughout the chapters, we explored various aspects of integrating transformers into drug discovery. In Chapter 3, by combining transformer and graph representation learning,

we improved the accuracy of binding affinity predictions, an essential step in evaluating potential drug candidates. Chapter 4 presented the initial version of DrugImprover, using transformer-predicted docking scores and LSTM generators with reinforcement learning, showing promising results in optimizing drug properties. In Chapters 5 and 6, we advanced the DrugImprover framework by incorporating GPT-based models and hybrid causal-masked language models, significantly enhancing the ability to generate optimized drug candidates with user-directed features.

The findings of this thesis have several implications for the field of computational drug discovery. The proposed methods have the potential to reduce the time and cost associated with drug discovery by improving the efficiency of me-too drug development. By focusing on enhancing existing drugs, we mitigate some of the risks associated with developing entirely new drugs, offering safer and more effective therapies. This work underscores the versatility and power of transformer models, not only in NLP but also in complex tasks like drug discovery.

While the contributions of this thesis are significant, several limitations must be acknowledged. The models developed and tested in this thesis, while promising, require further validation across a broader range of drug types and biological targets. Despite advances, the black-box nature of AI models remains a challenge, particularly in understanding the underlying mechanisms of drug action.

In conclusion, this thesis demonstrates that transformer models can significantly enhance the drug discovery process, particularly in the development of me-too drugs. By integrating advanced machine learning techniques, we have laid the groundwork for more efficient, cost-effective, and safer drug development methodologies. Future research should focus on further validating these models, improving their interpretability, and exploring their application to a wider range of drug discovery tasks. Through these efforts, we can continue to push the boundaries of what is possible in computational drug discovery, ultimately leading to better

therapeutic options for patients worldwide.

# APPENDIX A

# APPENDIX TO CHAPTER 3

## A.1 Hyperparameters

| Hyperparameter | Value or range |
|---|---|
| *Training* | |
| Learning rate | $10^{-4}$ |
| Optimizer | Adam |
| # of Epochs training | |
| PDBbind | 300 |
| Simulated dataset | |
| Serial Fusion | 20 |
| Adaptive Fusion | 10 |
| *Model* | |
| ESM layers | {6, 12, 30} |
| Hidden dimension | 256 |

Table A.1: Hyperparameters.

Table A.1 provides a list of hyperparameter ranges we used or searched among for our experiments.

# APPENDIX B

# APPENDIX TO CHAPTER 4

## B.1 Molecules and vocabulary

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

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

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

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

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

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

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

(a) 3CLPro. (b) RTCB.

Figure B.1: The binding sites of proteins 3CLPro (PDB ID: 7BQY) (**Left**) and RTCB (PDB ID: 4DWQ) (**Right**). Binding sites are defined around the crystallized compound using Open Eye software.

## B.2 Binding sites of 3clpro and RTCB

## B.3 The sequence generative model

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

**ORGAN.** ORGAN [32] is a generative model designed to optimize sequence distributions. It achieves this by leveraging a combination of domain-specific metrics (objective) and adversarial feedback obtained from a discriminator. The balance between these two components is maintained through a tunable parameter. Within the ORGAN architecture, the generator is constructed as an RNN equipped with LSTM cells. In contrast, the discriminator employs a Convolutional Neural Network (CNN) specifically tailored for text classification tasks. Notably, the Wasserstein distance is chosen as the loss function for the discriminator, ensuring

enhanced stability during training.

**Naive RL.** ORGAN employs a combination of a discriminator and domain objectives as rewards. And by setting the weight of discriminator to be zero, the model ignores the discriminator and becomes a "Naive" RL algorithm [32].

## B.4 Surrogate model

The surrogate model [111] is a simplified version of a BERT-like transformer, which is widely used in natural language processing. In the model, tokenized SMILES strings are inputted and then positionally embedded. Outputs are then passed to a stack of five transformer blocks, each containing a multi-head attention layer (21 heads), dropout layer, layer normalization with residual connection, and feed forward network. The feed forward network consists of two dense layers followed by dropout and layer normalization with residual connection. After the transformer block stack, a final feed forward network is used to output the predicted docking score.

