

# ChatGPT-powered Conversational Drug Editing Using Retrieval and Domain Feedback

Shengchao Liu<sup>1,2\*</sup>, Jiongxiao Wang<sup>3\*</sup>, Yijin Yang<sup>3</sup>, Chengpeng Wang<sup>4</sup>, Ling Liu<sup>5</sup>, Hongyu Guo<sup>6†</sup>, and Chaowei Xiao<sup>3†</sup>

<sup>1</sup>Mila-Québec Artificial Intelligence Institute, Montréal, QC H2S 3H1, Canada

<sup>2</sup>Université de Montréal, Montréal, QC H3T 1J4, Canada

<sup>3</sup>Arizona State University, Tempe, AZ 85281, United States

<sup>4</sup>Princeton University, Princeton, NJ 08544, United States

<sup>5</sup>University of Illinois Urbana-Champaign, Champaign, IL 61801, United States

<sup>6</sup>National Research Council Canada, Ottawa, ON K1A 0R6, Canada

## ABSTRACT

Recent advancements in conversational large language models (LLMs), such as ChatGPT, have demonstrated remarkable promise in various domains, including drug discovery. However, existing works mainly focus on investigating the capabilities of conversational LLMs on chemical reaction and retrosynthesis. While drug editing, a critical task in the drug discovery pipeline, remains largely unexplored. To bridge this gap, we propose ChatDrug, a framework to facilitate the systematic investigation of drug editing using LLMs. ChatDrug jointly leverages a prompt module, a retrieval and domain feedback (ReDF) module, and a conversation module to streamline effective drug editing. We empirically show that ChatDrug reaches the best performance on 33 out of 39 drug editing tasks, encompassing small molecules, peptides, and proteins. We further demonstrate, through 10 case studies, that ChatDrug can successfully identify the key substructures (*e.g.*, the molecule functional groups, peptide motifs, and protein structures) for manipulation, generating diverse and valid suggestions for drug editing. Promisingly, we also show that ChatDrug can offer insightful explanations from a domain-specific perspective, enhancing interpretability and enabling informed decision-making. This research sheds light on the potential of ChatGPT and conversational LLMs for drug editing. It paves the way for a more efficient and collaborative drug discovery pipeline, contributing to the advancement of pharmaceutical research and development. The source codes can be found in [this GitHub repository](#).

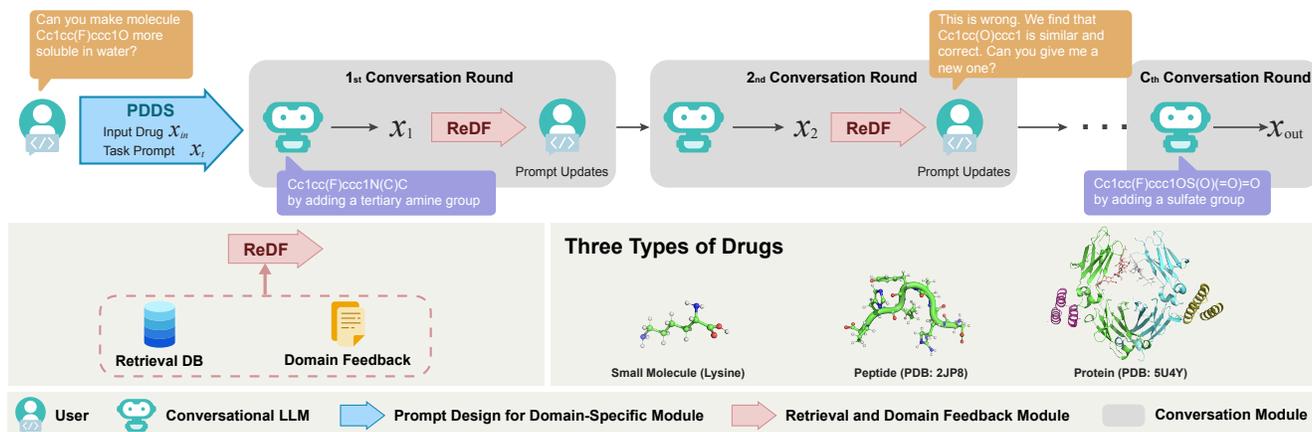
## 1 Introduction

In recent years, artificial intelligence (AI) tools have made remarkable strides in revolutionizing the field of drug discovery, offering tremendous potential for accelerating and enhancing various stages of the process [1], including but not limited to virtual screening [2, 3], lead optimization [4, 5, 6, 7], reaction and retrosynthesis [8, 9, 10], protein folding and inverse folding [11, 12]. However, much of the existing research has predominantly focused on the drug structure information, solely considering the inherent chemical structure of the drugs as a single modality. In contrast, the drug discovery pipeline involves iterative refining processes that entail conversations with domain experts to incorporate their feedback, ultimately achieving the desired outcome. On the other hand, significant advancements have been made in large language models (LLMs) [13, 14, 15], showcasing exceptional capabilities in understanding human knowledge and exhibiting promising reasoning abilities [16, 17, 18]. Such observations inspire us to investigate the potential of leveraging LLMs’ conversation and reasoning abilities for AI-assisted drug discovery in a multi-modality fashion.

