RL-Guider: Leveraging Historical Decisions and Feedback for Drug Editing with Large Language Models

Xufeng Liu[*], **Yixuan Ding**[*†], **Jingxiang Qu**, **Yichi Zhang**, **Wenhan Gao**[‡], **Yi Liu**[‡],
Stony Brook University
xufeng.liu@stonybrook.edu, yi.liu.4@stonybrook.edu

Abstract

Recent success of large language models (LLMs) in diverse domains showcases their potential to revolutionize scientific fields, including drug editing. Traditional drug editing relies on iterative conversations with domain experts, refining the drug until the desired property is achieved. This interactive and iterative process mirrors the strengths of LLMs, making them well-suited for drug editing. *In existing works, LLMs edit each molecule independently without leveraging knowledge from past edits.* However, human experts develop intuition about effective modifications over time through historical experience; accumulating past knowledge is pivotal for human experts, and so it is for LLMs. *In this work, we propose RL-Guider—a reinforcement-learning-agent to provide suggestions to LLMs; it uses the rich information provided from evaluating editing results made by the LLM based on the recommendations to improve itself over time.* RL-Guider is the first work that leverages both the comprehensive "world-level" knowledge of LLMs and the knowledge accumulated from historical feedback. As a result, RL-Guider mitigates several shortcomings of existing approaches and demonstrates superior performance. The code is available at `https://github.com/xufliu/RL-Guider`.

1 Introduction

The remarkable performance of large language models (LLMs) across various tasks has recently sparked growing interest in their application to scientific domains (Zhang et al., 2023), such as drug editing, which is typically a complex, iterative process that integrates expert knowledge and refinement. Drug editing is a specialized task of molecular optimization, aimed at refining drug-like molecules through small, localized structural modifications. Typically, it begins with a conversation with domain experts to gather suggestions on modifications. The molecule is then modified accordingly and subjected to computational or experimental property testing. If the revised drug meets the desired criteria, the process concludes; otherwise, the testing results are provided to the domain expert to inform further refinements (Zheng et al., 2024; Seidl et al., 2023; Cao et al., 2023; Wu et al., 2024). This iterative refinement process in drug editing perfectly aligns with the greatest strength of LLMs—their ability to engage in interactive conversations and integrate suggestions.

The process of LLM-assisted drug editing occurs in iterative rounds, where each cycle refines the molecular structure based on suggestions or guidance. These suggestions can range from simple responses, such as indicating whether the result is incorrect, to more complex guidance incorporating detailed chemical knowledge and specific modification suggestions. As a powerful generative tool with "world-level" knowledge, LLMs can leverage their vast pre-trained knowledge to propose structurally valid and chemically meaningful modifications. However, their effectiveness heavily depends on the quality of the provided prompts, suggestions, and guidance. Hence, the core of LLM-assisted drug editing lies in the design of valuable and knowledgeable suggestions to guide the LLM to reason and plan (Liu et al., 2024b; Ye et al., 2023; Wu et al., 2025; Ma et al., 2024).

Existing works mainly focus on leveraging known domain knowledge, such as through a retrieval database (Liu et al., 2024b) or another LLM (Sprueill et al., 2024), with the assumption that LLMs store the chemical domain knowledge. **However, these approaches can introduce biases that constrain the exploration of novel molecular structures.** Over-reliance on existing knowledge risks anchoring the drug editing process to

[*] Equal contribution.

[†] Work done during an internship at Stony Brook University.

[‡] Equal senior contribution.

Findings of the Association for Computational Linguistics: ACL 2025, pages 13121–13138
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

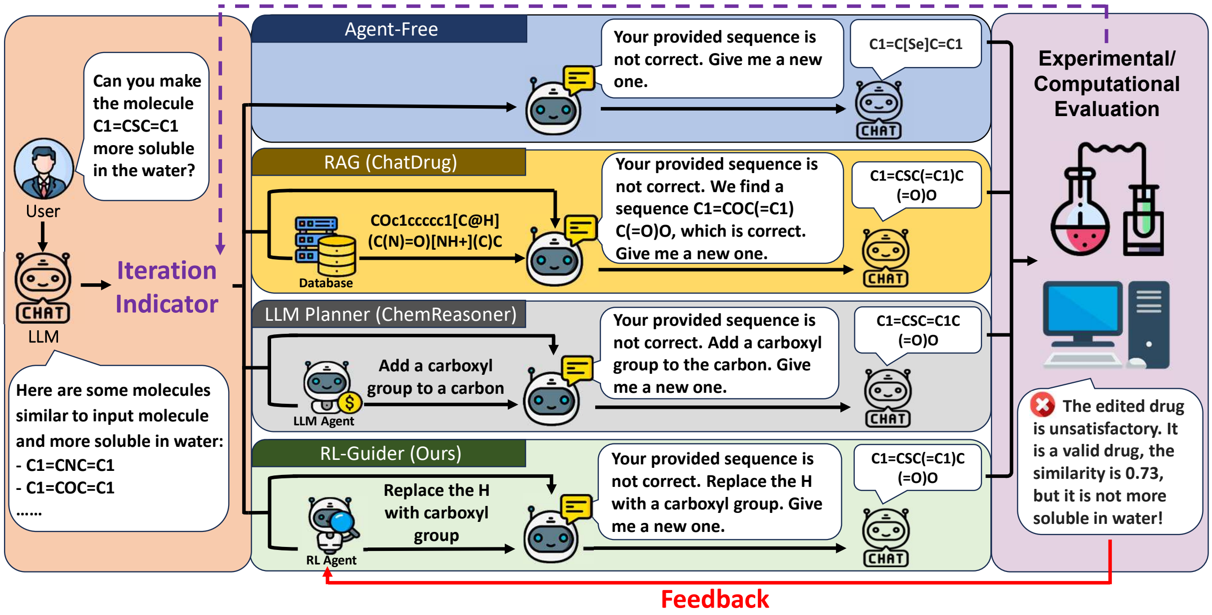
Can you make the molecule C1=CSC=C1 more soluble in the water?
User
LLM
CHAT
Iteration Indicator
Here are some molecules similar to input molecule and more soluble in water:
- C1=CNC=C1
- C1=COC=C1
......
Agent-Free
Your provided sequence is not correct. Give me a new one.
C1=C[Se]C=C1
RAG (ChatDrug)
Database
COc1ccccc1[C@H](C(N)=O)[NH+](C)C
Your provided sequence is not correct. We find a sequence C1=COC(=C1)C(=O)O, which is correct. Give me a new one.
C1=CSC(=C1)C(=O)O
LLM Planner (ChemReasoner)
LLM Agent
Add a carboxyl group to a carbon
Your provided sequence is not correct. Add a carboxyl group to the carbon. Give me a new one.
C1=CSC=C1C(=O)O
RL-Guider (Ours)
RL Agent
Replace the H with carboxyl group
Your provided sequence is not correct. Replace the H with a carboxyl group. Give me a new one.
C1=CSC(=C1)C(=O)O
Experimental/ Computational Evaluation
The edited drug is unsatisfactory. It is a valid drug, the similarity is 0.73, but it is not more soluble in water!
Feedback

Figure 1: An overview of various suggestion agents for LLM-assisted drug editing. LLM-assisted drug editing is an iterative process in which the LLM generates an edited drug, followed by an experimental or computational evaluation. If the edited drug is unsatisfactory, the LLM is informed, and the process restarts. To effectively adapt general-purpose LLMs with "world-level" knowledge for the drug editing task, providing proper suggestions or guidance in the form of prompts is crucial. Various agents have been proposed to automate prompt generation. ChatDrug (Liu et al., 2024b) is a Retrieval-Augmented Generation (RAG) based pipeline that retrieves the drug most similar to the original drug among all drugs satisfying the target property. ChemReasoner (Sprueill et al., 2024) uses another LLM as a planner to systematically generate suggestions, assuming it is knowledgeable about the chemical domain. **Existing approaches fail to leverage the rich feedback from evaluating edited drugs.** Contrarily, we propose RL-Guider, a reinforcement learning-based (RL-based) agent specifically designed to efficiently utilize this feedback. As RL-Guider continuously interacts with the LLM to assist in the editing process, **RL-Guider refines its strategies by learning from historical decisions and feedback, much like human experts who accumulate knowledge and expertise over time.**

familiar patterns, potentially overlooking novel and unseen molecular modifications that could lead to breakthrough discoveries. **Moreover, these methods can incur substantial computational costs**; for instance, retrieval-based inference requires iterating over the entire database, significantly increasing latency and resource consumption. Notably, in existing methods, each editing task is treated independently, overlooking the accumulation of knowledge. Motivated by how expert chemists—they accumulate intuition by learning from past modifications, gradually improving their ability to identify key substructures that require adjustment, the accumulation of historical experience from past edits is critical for both human experts and LLMs.