## B.5 Setup

**Setup.** To guarantee an equitable assessment, every algorithm (ORGAN, Naive RL, and APO), is trained using an identical pretrained LSTM-based generator $\pi_\theta^G$. During the training of ORGAN and Naive RL, we adhere to the multi-objective training approach described in [32], which involves alternating between objectives (synthesizability, solubility, docking score and druglikeliness). Specifically, each epoch of ORGAN is dedicated to a different objective, cycling through them for a total of 25 epochs per objective. APO enhances all objectives simultaneously in each epoch.

## B.6 Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of the Polaris supercomputer at the Argonne Leadership Computing Facility where each node contains CPUs (64 cores) and 4 A100 GPU nodes [23]. The training time for each model was approximately 3 hours. We conducted other RL experiments on a cluster that includes CPU nodes (approximately 280 cores) and GPU nodes (approximately 110 Nvidia GPUs, ranging from Titan X to A6000, set up mostly in 4- and 8-GPU configurations). Based on the computing infrastructure, we obtained the wall-time comparison in Table B.1 as follows.

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

Table B.1: Wall-time comparison between different methods.

## B.7 Hyperparameters and architectures

Table B.2 provides a list of hyperparameter settings we used for our experiments.

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

Table B.2: Hyperparameters.

## B.8 Code and data availability

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

Additionally, we release the pretrained model for each protein [69].

# APPENDIX C

# APPENDIX TO CHAPTER 5

## C.1 Pre-training dataset

We used the ZINC dataset, filtering for Standard, In-Stock, and Drug-Like molecules, resulting in approximately 11 million molecules.

## C.2 Generation with finetuned model

The top five epochs with the highest historical average normalized reward (as detailed in Section 5.3.1) 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 5.2.3) for generation.

## C.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.

By iteratively merging the most frequent pairs, BPE builds a robust vocabulary that captures common subword units, allowing for more efficient and flexible text representation.

## C.4 Surrogate model

The surrogate model [111] 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.

## C.5 Drug Optimization illustration on COVID benchmark

This is another example illustrating the effectiveness of DrugImproverLLM in enhancing the original molecule on the COVID benchmark. The results in Table C.1 show that the drugs generated by DrugImproverLLM 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.

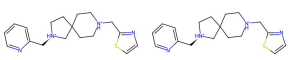
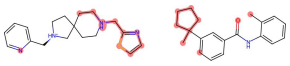
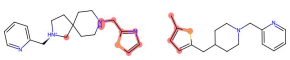
| | Original | improved 1 | improved 2 |
|---|---|---|---|
| **Molecule** |  |  |  |
| **SMILE String** | c1ccc(C[N@H+]2CCC3(CC[NH+](Cc4nccs4)CC3)C2)nc1 | CC1(c2cccc(C(=O)Nc3ccccc3C)c2)CCCC1 | Cc1ccc(CC2CCN(Cc3ccccn3)CC2)s1 |
| **Scaffold** | c1ccc(C[N@H+]2CCC3(CC[NH+](Cc4nccs4)CC3)C2)nc1 | - | - |
| **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 C.1: 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.

## C.6 Binding sites of 3clpro and RTCB

(a) 3CLPro. (b) RTCB.

Figure C.1: The binding sites of proteins 3CLPro (PDB ID: 7BQY) (**Left**) and RTCB (PDB ID: 4DWQ) (**Right**). Binding sites are defined around the crystallized compound using Open Eye software.

## C.7 Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of the supercomputer at the Argonne Leadership Computing Facility where each node contains CPUs (64 cores) and 4 A100 GPU nodes [23]. 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 C.2 as follows.

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

Table C.2: Wall-time comparison between different methods.

## C.8 Hyperparameters and architectures

Table C.3 and B.2 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 [63].

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 |
| Generation | |
| N (Top-N) | 1 |
| K (Number of possible next token) | 16 |
| TopK | 20 |
| TopP | 0.95 |

Table C.3: Hyperparameters for pretraining and generation.

| Parameter | Value |
|---|---|
| Shared | |
| # of Molecules Optimized | 1280 |
| Learning rate | $1 \times 10^{-4}$ |
| Optimizer | Adam |
| # of Epochs for Training | 100 |
| Batch size | 64 |
| Best-of-N | 8 |
| TopK | 20 |
| TopP | 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 | $-R_c(X)$ |
| RTCB | 0 |

Table C.4: Hyperparameters for APO.