**Potential of Conversational LLMs for Drug Discovery and Editing.** Conversational LLMs exhibit three compelling factors that make them highly promising for drug discovery. Firstly, these models, such as ChatGPT, are pretrained on a comprehensive knowledge base, enabling their application across various fields, including drug discovery. This extensive “world-level” knowledge serves as a robust foundation for drug-related tasks. Second, conversational LLMs possess outstanding abilities in fast adaptation and generalization. For example, by leveraging few-shot demonstrations, these models can effectively activate the relevant concepts learned during pretraining, enabling them to deliver more accurate and desired answers [19]. This adaptability and generalization capacity holds immense potential for addressing complex drug discovery challenges and generating valuable insights. Lastly, interactive communication is a vital characteristic of conversational LLMs. This feature allows for a dynamic exchange of information, enabling users to incorporate feedback from prior knowledge or domain experts

\*Equal contribution

†Equal advising



**Figure 1.** The pipeline for ChatDrug with 3 modules. PDDS generates drug editing prompts. ReDF updates the prompts using retrieved information and domain feedback. Finally, ChatDrug adopts the conversational module for interactive refinement. Further, we demonstrate 3 drug types: small molecules, peptides, and proteins.

into the model. This bidirectional flow of information facilitates self-calibration of the answers, leading to improved accuracy and relevance in drug discovery tasks. To sum up, these factors collectively highlight the untapped potential of conversational LLMs for drug discovery tasks. Noticeably, there exists an important and challenging task: **drug editing** (AKA *lead optimization* or *protein design*). This is a routine task in pharmaceutical companies, and it aims at updating the drug’s substructures [20], related to certain key tactics in drug discovery like functional group change [21] and scaffold hopping [22, 23]. Traditional solutions relying on domain experts for manual editing can be subjective or biased [24, 25]. Recent works [26, 27] have started to explore text-guided drug editing in a multi-modal manner. However, they do not possess conversational potentials like ChatGPT.

**Our Approach: ChatDrug.** Motivated by the aforementioned factors and challenges, we propose ChatDrug, a framework aiming to unlock new possibilities and enhance drug editing using contrastive LLMs like ChatGPT. As shown in Figure 1, in the ChatDrug framework for drug editing, users can activate the conversation with LLMs involving domain knowledge and inject such retrieved information into the conversation. Specifically, ChatDrug includes the following modules for conversational drug editing. First, ChatDrug adopts a PDDS (prompt design for domain-specific) module, enabling strong prompt engineering capability from LLMs. Second, ChatDrug integrates a ReDF (retrieval and domain feedback) module. By leveraging the vast domain knowledge available, such a ReDF module serves as guidance for prompt updates and augments the model’s performance in generating accurate outputs. Third, ChatDrug adopts a conversation-based approach, aligning with the iterative refinement nature of the drug discovery pipeline. Such interactive schema enables a dynamic and collaborative process, effectively incorporating feedback from domain experts to achieve desired outcomes.

Through our design, ChatDrug demonstrates two appealing properties for drug editing tasks: (1) ChatDrug exhibits an open vocabulary property, allowing for exploring novel drug concepts beyond a fixed set of pre-defined annotations. The model can generalize to new drug-related concepts due to the unrestricted nature of natural language. (2) ChatDrug possesses the compositional property. It can decompose complex concepts, such as multi-objective lead optimization, into simpler attributes like binding to a new protein and high permeability, enabling handling complicated drug editing tasks.<sup>1</sup>

Then to fully verify the effectiveness of ChatDrug, we need to design a benchmark for a wide range of drug editing tasks. Before going into details, we would like to claim two criteria for the task design: (1) The tasks should involve indeterministic answers, as they can serve as a source of inspiration for domain experts. (2) The tasks should be able to evaluate computationally since the lab experiment can be quite laborious and is beyond the discussion of this paper. Following these criteria, we introduce 39 editing tasks over three common drugs: 28 for small molecules, 9 for peptides, and 2 for proteins.