To overcome the aforementioned issues, we propose a reinforcement learning agent-enhanced drug editing pipeline, referred to as RL-Guider, that learns from the results of past edit actions gradually as the LLM tackles more editing tasks. **Unlike supervised approaches, RL-Guider operates without the need for re-training or fine-tuning of the LLM**, making it a lightweight and flexible solution for delivering high-quality suggestions and guidance in iterative drug editing. **RL-Guider is the first framework that leverages both the comprehensive "world-level" knowledge of LLMs and the historical experience accumulated from past edits by effectively utilizing the rich evaluation feedback of candidate drugs.** An overview of various suggestion agents is provided in Fig. 1. Overall, **the contributions of this work can be summarized**: (1) We identify a rich source of information that has been overlooked by existing works—the evaluation feedback on candidate drugs; (2) We propose RL-Guider, an RL-based agent that leverages this rich information without requiring retraining or fine-tuning of LLMs; (3) RL-Guider does not rely on predefined knowledge or a fixed retrieval dataset, offering greater flexibility and adaptability in editing novel drugs; (4) We demonstrate through extensive experiments on various tasks and backbone LLMs that RL-Guider is efficient and effective, consistently improving drug editing performance.

2 Related Work

Machine learning has achieved remarkable success in molecular analysis (Schütt et al., 2017; Brandstetter et al., 2021; Liu et al., 2020, 2022; Wang et al., 2022, 2023; Yan et al., 2022; Subedi et al., 2024; Liu et al., 2025; Qu et al., 2025), which provides a powerful tool for drug discovery (Zeng et al., 2022; Flam-Shepherd et al., 2022). Moreover, in recent years, LLMs (Team et al., 2023; Achiam et al., 2023) have shown superior performance across diverse NLP tasks including text translation (Huang et al., 2023) and editing (Bi et al., 2024). Consequently, several studies have explored the application of LLMs to drug discovery. For example, pioneering research has employed either fine-tuned LLMs (Cao et al., 2023) or LLMs retrained on existing datasets (Liang et al., 2023) to generate potential drug candidates.

Despite being a crucial task in drug discovery, drug editing remains exceedingly underexplored. Compared with discovery, drug editing has stricter requirements for new drugs, including molecular validity, structural similarity, and desired properties, as introduced in Sec. 3.1. To the best of our knowledge, there exist only a few works in this line of research. Notably, Sprueill et al. (2023); Liu et al. (2024b) employ a domain-specific database to provide similar known drugs that satisfy the desired property as guidance. Ma et al. (2024); Sprueill et al. (2024) utilize tools or trained models with predefined domain knowledge to guide the LLMs.

Relation with Prior Works. Existing works primarily focus on leveraging predefined knowledge, such as retrieval databases (Liu et al., 2024b) or pretrained knowledgeable LLMs (Sprueill et al., 2024). The details of these methods are introduced in Appendix A. While these methods can provide domain-specific insights, they may suffer from significant computational costs and potential biases introduced by relying on predefined knowledge. We discuss these shortcomings further in Sec. 4.2. We observe that existing works overlook a crucial source of knowledge, leaving it unused in the editing process. In particular, during the iterative editing process, the edited drugs are evaluated to assess their suitability, providing valuable feedback on key aspects such as validity, structural similarity, and chemical properties of the LLM-edited drug. This rich information can be used to refine and improve the LLM's editing performance over time; however, existing works utilize this feedback merely as a termination or continuation indicator for the iterative process based on whether the results are satisfactory. **In our proposed RL-Guider, we explicitly utilize this rich information from the feedback as a reward or penalty to train our reinforcement learning agent, allowing it to accumulate knowledge from historical experience and overcome limitations of existing works.**

3 Preliminaries

3.1 LLM-assisted Drug Editing

Drug editing modifies a molecule into another one while maintaining structural similarity and achieving desired properties (Chen et al., 2021). Given a drug x_{in} and a text prompt x_{t} describing the target or desired property, drug editing is a conditional generation problem (Liu et al., 2024b) in which the goal is to obtain an optimized drug $x_{\text{out}} \sim P(x \mid x_{\text{in}}, x_{\text{t}})$. In the context of LLM-assisted drug editing, P is realized as $x_{\text{out}} = \text{LLM}(x_{\text{in}}, x_{\text{t}})$. However, unlike traditional deep generative models, which are explicitly trained on large datasets to optimize an objective function—such as maximizing the likelihood or the evidence lower bound (ELBO)—LLMs operate primarily through in-context learning (Dong et al., 2022) and rely heavily on the quality of provided prompts and suggestions (Liu et al., 2023). Moreover, the utilization of the conversational potentials in LLMs is preferred (Liu et al., 2024b; Bubeck et al., 2023) and the editing can be an iterative process using LLMs. Therefore, we formally formulate LLM-assisted drug editing as:

$$x_i = \text{LLM}(x_{\text{in}}, x_{\text{t}}, x_{\text{s}}^i) \tag{1}$$

for iteration rounds $i = 0, 1, 2, \cdots, K$, with $x_{\text{out}} = x_K$. Here, for clarity of notation, we break the prompt into two parts: x_{t} and x_{s}. x_{t} is the textual description of the target or desired property as defined previously, and x_{s}^i is the suggestion provided to LLMs by the agent in the i-th round, analogous to the suggestions from expert chemists in the iterative drug editing process. Following Liu et al. (2024b), no suggestion will be provided in the 0-th round, i.e., x_{s}^0 is taken as an empty string. As a natural input to LLMs, drugs are typically represented in text formats, such as SMILES strings.

3.2 Reinforcement Learning Agent

Reinforcement learning is a machine learning method that enables an agent to learn optimal actions by interacting with its environment through rewards and penalties. Following Kumar et al. (2020), the RL framework is defined as (S, A, P, R, γ), where S represents the state space and A is the action space. At time step t, the state and action are denoted as s_t and a_t, respectively. The state transition probability is given by $P(s_{t+1} \mid s_t, a_t)$, and the reward function $R(s_t, a_t)$ evaluates the action a_t at state s_t. The discount factor γ determines the weight of future rewards in decision-making. The goal of training is to learn an optimal policy $\pi(\cdot)$ that determines an action based on the current state, i.e., $a_t = \pi(s_t)$. The optimization objective of RL is to maximize the expected cumulative reward as:

$$\pi \leftarrow \arg \max_{\pi(s_t)\in A} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, \pi(s_t))\right], \tag{2}$$

where T is the maximum number of time steps.

4 RL-Guider: Guidance By Historical Experience

In this section, we introduce RL-Guider, our proposed method for providing more effective guidance. In Sec. 4.1, we lay out the formulation of RL-Guider and in Sec. 4.2, we discuss the merits and significant potential it holds.

4.1 Formulation of RL-Guider

We first provide an overview of the RL-Guider pipeline:

> **Overview.** Given a molecule and the target, RL-Guider provides suggestions (actions) on the key components to be modified. Importantly, RL-Guider provides suggestions but does not directly edit the drug; for instance, a suggestion of adding a functional group may result in an invalid drug. Instead, the LLM leverages its extensive knowledge of chemistry to interpret the suggestion and modify the drug accordingly. The edited candidate drug will then be evaluated through computational or experimental testing to provide feedback on the molecular validity, structural similarity compared with the input drug, and the value of the desired chemical property. This feedback, in turn, serves as the reward/penalty for RL-Guider's suggestions. This process can be done interactively and iteratively for several rounds if the edited drug is unsatisfactory.

Notably, our framework is the only method that utilizes the rich information provided by the testing feedback; all existing works only use them as indicators to continue or terminate the iterative process. Over time, as more interactions with the LLMs occur and more editing are performed, RL-Guider accumulates extensive historical experience in providing effective suggestions.

Specifically, RL-Guider is a reinforcement learning pipeline that learns to make suggestions to maximize cumulative rewards; unlike supervised learning, which relies on a training dataset with predefined answers, RL-Guider involves learning through experience by giving suggestions and receiving feedback on the qualities of the suggestions made. As an RL-based learning pipeline, we summarize the key components of RL-Guider:

- **State:** The LLM's context that contains information about the original drug, target, and the string representation of the edited drug.
- **Action:** Giving suggestions in the form of a prompt in order to improve the LLM's context, especially the edited drug.
- **Reward:** Determined by evaluation feedback on the proposed edited drug.
- **Policy:** The strategy used by the RL agent to propose modifications.
- **Environment:** The LLM.