# APPENDIX D

# APPENDIX TO CHAPTER 6

## D.1 Pre-training Details

We used the ZINC dataset, filtering for Standard, In-Stock, and Drug-Like molecules, resulting in approximately 11 million molecules.

In the second phase of pre-training, we first trained for 10 epochs using a single mask. Subsequently, we trained for another 40 epochs with an equal probability of using either one or two masks. For each epoch, the masks were regenerated to create a more comprehensive masked dataset.

In the third phase of pre-training, we applied different mask configurations with specific probabilities: [one mask (0.1), two masks (0.1), one mask and one seq2seq (0.4), two masks and one seq2seq (0.4)] and train 20 epochs.Similar to the second phase, the masks were regenerated for each epoch to enhance the comprehensiveness of the masked dataset.

## D.2 Generation

For each mask and seq2seq, we utilize three random variables: the start index, the number of tokens to be masked, and the number of tokens to be generated. During generation, we apply two settings: [one mask + one seq2seq, and two masks], resulting in a total of six random variables for each setting.

During the generation phase, we randomly sample these six variables 10,000 times, using them as prompts for generation, regardless of whether the generated SMILES are valid or not. In addition, for a given prompt molecule, we adopt TOPPK [63] for generation strategy.

After generation, for each prompt molecule/SMILES, we select the top 10 generated molecules/SMILES based on their average normalized reward. The mean of these top 10 molecules/SMILES is then used to obtain the final result for the prompt molecule/SMILES.

## D.3 Baseline REINVENT

Following are detailed description of six different kinds of property change $Z$ included in REINVENT He et al. [35, 36]

- **MMP:** There are user-defined desirable property changes between molecules $X$ and $Y$.
- **Similarity** $\geq 0.5$**:** The Tanimoto similarity between molecules $X$ and $Y$ is greater than 0.5.
- **Similarity** $\in [0.5, 0.7)$**:** The Tanimoto similarity between the pair $(X, Y)$ ranges from 0.5 to 0.7.
- **Similarity** $\geq 0.7$**:** The Tanimoto similarity between molecules $X$ and $Y$ is greater than 0.7.
- **Scaffold:** Molecules $X$ and $Y$ share the same scaffold.
- **Scaffold generic:** Molecules $X$ and $Y$ share the same generic scaffold.

## D.4 BPE Tokenization

Byte Pair Encoding (BPE) is a tokenization algorithm initially designed for data compression and later adapted for use in NLP, particularly in the preprocessing of text for deep learning models. The core idea behind BPE is to iteratively merge the most frequent pair of consecutive bytes (or characters in the context of text) into a single, new byte (or token), thereby reducing the size of the data to be processed. This method has been particularly influential in the development of language models and machine translation systems. The BPE method follows these main steps:

1. **Initial vocabulary preparation:** The text is divided into a sequence of characters or symbols, and a special end-of-word symbol (like <\w> or another unique marker)

is added to each word to distinguish between the same character sequence occurring within a word and at the end of a word.

2. **Frequency Count:** The algorithm counts the frequency of each pair of adjacent characters (or symbols) in the text.

3. **Iterative Merging:**

   - Identify the most frequent pair of adjacent characters.
   - Merge this pair into a new single symbol (this does not mean changing the text itself but rather how the algorithm interprets the text).
   - Update the frequency count of all pairs, considering the newly created symbol.
   - Repeat this process for a predetermined number of iterations or until a desired vocabulary size is reached.

4. **Tokenization:** Once the merging process is complete, the original text can be tokenized (i.e., divided into a sequence of tokens) using the final set of symbols, including the merged ones. This results in a text representation where frequent words or subwords are encoded as single tokens, and less common words are broken down into smaller tokens.

A significant benefit of BPE lies in its capacity to manage rare and out-of-vocabulary words effectively. Since BPE operates at the character level, it can segment words that were not encountered during training, thus reducing the negative effects of unfamiliar words on the model's performance. In contexts where tokens of various lengths are randomly masked and relocated to the end of the sequence, as proposed in section 6.2.1, there's a high likelihood of generating a considerable number of unfamiliar tokens. BPE's approach is particularly beneficial here, as it ensures that the model can still process and understand these novel token

sequences by breaking them down into familiar subunits, thereby maintaining robustness and reducing the potential degradation in performance due to unexpected or rare words.