Last but not least, we offer empirical evidence substantiating the capability of ChatDrug for a wide range of drug editing tasks. Quantitatively, ChatDrug can reach the best performance on 33 out of 39 drug editing tasks compared to seven baselines. Qualitatively, we further provide 10 case studies (more qualitative results in Appendix G), illustrating that ChatDrug can successfully identify the important substructures for each type of drug, as follows. (1) For small molecules, ChatDrug is able to detect the key scaffold for molecule editing, such as changing polar or hydrophobic functional groups for tuning properties like solubility in water and permeability. (2) For peptides, ChatDrug accurately identifies the protein-specific binding motifs of the

<sup>1</sup>Note that ChatDrug aims to inspire domain experts rather than replace them. While ChatDrug can propose optimized drugs or novel attributes, its primary role is to serve as a tool for knowledge exploration. The generated outputs can provide valuable insights and spark inspiration for domain experts in the drug discovery process.

peptide sequences. (3) For proteins, ChatDrug can modify sequences with more  $\alpha$ -helix or  $\beta$ -strand structures after folding [11, 28]. We additionally illustrate that ChatDrug can provide insightful explanations, serving as a knowledge extraction tool.

## 2 Preliminaries

**Data Structure of Drugs.** Drugs [29, 30] refer to certain specific substances that can be adopted to prevent, diagnose, treat, or relieve symptoms of a disease or abnormal condition. In this paper, we would like to explore the three most common drugs: small molecules [31], proteins [32], and peptides [33]. Small molecules are sets of atoms connected together through the covalent bonds. Commonly-used data structures include SMILES (simplified molecular-input line-entry system) strings [34] and molecular graphs [35, 36, 37]. In ChatDrug, we consider using the SMILES strings. Proteins are complex macromolecules, and they are composed of 20 amino acids, where each amino acid is a small molecule. Regarding the protein data structure, we adopt the amino acid sequence (*i.e.*, amino acid string), and the mapping between 20 alphabets and amino acids can be found in Appendix D. Peptides are short chains of amino acids and can be viewed as a special type of protein. The demonstration of three data structures can be found in Figure 1.

**Drug Editing and Problem Formulation.** In this paper, we focus on the drug editing task. Drug editing is also known as *lead optimization* or *protein design*, an important drug discovery task. From the machine learning perspective, drug editing is a *conditional generation* problem and can be formulated as follows. Suppose the input drug (SMILES string or amino acid sequence) is  $\mathbf{x}_{in}$ , and a target or desired property in the textual description is also known as the *text prompt*  $\mathbf{x}_t$  in literature [38, 39]. Then condition on such text prompt, the goal is to obtain an optimized drug as:

$$\mathbf{x}_{out} = \text{ChatDrug}(\mathbf{x}_{in}, \mathbf{x}_t). \quad (1)$$

Then an evaluation metric  $E(\mathbf{x}_{in}, \mathbf{x}_{out}; \mathbf{x}_t) \in \{\text{True}, \text{False}\}$  is to check if the edited drugs can satisfy the desired properties compared to the input drugs, and we will average this over each corresponding task to get the *hit ratio*. Note that  $E(\cdot, \cdot; \cdot)$  is task-specific, as will be discussed in Section 4.

## 3 Method: ChatDrug Framework

**Overview.** Our framework is shown in Figure 1. ChatDrug consists of three components: (1) Prompt Design for Domain-Specific (PDDS) module, (2) Retrieval and Domain Feedback (ReDF) module, and (3) conversation module. Given a task prompt and input drug, PDDS aims to generate the domain-specific prompt and concatenate it with the input drug to request ChatGPT for answers. One problem for current LLMs is that it does not fully utilize the prior domain knowledge. Thus, we design the ReDF module aiming to (1) guide the LLMs to solve this task by retrieving structurally similar examples from the database and adding examples into the prompt as demonstrations and (2) verify the correctness of the output by using a domain feedback function. If the output drug is incorrect after ReDF, we then adopt the conversation module to ask LLMs to generate a new drug iteratively. Note that ChatDrug is a parameter-free scheme and does not require any learning procedure.

### 3.1 PDDS Module

ChatDrug is proposed to solve a challenging problem: generalization of a universally (w.r.t. data type and data source) well-trained LLM to solving scientific tasks. In natural language processing (NLP), prompt design or prompt engineering [39] has proven to be an effective paradigm for generalizing well-trained LLMs to various NLP downstream tasks, including but not limited to sentiment classification [40, 41], textual entailment [42, 43], text summarization [44, 45, 46].