The state and environment are straightforward and defined immediately. The primary challenge lies in designing an effective action space and formulating an appropriate reward and penalty structure to ensure meaningful and optimal learning. Once the action space and reward function are determined, the policy follows a standard Q-learning framework (Kumar et al., 2020). In the following, we illustrate the modeling of the action space using small molecules as a representative example for simplicity. However, this framework can be seamlessly extended to larger molecules, such as peptides and proteins, with minimal modifications.

Modeling the Action Space. We construct a structured representation of possible modification suggestions. Specifically, the action space is designed to encompass three fundamental operations on molecular structures: *addition*, *deletion*, and *replacement*. These operations correspond to suggestions for introducing, removing, or substituting specific atoms or functional groups in the input drug that RL-Guider identifies as beneficial for achieving the desired properties. A sample suggestion prompt is provided below:

> *Edit the molecule O=C(C)Oc1ccccc1C(=O)O by following the suggestion: Replace ester group with hydroxyl group. Give me results in SMILES only and list them using bullet points.*

Reward and Penalty. The reward function is critical for guiding RL-Guider toward meaningful suggestions. Instead of a simple binary reward structure—whether the suggestion leads to a satisfactory editing result or not, we design a multifaceted reward signal incorporating three main factors: ❶ *Molecular Validity*—Ensuring that the modified molecule remains chemically valid and synthesizable; ❷ *Structural Similarity*—Encouraging modifications that retain the core features of the original drug while introducing beneficial changes; ❸ *Target Property Optimization*—Rewarding suggestions that lead to improvements in desired chemical properties, such as binding affinity and solubility. Mathematically, the reward function given the state s_t and action a_t for the t-th round is:

$$R(s_t, a_t) = \psi(\sigma_t, s_m, p_m), \tag{3}$$

where σ_t, s_m, and p_m quantify molecular validity, structural similarity, and the degree to which the property value of the edited drug aligns with that of the target, respectively, as numerical values. ψ is a reward aggregation function that combines these quantities, e.g., through weighted addition or multiplication, to produce a single scalar value that quantifies the quality of the action. More details on implementation of RL-Guider building and training are provided in Appendix B.

4.2 Merits of RL-Guider

In all existing works, editing tasks are treated independently, with no knowledge transfer to future edits. Once an edit is completed, it offers no insights for future tasks, leaving valuable feedback from testing the edited drug unutilized. RL-Guider utilizes this rich information as the reward/penalty to improve itself. Over time, RL-Guider accumulates knowledge and develops intuition, enabling it to make more informed and efficient edit suggestions. By doing so, RL-Guider not only provides superior performance, as demonstrated in Sec. 5.1, but also overcomes two significant drawbacks in existing works: (1) computational efficiency; (2) bias mitigation.

Computational Efficiency. Existing approaches exhibit significant computational overhead—retrieval-based approaches require iterating over the entire database and LLM-based approaches suffer from the computational costs of LLMs. On the other hand, RL-Guider mitigates these computational challenges by learning a policy that generalizes from past editing tasks. Once RL-Guider has become a knowledgeable agent, a single inference is significantly more efficient. As shown in the experimental results in Table 4 in Sec. 5, on average, **it takes only 0.03 seconds for RL-Guider to generate a suggestion prompt while it takes 5.67 seconds and 5.15 seconds for ChatDrug and ChemReasoner, respectively**. Additionally, even including the training time, RL-Guider is still far more efficient than these two methods—**it takes 2.28 seconds for RL-Guider to both generate a suggestion and learn (by training) from the feedback from the resulting edited drug**, and it is still significantly faster than both methods.

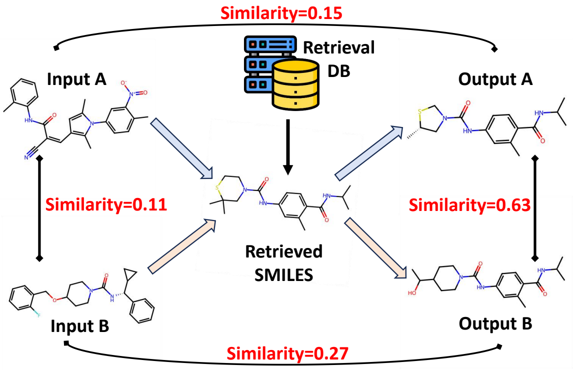

Figure 2: Editing results from *real* experiments using ChatDrug. It is evident that the outputs are biased toward the retrieved drug—the outputs are highly similar to the retrieved drug, resulting in high similarity between the outputs and low similarity between the input-output pairs.

Bias Mitigation. Existing approaches often introduce biases. For example, LLMs inherently reflect biases from their training data (Gallegos et al., 2024), making them prone to generating modifications based on drugs they have seen. Similarly, retrieval-based methods limit exploration to known molecules within the database. **As a result, these biases cause the edited drug to closely resemble the reference drug used as guidance**, potentially hindering the discovery of novel and structurally diverse modifications. For different inputs A and B, they may share the same retrieved reference

Single Target Property	Δ	**LLaMA**				**DeepSeek**			
		Agent Free	**Chat Drug**	**Chem Reasoner**	**RL-Guider**	**Agent Free**	**Chat Drug**	**Chem Reasoner**	**RL-Guider**
More soluble in water	0	45.50	45.00	51.50	60.00	81.00	63.50	81.00	82.00
	0.5	33.50	45.50	31.50	52.50	68.50	68.50	66.00	75.50
Less soluble in water	0	55.00	54.00	57.00	58.50	89.00	85.00	87.50	92.00
	0.5	51.50	56.50	51.00	55.50	80.50	80.50	81.00	83.00
More like a drug	0	30.00	43.50	32.00	43.00	34.00	64.00	53.00	78.00
	0.1	5.50	24.00	5.00	9.50	5.50	34.00	10.50	10.00
Less like a drug	0	43.50	44.50	48.00	57.00	73.50	73.00	73.00	69.50
	0.1	17.00	43.50	14.00	60.00	28.00	58.50	13.00	70.50
Higher permeability	0	26.00	44.00	44.00	62.00	38.00	72.50	64.50	78.00
	10	11.00	46.50	34.00	44.50	23.50	71.50	50.50	57.00
Lower permeability	0	48.00	48.00	51.50	60.00	76.00	53.00	74.00	82.00
	10	50.50	41.50	51.00	54.00	74.50	52.50	74.50	77.50
More hydrogen bond acceptors	0	49.00	53.50	51.50	61.00	75.00	76.50	74.50	75.00
	1	35.50	38.50	13.50	46.00	30.00	64.50	19.00	65.00
More hydrogen bond donors	0	39.00	29.00	39.00	50.50	65.50	24.50	50.00	57.00
	1	15.00	16.50	5.50	20.50	4.00	26.50	10.50	20.50

Table 1: Quantitative comparison of success rate (%) based on LLaMA and DeepSeek with **multi-round** interaction. The best and second-best results are highlighted in red and blue, respectively. Obviously, RL-Guider demonstrates superior performance against baseline methods across a diverse set of tasks.

drug, as the reference drug needs to satisfy the target property, leading to limited selection. Even if A and B are structurally very different, the retrieved reference drug might still be the same. In Fig. 2, we present **real** experimental results from ChatDrug. It is clear that the reference drug introduces bias as the editing results resemble the retrieved drug, resulting in extremely low similarity between the edited drug and the original drug.

In contrast, RL-Guider mitigates these biases by learning directly from interactions and feedback from past editing results with a reinforcement learning framework rather than predefined knowledge (RAG or pretrained LLM). **Notably, as a learning-based agent, RL-Guider penalizes itself when the edited drug deviates from the original drug, as structural similarity from feedback provided during the evaluation of edited drugs is inherently considered in the reward function. This allows RL-Guider to mitigate biases that often arise from reliance on predefined knowledge-based data**.

5 Experiments

In this section, we demonstrate the effectiveness and efficiency of RL-Guider through extensive experiments. First, we compare the performance of our RL-Guider against several baselines in Sec. 5.1. Additionally, we conduct ablation studies in Sec. 5.2 to evaluate the computational efficiency, learning capability from accumulated historical experience, and hyperparameter sensitivity of RL-Guider.

5.1 RL-Guider Performance

Setup. We first conduct experiments to verify the effectiveness of RL-Guider against the baseline LLMs without any guidance (AgentFree), as well as ChatDrug and ChemReasoner, following the same drug editing tasks and setup as in Liu et al. (2024b). Specifically, the editing process consists of up to three rounds. In the first round, no method incorporates an agent or planner, and the edited drug is then evaluated. If the results are unsatisfactory, a second round is performed with suggestions provided to all methods. Similarly, a third round is conducted if the results remain unsatisfactory. This setup is referred to as *multi-round interactions*.