### *D.4.1 Binding sites of 3clpro and RTCB*

(a) 3CLPro. (b) RTCB.

Figure D.1: The binding sites of selected target proteins 3CLPro (PDB ID: 7BQY) (**Left**) and RTCB (PDB ID: 4DWQ) (**Right**). Atoms around the crystallized compound are defined as binding sites using Open Eye software.

## D.5 Surrogate model

The surrogate model [111] is a simplified version of a BERT-like transformer, widely employed in natural language processing. In this model, tokenized SMILES strings are inputted and then positionally embedded. The outputs are subsequently 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. The 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 produce the predicted docking score.

## D.6 Computing infrastructure and wall-time comparison

We trained our docking surrogate models using 4 nodes of a supercomputer, each node equipped with CPUs (64 cores) and 4 A100 GPUs. The training time for each model was approximately 3 hours. We performed additional pretraining on a cluster consisting of CPU nodes (approximately 280 cores) and GPU nodes (approximately 110 Nvidia GPUs, ranging from Titan X to A6000, primarily configured in 4- and 8-GPU setups).

Pretraining utilizes 8 A100 GPUs, while one single generation uses a single Tesla T4 GPU. Based on the computing infrastructure, pretraining details as described in Appendix D.1 and generation details as described in Appendix D.2, we obtained the wall-time comparison in Table D.1 as follows.

| | Total Run Time |
|---|---|
| **Initial Phase Pretraining** | 18h |
| **Second Phase Pretraining** | 48h |
| **Third Phase Pretraining** | 20h |
| **Generation 10k times for one molecule** | 15mins |

Table D.1: Wall-time comparison between different methods.

## D.7 Hyperparameters and architectures

Table B.2 provides a list of hyperparameter settings we used for our experiments.

For experimentation, 1280 molecules from each of the RTCB and 3CLPro datasets, with docking scores ranging from -14 to -6, are selected. This range is based on [63].

In addition, when calculating the average normalized reward for the original molecule, where similarity is not considered, we select the weights for docking, drug-likeness, synthesizability, and solubility as $[0.25] \times 4$.

| Parameter | Value |
|---|---|
| Pretraining | |
| Learning rate | $5 \times e^{-5}$ |
| Batch size | 24 |
| Optimizer | Adam |
| # of Epochs for Training Initial Phase | 10 |
| # of Epochs for Training Second Phase | 50 |
| # of Epochs for Training Third Phase | 20 |
| Generation | |
| # of Molecules Optimized | 1280 |
| TopK | 20 |
| TopP | 0.95 |

Table D.2: Hyperparameters.

# REFERENCES


[1] Schrödinger announces fda clearance of investigational new drug application for sgr-1505, a malt1 inhibitor, Jun 2022. URL `https://www.businesswire.com/news/home/20220628005165/en/Schr%C3%B6dinger-Announces-FDA-Clearance-of-Investigational-New-Drug-Application-for-SGR-1505-a-MALT1-Inhibitor`.

[2] Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. Htlm: Hyper-text pre-training and prompting of language models. *arXiv preprint arXiv:2107.06955*, 2021.

[3] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

[4] Jeffrey K Aronson and A Richard Green. Me-too pharmaceutical products: History, definitions, examples, and relevance to drug shortages and essential medicines lists. *British Journal of Clinical Pharmacology*, 86(11):2114–2122, 2020.

[5] Stephanie K Ashenden. *The era of artificial intelligence, machine learning, and data science in the pharmaceutical industry*. Academic Press, 2021.

[6] 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.

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

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

[9] Brian J Bender, Stefan Gahbauer, Andreas Luttens, Jiankun Lyu, Chase M Webb, Reed M Stein, Elissa A Fink, Trent E Balius, Jens Carlsson, John J Irwin, et al. A practical guide to large-scale docking. *Nature protocols*, 16(10):4799–4832, 2021.

[10] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38 (8):2102–2110, 2022.

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

[12] 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.

[13] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

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

[17] Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.

[18] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[19] Marianne Defresne, Sophie Barbe, and Thomas Schiex. Protein design with deep learning. *International Journal of Molecular Sciences*, 22(21):11741, 2021.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] 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.