But the explorations of adapting ChatGPT for drug editing tasks have been lagging behind. In this paper, we are interested in investigating this problem on the three most common types of drugs: small molecules, protein-binding peptides, and proteins. Recall that the goal of ChatDrug is (as in Equation (1)):  $\mathbf{x}_{out} = \text{ChatDrug}(\mathbf{x}_{in}, \mathbf{x}_t)$ . Here the text prompts  $\mathbf{x}_t$  should be specifically designed to enable the generalization for domain-specific tasks with computationally feasible metrics. Additionally, we want to highlight that the objectives for drug editing (in  $\mathbf{x}_t$ ) should be about the *high-level property* instead of *exact substructure replacement*. There are two main reasons, as follows. (1) As discussed in Appendix B, ChatDrug suits better for fuzzy matching like edited drugs with desired properties. In contrast, exact substructure replacement can be easily and precisely performed by domain experts, and such replacement may lack the creative inspiration for humans. (2) Property-related questions have an ambiguous nature, leading to dispersed answers that spark inspiration for domain experts in the drug discovery process.

Then concretely on the prompt design, for small molecules, we consider properties like solubility, drug-likeness, permeability, and the number of acceptors/donors. For peptides, we consider the properties of peptide-MHC binding. For proteins, we consider the secondary structure. The text prompts are to explicitly depict the desired properties to be either higher or lower, and corresponding task prompts will be briefly explained in Section 4. One concrete example for molecule editing is “*Can you make molecule  $[\mathbf{x}_{in}]$  more soluble in water.*”, and more details can be found in Appendix E.

### 3.2 ReDF Module

To better utilize the domain knowledge, we propose an important module: the ReDF (retrieval and domain feedback) module. The intuition is that there exists rich domain knowledge in the form of a retrieval database (DB), and ReDF will retrieve the useful information and inject it into the text prompt, adopting the fascinating language understanding ability of conversational LLMs.

Specifically, for each input drug  $\mathbf{x}_{in}$  and prompt  $\mathbf{x}_t$ , we have a candidate drug  $\tilde{\mathbf{x}}$ , which does not satisfy the desired property change in  $\mathbf{x}_t$ . The candidate drug has multiple data resources, depending on the problem setup; in ChatDrug, it is the output drug with the negative result at each conversation round (will be introduced in Section 3.3). Based on these, ReDF will return a drug  $\mathbf{x}_R$  satisfying:

$$\mathbf{x}_R = \text{ReDF}(\mathbf{x}_{in}, \tilde{\mathbf{x}}; \mathbf{x}_t) = \arg \max_{\mathbf{x}'_R \in \text{Retrieval DB}} \langle \tilde{\mathbf{x}}, \mathbf{x}'_R \rangle \wedge D(\mathbf{x}_{in}, \mathbf{x}'_R; \mathbf{x}_t), \quad (2)$$

where  $D(\cdot, \cdot; \cdot) \in \{\text{True}, \text{False}\}$  is the domain feedback function, and  $\langle \tilde{\mathbf{x}}, \mathbf{x}'_R \rangle$  is the similarity function. We use Tanimoto similarity [47] for small molecules and Levenshtein distance for peptides and proteins. Notice that here we take  $D(\cdot, \cdot; \cdot)$  the same as evaluation metric  $E(\cdot, \cdot; \cdot)$ , while there is some critical difference on the task-specific thresholds, as will be discussed in the ablation study in Section 4.6. Then the ReDF module injects  $\mathbf{x}_R$  into a new prompt, e.g., the updated prompt for a molecule task is "Your provided sequence [ $\tilde{\mathbf{x}}$ ] is not correct. We find a sequence [ $\mathbf{x}_R$ ] which is correct and similar to the molecule you provided. Can you give me a new molecule?"

We also want to highlight that the domain feedback injection in ReDF is similar to the *in-context learning* (ICL) paradigm [48]. Such knowledge injection can result in performance gain [49] not only because of the mapping between ground truth data-label pairs, but also the format or demonstration of the in-distribution data and label space. In Section 4, we will conduct an ablation study on ICL.

### 3.3 Conversation Module

Another appealing attribute of conversational LLMs (like ChatGPT) is their interactive capability. This enables the LLMs to iteratively update the results by injecting prior knowledge. Inspired by this, we also consider adapting the conversational strategy for ChatDrug, which can naturally fit the ReDF module as described in Section 3.2. Then concretely on this conversational strategy in ChatDrug, first suppose there are  $C$  conversation rounds, and we have an edited drug  $\mathbf{x}_c$  for the conversation round  $c$ . If  $\mathbf{x}_c$  satisfies our condition in the task prompt, then ChatDrug will exit. Otherwise, users will tell ChatDrug that  $\mathbf{x}_c$  is wrong, and we need to retrieve another similar but correct drug from the retrieval DB using ReDF:  $\mathbf{x}_R = \text{ReDF}(\mathbf{x}_{in}, \mathbf{x}_c)$ , with  $\tilde{\mathbf{x}} = \mathbf{x}_c$  in Equation (2).