All methods are agnostic to the choice of LLM; experiments are conducted with Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and DeepSeek-V3 (Liu et al., 2024a). A detailed description of the experimental setup and prompts is provided in Appendix C and Appendix D, respectively.

Evaluation Metric. The performance is evaluated with *success rate*, which is defined as the proportion of generated molecules that are both valid and meet the desired target. An edited drug meets the target if its property value improves by at least Δ in the desired direction (greater or smaller) compared to the original drug. For example, if $\Delta = 10$ and the task is to have higher permeability, the permeability of the edited drug must be greater than that of the original by 10 to be considered a success.

Main Results. The success rates of all methods under multi-round interactions are shown in Table 1. Additionally, we provide the results under

Single Target Property	Δ	LLaMA				DeepSeek			
		Agent Free	Chat Drug	Chem Reasoner	RL-Guider	Agent Free	Chat Drug	Chem Reasoner	RL-Guider
More soluble in water	0	47.50	27.50	35.00	49.50	65.00	74.00	80.00	80.00
	0.5	22.50	23.00	24.00	44.00	45.00	70.00	36.50	76.00
Less soluble in water	0	48.00	21.00	46.00	68.00	67.00	83.50	67.50	69.00
	0.5	33.50	22.00	26.50	46.50	37.50	62.50	38.50	64.00
More like a drug	0	17.00	16.00	21.00	36.50	26.00	37.00	22.00	39.50
	0.1	2.50	9.50	3.00	7.00	2.50	13.50	3.50	10.50
Less like a drug	0	41.00	33.50	30.00	65.00	35.00	51.00	42.00	72.50
	0.1	2.50	19.00	11.00	52.50	2.00	19.00	23.50	68.00
Higher permeability	0	12.00	6.00	32.00	53.50	36.00	63.00	65.00	84.50
	10	8.00	7.50	35.00	63.00	27.50	59.00	59.00	73.50
Lower permeability	0	40.00	13.50	41.50	54.00	65.00	54.50	62.00	78.50
	10	38.00	11.50	37.50	50.50	66.50	47.00	67.00	80.50
More hydrogen bond acceptors	0	47.00	51.50	37.50	51.50	65.50	73.00	63.50	79.00
	1	3.50	40.00	2.00	25.50	7.00	20.00	5.00	49.00
More hydrogen bond donors	0	28.00	41.50	38.50	67.00	46.50	68.00	75.00	39.50
	1	2.50	37.50	3.50	22.50	3.50	19.50	1.00	3.50

Table 2: Quantitative comparison of success rate (%) based on LLaMA and DeepSeek with **single-round** interaction to focus on the effect of different agents. The best and second-best results are highlighted in red and blue, respectively. RL-Guider demonstrates even better performance compared to the multi-round setting; RL-Guider is the best performer in more tasks. This result clearly once again demonstrates the effectiveness of RL-Guider.

the same setting but that need to satisfy multiple target properties in Table 8 in Appendix E.1. Clearly, RL-Guider achieves superior performance, attaining the highest success rates in most cases. **RL-Guider does not perform the best on only a few tasks, and these results may stem from inherent variations in the LLM itself.** In certain cases, conflicts between the guidance provided by the agents and the knowledge encoded within the LLM can affect the generation process, leading to sub-optimal suggestions despite RL-Guider's overall effectiveness. **This phenomenon is also observed in other methods, as the second-best method varies significantly across different tasks.** A general outperformance trend across all tasks strongly indicates the superiority and potential of RL-Guider. Furthermore, we conduct experiments on peptide editing tasks using MHCFlurry (O'Donnell et al., 2020), with results presented in Appendix E.2. RL-Guider consistently outperforms the baseline methods, demonstrating its potential in editing more complex and larger molecular structures.

Isolating the Effect of Agents. In Table 2, we conduct experiments to isolate the effect of various agents for a more principled comparison between different methods. The interaction with LLMs is restricted to a single round, with suggestions by various agents. RL-Guider consistently outperforms or matches the best results in most cases. Despite the absence of iterative feedback, the underlying RL-based mechanism of our method still effectively facilitates accurate drug editing.

Generalizability to the Chemical Foundation Model. We also evaluate the generalizability of the proposed method using ChemDFM-v1.5-8B (Zhao et al., 2024), an open-source dialogue foundation model for chemistry and molecular science. ChemDFM is fine-tuned from LLaMA-13B using 3.9 million chemical papers and 1.4 thousand books for domain pretraining, along with 1.7 million molecule-related prompts and 1.0 million natural language prompts for instruction tuning. The success rates of all methods on single-property editing tasks under multi-round interactions are shown in Table 3. Compared to the results obtained with the general-purpose LLaMA model (Table 1), all methods show improved performance. Notably, RL-Guider surpasses all baselines by an even larger margin. It achieves the highest success rate on 15 out of 16 tasks and ranks second on the remaining one, clearly demonstrating the strong generalizability of our proposed RL-Guider.

Case Study of Similarity. Maintaining similarity in drug editing is crucial to ensure that the modified drug retains the core structure and functionality of the original molecule. We provide visualizations and similarity calculations for a molecule editing task performed by the three agent-based methods in Fig. 3. In addition, while it is difficult to provide visualizations for all the test examples, we provide quantitative results in Table 10 in Appendix E.3. These results can be summarized as follows: **RL-Guider achieves the highest similarity of 0.718 while ChemReasoner is the second**

Single Target Property	Δ	Agent Free	Chat Drug	Chem Reasoner	RL-Guider
More soluble in water	0	65.50	59.50	72.50	85.50
	0.5	61.50	49.00	63.00	77.50
Less soluble in water	0	78.50	74.50	67.00	81.00
	0.5	61.50	62.50	64.00	68.00
More like a drug	0	42.50	54.00	49.00	57.50
	0.1	14.00	30.00	10.00	20.50
Less like a drug	0	75.50	66.00	73.50	83.50
	0.1	55.00	59.00	48.00	80.50
Higher permeability	0	46.00	66.50	52.50	75.50
	10	28.00	52.00	33.00	62.00
Lower permeability	0	54.00	64.50	59.00	84.00
	10	40.50	55.50	56.00	71.00
More hydrogen bond acceptors	0	64.50	64.00	61.00	81.00
	1	34.00	54.00	33.00	79.50
More hydrogen bond donors	0	54.50	56.00	66.00	69.50
	1	15.50	34.00	29.00	64.50

Table 3: Quantitative comparison of success rate (%) based on ChemDFM with **multi-round** interaction. The best and second-best results are highlighted in red and blue, respectively. Notably, for RL-Guider, we do not retrain the agent; instead, we directly reuse the agent trained for the results in Table 1. This demonstrates its generalizability and transferability across different backbone LLMs without requiring additional training or fine-tuning. RL-Guider can be easily extended to other LLMs and consistently delivers strong performance.

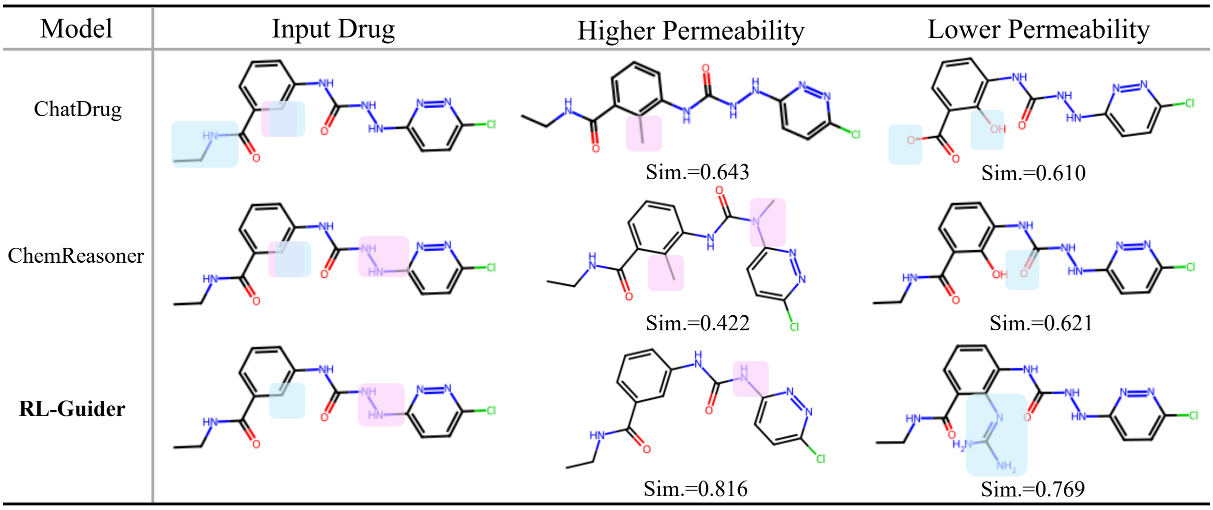

Figure 3: Visualization of a drug editing task performed by all agent-based models. RL-Guider is the only method that consistently maintains high structural similarity while achieving the desired property changes. Other methods that are based on predefined knowledge all suffer from inherent bias towards the predefined knowledge and often produce drugs unrelated to the original input drug. Sim. is a value between 0 and 1 that measures the similarity between the input drug and the output edited drug **(the closer to 1, the better)**. The purple regions and blue regions correspond to the edited substructures with respect to tasks of higher permeability and lower permeability, respectively.