[22] 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.

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

[24] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

[25] Noelia Ferruz and Birte Höcker. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6):521–532, 2022.

[26] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[27] 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.

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

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

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

[31] Zhonghui Gu, Xiao Luo, Jiaxiao Chen, Minghua Deng, and Luhua Lai. Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics*, 39(7):btad410, 2023.

[32] 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.

[33] Anvita Gupta, Alex T Müller, Berend JH Huisman, Jens A Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for de novo drug design. *Molecular informatics*, 37(1-2):1700111, 2018.

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

[35] 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.

[36] 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.

[37] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20:1–17, 2019.

[38] Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *arXiv preprint arXiv:2007.06252*, 2020.

[39] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[40] Xiuyuan Hu, Guoqing Liu, Yang Zhao, and Hao Zhang. De novo drug design using reinforcement learning with multiple gpt agents. *Advances in Neural Information Processing Systems*, 36, 2024.

[41] Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, pages 1–11, 2024.

[42] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1): 177–182, 2005.

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

[44] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.

[45] Mingjian Jiang, Zhen Li, Shugang Zhang, Shuang Wang, Xiaofeng Wang, Qing Yuan, and Zhiqiang Wei. Drug–target affinity prediction using graph neural network and contact maps. *RSC advances*, 10(35):20701–20712, 2020.

[46] Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, WF Drew Bennett, Daniel Kirshner, Sergio E Wong, Felice C Lightstone, and Jonathan E Allen. Improved protein–ligand binding affinity prediction with structure-based deep fusion inference. *Journal of chemical information and modeling*, 61(4):1583–1592, 2021.

[47] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77, 2020.

[48] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. drugan: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics*, 14(9):3098–3104, 2017.

[49] 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.

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

[51] Surafel M Lakew, Mauro Cettolo, and Marcello Federico. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *arXiv preprint arXiv:1806.06957*, 2018.

[52] Greg Landrum et al. RDkit: Open-source cheminformatics software, 2016. `https://www.rdkit.org`. Accessed Oct 2023.

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

[54] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 975–985, 2021.

[55] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11418–11425, 2020.

[56] Yibo Li, Jianxing Hu, Yanxing Wang, Jielong Zhou, Liangren Zhang, and Zhenming Liu. Deepscaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. *Journal of chemical information and modeling*, 60(1):77–91, 2019.

[57] 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.

[58] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 10(1):1–9, 2018.

[59] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

[60] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

[61] 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*, 2023.

[62] Xuefeng Liu, Takuma Yoneda, Chaoqi Wang, Matthew R Walter, and Yuxin Chen. Active policy improvement from multiple black-box oracles. In *icml*, pages 22320–22337, 2023.

[63] 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.

[64] 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*, 2024.

[65] Qi Lv, Feilong Zhou, Xinhua Liu, and Liping Zhi. Artificial intelligence in small molecule drug discovery from 2018 to 2023: Does it really work? *Bioorganic Chemistry*, page 106894, 2023.

[66] Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Universal text representation from bert: An empirical study. *arXiv preprint arXiv:1910.07973*, 2019.

[67] Kit-Kay Mak, Yi-Hang Wong, and Mallikarjuna Rao Pichika. Artificial intelligence in drug discovery and development. *Drug Discovery and Evaluation: Safety and Pharmacokinetic Assays*, pages 1–38, 2023.

[68] 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.

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

[70] Hamilton Moses, E Ray Dorsey, David HM Matheson, and Samuel O Thier. Financial anatomy of biomedical research. *Jama*, 294(11):1333–1342, 2005.

[71] Scott Myers and Ann Baker. Drug discovery—an operating model for a new era. *Nature biotechnology*, 19(8):727–730, 2001.

[72] 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.

[73] Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021.

[74] Wern Juin Gabriel Ong, Palani Kirubakaran, and John Karanicolas. Poor generalization by current deep learning models for predicting binding affinities of kinase inhibitors. *bioRxiv*, 2023.

[75] Si-sheng Ou-Yang, Jun-yan Lu, Xiang-qian Kong, Zhong-jie Liang, Cheng Luo, and Hualiang Jiang. Computational drug discovery. *Acta Pharmacologica Sinica*, 33(9): 1131–1140, 2012.