best with a similarity of 0.661. Also, ChemReasoner produces significantly fewer successful edits that satisfy the target property compared to RL-Guider as shown in Tables 1, 2, and 3. ChatDrug has the lowest similarity, often producing drugs that meet chemical property requirements but are structurally unrelated to the original drug. This highlights that baseline methods exhibit low similarity due to inherent biases, as discussed in Sec. 4.2, whereas RL-Guider overcomes this issue by avoiding reliance on predefined knowledge, instead learning to maintain similarity while achieving target properties through its reward function.

5.2 Ablation Study

In this ablation study, we demonstrate three key properties of RL-Guider—① computational efficiency, ② learning with historical experience, and ③ hyperparameter sensitivity.

Computational Efficiency. We conduct experiments to assess the computational time required for different agents to provide suggestions. The results on the test dataset are presented in Table 4. Specifically, as a reinforcement learning approach, RL-Guider generates suggestions **hundreds of times faster than existing methods**. Even when accounting for the training time, it remains significantly faster than existing approaches. Notably, the training phase can be eliminated once RL-Guider has gained sufficient experience.

Time (s)	ChatDrug	Chem Reasoner	**RL-Guider (Total)**	**RL-Guider (Suggest)**
Mean	5.674	5.147	**2.283**	**0.034**
Std.	0.636	7.447	**0.385**	**0.018**

Table 4: The comparison of suggestion generation time between RL-Guider and the baselines. As RL-Guider is a reinforcement learning agent that requires training, we record both the inference time it takes to generate a prompt (denoted as Suggest) and the time it takes to both train and infer per round (denoted as Total). Obviously, RL-Guider is much more efficient than baselines.

Learning with Historical Experience. We conduct experiments to study whether RL-Guider learns to make more informed and effective decisions as it witnesses and learns from more editing tasks. As shown in Fig. 4, it is clear that as RL-Guider gains more experience with these tasks, its capability to provide valuable suggestions that ultimately lead to successful drug editing by the LLM improves significantly.

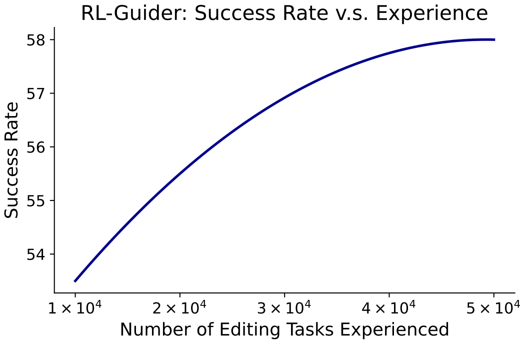

Figure 4: A visualization of the increasing success rate (%) as RL-Guider gains experience with more editing tasks. Clearly, there is a positive trend between the success rate and the number of tasks experienced, demonstrating the effectiveness of RL-Guider and its ability to continuously improve through historical experience.

Hyperparameter Sensitivity. We perform sensitivity analysis to assess the impact of key hyperparameters and reward function components on the performance of RL-Guider. This is because for many machine learning models, the choice of hyperparameters significantly influences the model's performance. Table 5 presents the results of varying the parameter τ on the "More soluble in water" task using DeepSeek. RL-Guider remains robust and performs strongly in all cases except when the value is too extreme. In addition, we evaluate the contribution of each component in the reward function through an ablation study, as shown in Table 6. The results indicate that all components are important, with the target property value being the most significant.

Parameter	0.005	0.01	0.02	0.05
Success Rate (%)	**75.50**	73.50	69.00	12.00

Table 5: Hyperparameter sensitivity analysis of RL-Guider with varying values of τ on the "More soluble in water" task, with $\Delta = 0.5$ using DeepSeek. The best result is bolded. The default setting for τ is 0.005.

Reward Function Setting	**Success Rate (%)**
Base (Full Reward)	**75.50**
w/o Validity	62.50
w/o Similarity	68.50
w/o Target Property	27.50

Table 6: Ablation study on the reward function components of RL-Guider for the "More soluble in water" task, $\Delta = 0.5$ using DeepSeek. The full reward achieves the highest success rate, and each component contributes significantly, especially the target property term.

6 Conclusion and Future Work

In this work, we propose a novel framework, RL-Guider, which leverages a reinforcement learning agent to interact with LLMs and provide valuable suggestions for efficient and effective drug editing. RL-Guider is the first work that learns from the rich feedback obtained through the evaluation of candidate edited drugs; as more editing tasks are performed, RL-Guider accumulates historical experience and progressively improves its recommendation-making capabilities. RL-Guider mitigates several limitations of existing methods, leading to superior performance demonstrated with experiments across various tasks.

Future Work. RL-Guider represents a pioneering approach to reinforcement learning-enhanced drug editing with LLMs, offering vast unexplored potential. For example, exploring different action space designs, including learnable action spaces, could significantly enhance interactions with LLMs. Moreover, RL-Guider can be seamlessly integrated with domain-specific models, such as a "knowledge LLM", where RL-Guider suggests modifications, the LLM provides reasoning on why these suggestions could lead to successful edits, and the main LLM incorporates both the suggestions and reasoning to perform the editing task.

7 Limitations

One limitation is the reliance on predefined substructures (e.g., atoms, functional groups) for actions, which can be vast and hard to design for complex tasks. However, RL-Guider isn't restricted to static actions—it can dynamically expand its action space using LLM-guided suggestions. By generating edits based on chemical knowledge, it allows adaptive learning and refinement of action spaces, though this remains a direction for future work.

Another limitation is the lack of thorough comparison with non-LLM deep learning methods, which typically rely on training data, unlike our LLM-based approach. Still, preliminary results in Appendix E.4 show RL-Guider outperforms two such methods, underscoring the strength of LLM-based strategies. Future work will explore integrating non-LLM advantages to further enhance LLM-guided drug editing.

8 Ethical Considerations

This work advances drug editing with large language models by introducing a novel reinforcement learning agent framework. Although large language models in general may have some ethical considerations, there are limited potential societal and ethical consequences of our work as it focuses on advancing the field of drug discovery, and none require specific highlighting at this time.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. 2021. Geometric and physical quantities improve e (3) equivariant message passing. arXiv preprint arXiv:2110.02905.

Sébastien Bubeck, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2023. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. arXiv preprint arXiv:2311.16208.

Ziqi Chen, Martin Renqiang Min, Srinivasan Parthasarathy, and Xia Ning. 2021. A deep generative model for molecule optimization via one fragment modification. Nature machine intelligence, 3(12):1040–1049.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. arXiv preprint arXiv:2301.00234.

Peter Ertl, Eva Altmann, and Jeffrey M McKenna. 2020. The most common functional groups in bioactive molecules and how their popularity has evolved over time. Journal of medicinal chemistry, 63(15):8408–8418.

Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. 2022. Language models can learn complex molecular distributions. Nature Communications, 13(1):3293.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. Computational Linguistics, 50(3):1097–1179.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning, pages 1861–1870. PMLR.

Hui Huang, Shuangzhi Wu, Xinnian Liang, Bing Wang, Yanrui Shi, Peihao Wu, Muyun Yang, and Tiejun Zhao. 2023. Towards making the most of llm for translation quality estimation. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 375–386. Springer.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. 2012. Zinc: a free tool to discover chemistry for biology. Journal of chemical information and modeling, 52(7):1757–1768.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In International conference on machine learning, pages 2323–2332. PMLR.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. Advances in Neural Information Processing Systems, 33:1179–1191.

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

Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. 2023. Drugchat: towards enabling chatgpt-like capabilities on drug molecule graphs. arXiv preprint arXiv:2309.03907.

Long-Ji Lin. 1992. Reinforcement learning for robots using neural networks. Carnegie Mellon University.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM computing surveys, 55(9):1–35.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2024b. Conversational drug editing using retrieval and domain feedback. In The Twelfth International Conference on Learning Representations.

Xufeng Liu, Dongsheng Luo, Wenhan Gao, and Yi Liu. 2025. 3DGraphX: Explaining 3D molecular graph models via incorporating chemical priors. In Proceedings of the 31th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. 2022. Spherical message passing for 3D molecular graphs. In International Conference on Learning Representations.

Yi Liu, Hao Yuan, Lei Cai, and Shuiwang Ji. 2020. Deep learning of high-order interactions for protein interface prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 679–687.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. 2024. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. arXiv preprint arXiv:2405.09783.

Timothy J O'Donnell, Alex Rubinsteyn, and Uri Laserson. 2020. Mhcflurry 2.0: improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. Cell systems, 11(1):42–48.

Jingxiang Qu, Wenhan Gao, Jiaxing Zhang, Xufeng Liu, Hua Wei, Haibin Ling, and Yi Liu. 2025. RISE: Radius of influence based subgraph extraction for 3d molecular graph explanation. In Proceedings of the 42nd International Conference on Machine Learning.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. Advances in neural information processing systems, 30.

Philipp Seidl, Andreu Vall, Sepp Hochreiter, and Günter Klambauer. 2023. Enhancing activity prediction models in drug discovery with the ability to understand human language. In International Conference on Machine Learning, pages 30458–30490. PMLR.

Henry W Sprueill, Carl Edwards, Khushbu Agarwal, Mariefel V Olarte, Udishnu Sanyal, Conrad Johnston, Hongbin Liu, Heng Ji, and Sutanay Choudhury. 2024. Chemreasoner: Heuristic search over a large language model's knowledge space using quantum-chemical feedback. arXiv preprint arXiv:2402.10980.

Henry W Sprueill, Carl Edwards, Mariefel V Olarte, Udishnu Sanyal, Heng Ji, and Sutanay Choudhury. 2023. Monte carlo thought search: Large language model querying for complex scientific reasoning in catalyst design. arXiv preprint arXiv:2310.14420.

Ronast Subedi, Lu Wei, Wenhan Gao, Shayok Chakraborty, and Yi Liu. 2024. Empowering active learning for 3D molecular graphs with geometric graph isomorphism. In The 38th Annual Conference on Neural Information Processing Systems.

Taffee T Tanimoto. 1958. An elementary mathematical theory of classification and prediction. International Business Machines Corp.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. 2023. Learning hierarchical protein representations via complete 3D graph networks. In The Eleventh International Conference on Learning Representations.

Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. 2022. ComENet: Towards complete and efficient message passing for 3D molecular graphs. In The 36th Annual Conference on Neural Information Processing Systems, pages 650–664.

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis Ioannidis, Karthik Subbian, Jure Leskovec, and James Y Zou.

2025. Avatar: Optimizing llm agents for tool usage via contrastive reasoning. Advances in Neural Information Processing Systems, 37:25981–26010.

Zhenxing Wu, Odin Zhang, Xiaorui Wang, Li Fu, Huifeng Zhao, Jike Wang, Hongyan Du, Dejun Jiang, Yafeng Deng, Dongsheng Cao, et al. 2024. Leveraging language model for advanced multiproperty molecular optimization via prompt engineering. Nature Machine Intelligence, pages 1–11.

Keqiang Yan, Yi Liu, Yuchao Lin, and Shuiwang Ji. 2022. Periodic graph transformers for crystal material property prediction. In The 36th Annual Conference on Neural Information Processing Systems, pages 15066–15080.

Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. 2023. Drugassist: A large language model for molecule optimization. arXiv preprint arXiv:2401.10334.

Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2022. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. Nature communications, 13(1):862.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji. 2023. Artificial intelligence for science in quantum, atomistic, and continuum systems. arXiv preprint arXiv:2307.08423.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, et al. 2024. Chemdfm: A large language foundation model for chemistry. arXiv preprint arXiv:2401.14818.

Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T May, Geoffrey I Webb, Shirui Pan, and George Church. 2024. Large language models in drug discovery and development: From disease mechanisms to clinical trials. arXiv preprint arXiv:2409.04481.

A Detailed Descriptions of Existing Works

To the best of our knowledge, there are very few existing works focusing on LLM-assisted drug editing, including ChatDrug (Liu et al., 2024b), which uses a retrieval database to produce guidance, and ChemReasoner (Sprueill et al., 2024), which uses another pretrained LLM to provide guidance.

ChatDrug (Liu et al., 2024b) identifies the drug most similar to the input drug from the retrieval database that satisfies the desired properties. Subsequently, a suggestion prompt is constructed based on the identified drug to guide the editing.

ChemReasoner (Sprueill et al., 2024) employs another LLM as a planner, assuming it possesses chemical domain knowledge, to generate suggestions that direct the LLM toward more effective candidates. It is noted that the original ChemReasoner framework was designed for catalyst discovery. In our work, we adapt the prompt to address drug editing tasks without modifying its pipeline.

B RL Architecture and Training Details

In this section, we introduce the architecture of RL-Guider and the components of a reward function. **It is worth noting that RL-Guider can be trained without a pre-obtained dataset in an online learning setting.** However, to improve training efficiency, we adopt the offline setting, experience replay (Lin, 1992), with a pre-collected training dataset introduced in Appendix C.1.

B.1 RL Architecture

RL-Guider adopts an actor-critic framework with policy optimization based on the soft actor-critic (SAC) approach (Haarnoja et al., 2018), integrating conservative Q-learning (CQL) for offline learning. The model consists of three sub-networks:

- **Actor Network**: The actor network is a multilayer perceptron (MLP) designed for policy learning. It comprises four fully connected layers with ReLU activation functions, mapping input states to probability distributions over actions. Finally, a linear layer is applied to determine the action sampled from the distribution.

- **Critic Network**: We design two critic networks to estimate state-action values $Q(s, a)$ to mitigate overestimation bias. Each critic network shares the same MLP architecture as the actor. The critic networks are updated using the Bellman backup equation with a soft Q-value function, where γ controls the weights of future expected rewards.

- **Target Network**: The target network decouples the target from the rapidly changing critic network. It helps prevent the feedback loop of self-amplifying errors. In practice, it is initialized as copies of the main critic networks and the target weights at different steps are balanced with the parameter τ.

The output of the network will be fed into the reward function, which is introduced in Sec. 4.1. The resulting rewards and penalties are backpropagated to optimize the model.

B.2 Factors in RL-Guider's Reward Function

To calculate the reward for RL-Guider, we focus on three main factors provided by evaluation feedback: **molecular validity** (σ_t), **structural similarity** (s_m), and **target property** (p_m).

Validity determines whether a molecular structure adheres to the fundamental chemical rules, such as:

- Valency Rules: Atoms must comply with their standard valency constraints.

- Bond Types: Bonds must belong to chemically valid types.

- Connectivity: Atoms must form a connected structure; no disconnected fragments should be present.

In our work, the validity is defined as a binary variable: $\sigma_t \in \{0, 1\}$.

Molecular similarity measures how similar two molecules are based on their structural or chemical properties. Before comparing the similarity of molecules, they must be converted into a numerical representation (e.g., molecular fingerprints). Then, the Tanimoto Similarity (Tanimoto, 1958) can be used to calculate the numerical score between 0 and 1. The Tanimoto Coefficient is defined as:

$$S(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4}$$

where A and B are two binary fingerprint vectors, $|A \cap B|$ is the number of common bits set to 1 in both fingerprints, and $|A \cup B|$ is the number of bits to 1 in either fingerprint. $S(A, B) \in [0, 1]$.

Target property indicates the change in chemical properties toward the desired direction defined in the task prompt.

After obtaining the output SMILES string, all components of the reward function are computed using the cheminformatics tool RDKit (Landrum et al., 2013). In general, an evaluation method should always be available as part of the drug editing pipeline to assess whether the edited drug meets the requirements and can be of use. In this work, we use RDKit for this purpose. RDKit is a widely adopted and lightweight cheminformatics toolkit that introduces minimal computational overhead, making it efficient and scalable for molecule evaluation tasks. Since evaluation is already an integral part of the drug editing process, we do not consider the additional computational complexity and cost introduced by RDKit in our experiments.

C Experimental Setting

C.1 Dataset

ZINC (Irwin et al., 2012) is a publicly accessible database that consolidates commercially available, annotated chemical compounds. Those compounds can be represented in SMILES strings or 3D coordinates. In this work, we only consider SMILES strings as the molecular representation. Following the setting of (Liu et al., 2024b), we randomly extract 200 compounds for evaluation.