[76] 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.

[77] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[78] 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.

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

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

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

[82] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[83] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[84] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21 (1):5485–5551, 2020.

[85] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[86] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[87] Mohammad A Rezaei, Yanjun Li, Dapeng Wu, Xiaolin Li, and Chenglong Li. Deep learning in drug design: protein-ligand binding affinity prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(1):407–417, 2020.

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

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

[90] Anastasiia V Sadybekov and Vsevolod Katritch. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685, 2023.

[91] Neil Savage. Drug discovery companies are customizing chatgpt: here's how. *Nature Biotechnology*, 2023.

[92] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005.

[93] Schrödinger, LLC. Hit development to candidate in 10 months: Rapid discovery of a novel potent malt1 inhibitor. `https://newsite.schrodinger.com/life-science/learn/case-studies/hit-development-candidate-10-months-rapid-discovery-novel-potent-malt1-inhibitor/`, 2023. Accessed: 2024-04-03.

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

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

[96] Chandra Shekhar. In silico pharmacology: computer-aided methods could transform drug development. *Chemistry & biology*, 15(5):413–414, 2008.

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

[98] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018.

[99] Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe. Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395, 2014.

[100] Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34: 25244–25255, 2021.

[101] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[102] 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.

[103] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5229–5238, 2019.

[104] Jing Tang, Agnieszka Szwajda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.

[105] 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.

[106] Xiaochu Tong, Xiaohong Liu, Xiaoqin Tan, Xutong Li, Jiaxin Jiang, Zhaoping Xiong, Tingyang Xu, Hualiang Jiang, Nan Qiao, and Mingyue Zheng. Generative models for de novo drug design. *Journal of Medicinal Chemistry*, 64(19):14011–14027, 2021.

[107] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[108] Alex M Tseng, Nathaniel Diamant, Tommaso Biancalani, and Gabriele Scalia. Graphguide: interpretable and controllable conditional graph generation with discrete bernoulli diffusion. *arXiv preprint arXiv:2302.03790*, 2023.

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

[110] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage

of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1): D439–D444, 2022.

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

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

[113] S Vincent Rajkumar. The high cost of prescription drugs: causes and solutions. *Blood cancer journal*, 10(6):71, 2020.

[114] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. Comenet: Towards complete and efficient message passing for 3d molecular graphs. *Advances in Neural Information Processing Systems*, 35:650–664, 2022.

[115] Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3d graph networks. In *International Conference on Learning Representations (ICLR)*, 2023.

[116] Mingyang Wang, Zhe Wang, Huiyong Sun, Jike Wang, Chao Shen, Gaoqi Weng, Xin Chai, Honglin Li, Dongsheng Cao, and Tingjun Hou. Deep learning approaches for de novo drug design: An overview. *Current Opinion in Structural Biology*, 72:135–144, 2022.

[117] Penglei Wang, Shuangjia Zheng, Yize Jiang, Chengtao Li, Junhong Liu, Chang Wen, Atanas Patronov, Dahong Qian, Hongming Chen, and Yuedong Yang. Structure-aware multimodal deep learning for drug–protein interaction prediction. *Journal of chemical information and modeling*, 62(5):1308–1317, 2022.

[118] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12): 4111–4119, 2005.

[119] 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.

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

[121] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1): D1074–D1082, 2018.

[122] 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.

[123] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021.

[124] Ziduo Yang, Weihe Zhong, Qiujie Lv, Tiejun Dong, and Calvin Yu-Chian Chen. Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3d structures (gign). *The Journal of Physical Chemistry Letters*, 14(8): 2020–2033, 2023.

[125] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

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

[127] 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.

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

[129] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[130] Haiping Zhang, Linbu Liao, Konda Mani Saravanan, Peng Yin, and Yanjie Wei. Deepbindrg: a deep learning based method for estimating effective protein–ligand affinity. *PeerJ*, 7:e7362, 2019.

[131] 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*, 2023.

[132] 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.

[133] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

[134] Zuobai Zhang, Minghao Xu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Enhancing protein language models with structure-based encoder and pre-training. *arXiv preprint arXiv:2303.06275*, 2023.

[135] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

[136] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 37(9):1038–1040, 2019.

[137] 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.