The training data of RL-Guider is constructed as a tuple (**input SMILES string**, **action**, **output SMILES string**). Specifically, we extract 100 chemical compounds from the ZINC dataset as the set of **input SMILES strings**. Subsequently, we identify the most common functional groups (the MCFG), as described in Ertl et al. (2020), and define the **action** set based on the three kinds of actions (introduced in Sec. 4.1) applied to MCFG and atoms. The **input SMILES string** and **action** are then incorporated into the prompt (introduced in Appendix D) to generate the **output SMILES string**. Following this process, we obtain $80,000$ edited samples as the training dataset of RL-Guider.

C.2 Baselines and Backbone Models

We evaluate our proposed method against diverse approaches for multi-round conversations with LLMs for drug editing. These include ❶ *Agent-free*: only relies on the intrinsic capabilities of LLMs through iterative multi-round conversations, ❷ *ChatDrug* (Liu et al., 2024b): utilizes a database to provide similar and desirable molecules as guidance, and ❸ *ChemReasoner* (Sprueill et al., 2024): incorporates another LLM to generate suggestions during iterative conversations with the drug editing LLM. To evaluate the generalization ability of our RL-Guider, we conduct experiments on two mainstream LLMs as the backbone models: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and DeepSeek-V3 (Liu et al., 2024a). Both are parameterized with the officially provided weights.

C.3 Training Settings

In the RL-Guider training, the *ChemBERTa-zinc-base-v1* is first employed to convert the SMILES strings into embeddings. The model is then optimized using the Adam optimizer with learning rate set to 3×10^{-3}, and scheduled using cosine annealing. Moreover, an entropy coefficient $\alpha = -1$ is set on the conservative loss to penalize unseen actions, thereby balancing the exploration and exploitation in RL. The hyperparameters τ and γ are set to 0.005 and 0.99, respectively. The model is trained for 10 epochs, sampling $51,200$ tuples per epoch from the training dataset. Additionally, we limit each conversation iteration to a maximum of 30 actions to better learn transition probabilities.

For the drug editing tasks, we assess five molecular properties: **logP** (Octanol-Water Partition Coefficient), **QED** (Quantitative Estimation of Drug-likeness), **tPSA** (Topological Polar Surface Area), **HBA** (Number of Hydrogen Bond Acceptors), and **HBD** (Number of Hydrogen Bond Donors). All tasks are tested on a computational platform equipped with NVIDIA RTX A6000 GPUs.

D Prompt List

In this section, we introduce the prompts used to interact with LLMs in RL training data acquisition, drug editing, and comparison methods. First, the working environment of LLM is set with the system prompt:

> System Prompt
>
> You are a helpful chemistry expert with extensive knowledge of drug design.

D.1 RL Training Data Acquisition Prompt

To obtain the training data for RL-Guider, we sampled some edited molecules from LLMs using predefined action sets. The prompt is designed as:

Prompt for RL Data Acquisition

Edit the molecule {mol} by following the suggestion: {suggestion}. Please provide five molecules in SMILES format only and list them using bullet points.

{mol}: Input molecule for LLM to edit.

{suggestion}: Sampled action from a predefined action set.

D.2 Drug Editing Prompt

Our drug editing experiments consist of two scenarios: ❶ single-round editing, which involves a single interaction with the LLM, and ❷ multi-round editing, which involves multiple interactions. The prompt for each scenario is introduced as follows.

Single-round Editing Prompt

Prompt for Single-round Editing

Can you make molecule {root_mol} {task_objective} and {threshold_specific_prompt}? The output molecule should be similar to the input molecule. {suggestion}. Please provide five molecules in SMILES format only and list them using bullet points. {reasoning_instruction}.

Multi-round Editing Prompt

Prompt for Round 1 Editing

Can you make molecule {root_mol} {task_objective} and {threshold_specific_prompt}? The output molecule should be similar to the input molecule. Please provide five molecules in SMILES format only and list them using bullet points. {reasoning_instruction}.

Prompt for Round X Editing ($X \geq 2$)

Your provided sequence {prev_wrong_mol} could not achieve the goal. {suggestion}. Can you give me new molecules?

{root_mol}: Input molecule for LLM to edit.

{task_objective}: As shown in Table 7.

{threshold_specific_prompt}: As shown in Table 7.

{suggestion}: Suggestion for LLM to conduct drug editing and can be filled with different formats under different methods. In the Agent-free method, this field is empty. For ChemReasoner, the suggestion is parsed from planner LLM's result. For Chatdrug, we utilize part of its ReDF prompt (Liu et al., 2024b) to serve as the suggestion. For RL-Guider, the suggestion is predicted by the actor network. An example suggestion is displayed in Appendix. D.1.

{prev_wrong_mol}: Previously edited molecule that could not satisfy the query.

{reasoning_instruction}: Specific requirement for LLMs, including options like "no explanation is needed" and "let's think step by step".

D.3 LLM Planner (ChemReasoner) Prompt

Since ChemReasoner (Sprueill et al., 2024) was originally designed for catalyst discovery tasks, we adapt its core methodology and apply it to the field of drug editing, with modifications tailored to our specific task.

Prompt for LLM Planning Suggestion

$root_question: {root_prompt}
$root_property: {root_property}
$threshold: {threshold}
{previous_prompt_answer}
Consider the {current_conditions}. Your task is to suggest possible actions that could achieve the intent of $root_question.
$search_state: Molecule to be optimized, as specified in the message.
$action_space: Add, delete, or replace an atom or functional group.
{guidelines}
{final_task}

{root_prompt}: Root query including input molecule, task objective, and instruction.

{root_property}: The quantitative property of the input molecule.

{threshold}: Specifies the amount by which the input molecule needs to be optimized.

{previous_prompt_answer}: Empty in a single-round conversation or in round 1. Filled with the latest conversation history.

{current_conditions}: Specifies the factors the LLM needs to consider. Changes based on the

`{task_objective}`	**Threshold**	`{threshold_specific_prompt}`
More soluble in water	loose	decrease logP by at least 0
	strict	decrease logP by at least 0.5
Less soluble in water	loose	increase logP by at least 0
	strict	increase logP by at least 0.5
More like a drug	loose	increase QED by at least 0
	strict	increase QED by at least 0.1
Less like a drug	loose	decrease QED by at least 0
	strict	decrease QED by at least 0.5
Higher Permeability	loose	decrease tPSA by at least 0
	strict	decrease tPSA by at least 0.5
Lower Permeability	loose	increase tPSA by at least 0
	strict	increase tPSA by at least 10
More hydrogen bond acceptors	loose	increase HBA by at least 0
	strict	increase HBA by at least 1
More hydrogen bond donors	loose	increase HBD by at least 0
	strict	increase HBD by at least 1
More soluble in water and more hydrogen bond acceptors	loose	decrease logP by at least 0 and increase HBA by at least 0
	strict	decrease logP by at least 0.5 and increase HBA by at least 1
Less soluble in water and more hydrogen bond acceptors	loose	increase logP by at least 0 and increase HBA by at least 0
	strict	decrease logP by at least 0.5 and increase HBA by at least 1
More soluble in water and more hydrogen bond donors	loose	decrease logP by at least 0 and increase HBD by at least 0
	strict	decrease logP by at least 0.5 and increase HBA by at least 1
Less soluble in water and more hydrogen bond donors	loose	increase logP by at least 0 and increase HBD by at least 0
	strict	increase logP by at least 0.5 and increase HBD by at least 1
More soluble in water and higher permeability	loose	decrease logP by at least 0 and decrease tPSA by at least 0
	strict	decrease logP by at least 0.5 and decrease tPSA by at least 10
More soluble in water and lower permeability	loose	decrease logP by at least 0 and increase tPSA by at least 0
	strict	decrease logP by at least 0.5 and increase tPSA by at least 10

Table 7: Detailed information on `{task_objective}` and `{threshold_specific_prompt}` in prompt template.

content in `{previous_prompt_answer}`.

`{guidelines}`: Hint for the LLM to consider when reasoning about its suggestions.

`{final_task}`: Instruction for the LLM to generate a reasonable and parsable answer.

Below is an example of an LLM Planner Prompt:

$root_question: Can you make molecule C1CC[C@@H](CCc2nc([C@@H]3CSCCO3)no2)-

Multiple Target Property	Δ	LLaMA				DeepSeek			
		Agent Free	Chat Drug	Chem Reasoner	RL-Guider	Agent Free	Chat Drug	Chem Reasoner	RL-Guider
More soluble in water	0, 0	43.00	44.00	37.00	50.00	65.50	53.00	75.00	77.00
More hydrogen bond acceptors	0.5, 1	8.50	34.00	8.00	37.50	43.50	41.00	28.50	63.00
Less soluble in water	0, 0	12.50	34.00	21.00	37.50	56.00	50.50	27.00	63.00
More hydrogen bond acceptors	0.5, 1	6.00	31.00	7.00	24.00	4.00	39.50	3.00	6.00
More soluble in water	0, 0	40.50	22.50	43.00	45.00	72.50	76.50	72.00	71.50
More hydrogen bond donors	0.5, 1	15.00	12.50	11.50	20.00	44.00	60.00	23.50	68.50
Less soluble in water	0, 0	17.50	14.00	22.00	23.00	14.50	10.50	16.00	9.50
More hydrogen bond donors	0.5, 1	6.00	8.00	5.00	8.50	1.50	6.50	1.50	10.50
More soluble in water	0, 0	4.00	23.50	3.00	8.00	6.00	20.50	5.50	8.00
Higher permeability	0.5, 10	0.50	17.50	2.00	3.00	2.50	3.00	4.50	5.50
More soluble in water	0, 0	44.50	46.00	51.50	44.00	71.50	54.50	74.50	79.00
Lower permeability	0.5, 10	28.00	30.50	33.50	38.50	57.00	43.00	52.50	73.00

Table 8: Quantitative comparison of the success rate (%) based on LLaMA and DeepSeek for multi-round interactions in multi-property editing tasks, where multiple desired property changes are achieved simultaneously. The best and second-best results are highlighted in red and blue, respectively. Clearly, our RL-Guider outperforms baseline methods in most scenarios.

[NH2+]C1 more soluble in water and increase the number of hydrogen bond donors, decrease logP by at least 0.5, and increase HBD by at least 1? The output molecule should be similar to the input molecule. Please provide five molecules in SMILES format only and list them using bullet points. No explanation is needed. $root_property: The logP of the root molecule is 0.9226. The HBD of the root molecule is 1. $threshold: You should decrease the logP by more than 0.5 and increase the HBD by more than 1. Consider the $root_question, $root_property, $threshold. Your task is to suggest possible actions that could achieve the intent of the $root_question. $search_state: C1CC[C@@H](CCc2nc([C@@H]3CSCCO3)no2)-[NH2+]C1 . $action_space: Add, delete, replace an atom or functional group. Your answers should use the following guidelines: 1. You should return a python list named final_suggestion, which contains the top-1 suggestion based on the previous information. 2. You should learn from the previous experience, especially the substructural changes in molecules. 3. Your suggestion should not repeat the previous suggestion in $previous_prompt. 4. In your suggestion, please do not use any abbreviation of an atom or functional group. For example, when you need to show "hydroxyl group", do not show "(OH)" in your suggestion! 5. Each of your suggestions should be a sentence of modification instruction rather than a SMILES string. 6. Please note that your suggestion should also consider the similarity before and after modification. Take a deep breath and let's think about the goal and guidelines step by step. Remember, you should give your reasoning process first and finally return a Python list named final_suggestion!

E Additional Experimental Results

E.1 Multi-property Tasks

In addition to single-property drug editing, we also conduct experiments on multi-property editing tasks, which require the edited drug to simultaneously satisfy multiple properties, which makes them more challenging tasks. The results are presented in Table 8. Clearly, RL-Guider demonstrates superior performance compared to baseline models.

Source Allele Type	Target Allele Type	ChatDrug	RL-Guider
HLA-C*16:01	HLA-B*44:02	38.50	**59.00**
HLA-C*12:02	HLA-B*40:01	32.50	**36.50**

Table 9: Quantitative comparison of success rate (%) using LLaMA with multi-round interactions on peptide editing tasks. The best results are shown in bold. Obviously, RL-Guider outperforms ChatDrug on the challenging peptide editing tasks.

	ChatDrug	Chem Reasoner	**RL-Guider**
Similarity	0.518	0.661	**0.718**

Table 10: A comparison of the similarity between the input drug and output edited drug by various methods. The mean value of similarity is reported (the closer to 1, the better).

E.2 Performance on Peptides

To evaluate the effectiveness of our RL-Guider on larger and more complex molecular structures, we have additionally performed experiments using MHCFlurry (O'Donnell et al., 2020), a benchmark

δ	JT-VAE			RL-Guider		
	Imp.	Similarity	Success Rate (%)	Imp.	Similarity	Success Rate (%)
0.0	1.91 ± 2.04	0.30 ± 0.18	81.50	$\mathbf{1.89 \pm 2.30}$	$\mathbf{0.52 \pm 0.29}$	**91.53**
0.2	1.64 ± 1.85	0.35 ± 0.17	79.88	$\mathbf{1.70 \pm 2.09}$	$\mathbf{0.59 \pm 0.26}$	**89.82**
0.4	0.55 ± 1.21	0.62 ± 0.21	55.00	$\mathbf{1.33 \pm 2.07}$	$\mathbf{0.71 \pm 0.21}$	**84.80**
0.6	0.10 ± 0.50	0.86 ± 0.16	30.00	$\mathbf{0.66 \pm 1.50}$	$\mathbf{0.83 \pm 0.15}$	**72.81**

Table 11: Comparison with Junction Tree (JT-VAE) on the ZINC dataset using the logP property. All results follow JT-VAE's original evaluation protocol. The result for RL-Guider is generated using DeepSeek as the LLM. δ denotes the minimum required similarity between the input and output molecules. "Imp." refers to the improvement in property value. For both "Imp." and "Similarity", we report the mean ± standard deviation, while "Success Rate" is reported as a percentage. The best results in each row are bolded. Notably, our RL-Guider achieves the highest success rate across all evaluation settings when compared with Junction Tree.

focused on immunogenic peptide binding optimization. In this setting, a successful edit must satisfy the following three criteria: (1) The resulting peptide must be valid; (2) The output peptide should exhibit a higher binding affinity with the target allele than the input peptide; (3) The binding affinity between the output peptide and the target allele must be above a certain threshold, which is set to be one-half of the average binding affinity of experimental data on the target allele, following the same setting as in Liu et al. (2024b). The success rates of RL-Guider and ChatDrug, both using LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), under multi-round interactions are reported in Table 9. Other baselines perform particularly poorly on this task, with a success rate below 5%, likely due to the difficulty of the task. Therefore, we do not include them here. RL-Guider outperforms ChatDrug on this task, demonstrating its capability in peptide editing tasks. These results highlight the potential of RL-Guider for handling more complex drug editing tasks. Further exploration in this direction is left for future work.

E.3 Case Study of Similarity

In addition to the visualizations in Fig. 3 in the main paper, the average similarity across the entire testing dataset achieved by different methods is presented in Table 10. RL-Guider achieves the highest similarity among all methods. This is because RL-Guider does not suffer from the inherent bias from predefined knowledge as discussed in Sec. 4.2 in the main paper.

E.4 Comparison with Non-LLM Approaches

A preliminary study is conducted to evaluate our method against non-LLM-based approaches. A key distinction between our work and non-LLM models is that non-LLM models require a training dataset, while our method does not require any training data beforehand. However, our work leverages the "world-level" knowledge of LLMs and an agent that accumulates knowledge through historical decisions. Meanwhile, non-LLM methods are typically trained on a specific dataset, such as ZINC, and then used to make inferences on the similar data. This practice could introduce bias into the trained model, artificially improving performance on that dataset. In contrast, our method uses the dataset solely as a benchmark for evaluation and comparison with baselines. Our method can generalize to other datasets, whereas non-LLM methods often require retraining.

	Prompt-MolOpt	**RL-Guider**
Success Rate (%)	50.00	**75.00**

Table 12: Comparison with Prompt-MolOpt on a subset of the original dataset. The result for RL-Guider is generated using DeepSeek. In Prompt-MolOpt, the editing site must be predefined by human experts or external models. We evaluate only on the subset with available editing sites due to time constraints. The best result is bolded. Obviously, RL-Guider outperforms Prompt-MolOpt.

We provide comparisons with two representative methods: Junction Tree (Jin et al., 2018) and Prompt-MolOpt (Wu et al., 2024). Junction Tree is an end-to-end generative model, and Prompt-MolOpt is a language model (non-LLM) trained on drug-editing data. As shown in Table 11, RL-Guider achieves the highest success rate across all evaluation settings when compared with Junction Tree. Notably, under high similarity constraints, RL-Guider significantly outperforms Junction Tree (72.81% vs. 30.00% success rate). Similarly, in the comparison with Prompt-MolOpt presented in Table 12, RL-Guider demonstrates superior performance (75.00% vs. 50.00% success rate). These results clearly demonstrate the effectiveness and potential of our approaches over traditional models.