To sum up, for each conversation round, we request a drug  $\mathbf{x}_R$  similar to  $\mathbf{x}_c$ , which will be updated at each conversation round. The  $\mathbf{x}_c$  and  $\mathbf{x}_R$  serve as two in-context pairs to feed into ChatDrug, i.e., "The output drug at round [ $c$ ] is [ $\mathbf{x}_c$ ], which is wrong. We find a sequence [ $\mathbf{x}_R$ ] which is correct and similar. Can you help improve the edited results?" An illustration of this conversation is in Figure 1.

## 4 Experiment

**Specifications for ChatDrug.** In this section, we verify the effectiveness of ChatDrug for drug editing on three types of drugs: small molecules, peptides, and proteins. Here we select GPT-3.5 in our experiment. We introduce three types of drugs and five categories of tasks accordingly: task 1xx and 2xx are single- and multi-objective tasks for small molecules (each task further includes 2 subtasks w.r.t. two thresholds as will be discussed next), task 3xx and 4xx are single- and multi-objective editing tasks for peptides, and task 5xx is for single-objective protein editing. Due to the space limitation, please check Appendix E for the full list. Details of implementation and hyperparameters are in Appendix F.

### 4.1 Text-guided Molecule Property Editing

The first experiment is text-guided molecule editing or molecule optimization. We adopt 16 single-objective tasks and 12 multi-objective editing tasks from MoleculeSTM [27]. These tasks are about the high-level properties of small molecules, like solubility in water and permeability.

**Data:** Both the input molecules and retrieval DB are sampled from ZINC [50]: we sample 200 and 10K molecules (with SMILES strings) from ZINC as input molecules and retrieval DB, respectively. **Prompt:** The text prompt is "Can you make molecule [SMILES placeholder] [task requirement]? The output molecule should be similar to the input molecule". The [task requirement] is the textual description for each specific task, e.g., *more soluble in water* and *with higher permeability*.

**Evaluation.** We take the hit ratio to measure the success ratio of edited molecules, i.e., the percentage of edited molecules that can reach the desired properties compared to the input molecules. All the properties for small molecules considered here can be calculated deterministically using RDKit [51]. Another important argument is the threshold  $\Delta$ : it is a successful hit if

**Table 1.** Results on 16 single-objective small molecule editing, and the evaluation is the hit ratio of the property change. For ChatDrug, we report the mean and std of five random seeds. The best results are marked in **bold**.

Single Target Property	$\Delta$	Random	PCA	High Variance	GS-Mutate	MoleculeSTM (SMILES)	MoleculeSTM (Graph)	ChatDrug (Ours)
101 <i>more soluble in water</i>	0	35.33 $\pm$ 1.31	33.80 $\pm$ 3.63	33.52 $\pm$ 3.75	52.00 $\pm$ 0.41	61.87 $\pm$ 2.67	67.86 $\pm$ 3.46	<b>94.13<math>\pm</math>1.04</b>
	0.5	11.04 $\pm$ 2.40	10.66 $\pm$ 3.24	10.86 $\pm$ 2.56	14.67 $\pm$ 0.62	49.02 $\pm$ 1.84	54.44 $\pm$ 3.99	<b>88.67<math>\pm</math>0.95</b>
102 <i>less soluble in water</i>	0	43.36 $\pm$ 3.06	39.36 $\pm$ 2.55	42.89 $\pm$ 2.36	47.50 $\pm$ 0.41	52.71 $\pm$ 1.67	64.79 $\pm$ 2.76	<b>96.86<math>\pm</math>1.10</b>
	0.5	19.75 $\pm$ 1.56	15.12 $\pm$ 2.93	18.22 $\pm$ 0.33	12.50 $\pm$ 0.82	30.47 $\pm$ 3.26	47.09 $\pm$ 3.42	<b>70.08<math>\pm</math>3.44</b>
103 <i>more like a drug</i>	0	38.06 $\pm$ 2.57	33.99 $\pm$ 3.72	36.20 $\pm$ 4.34	28.00 $\pm$ 0.71	36.52 $\pm$ 2.46	39.97 $\pm$ 4.32	<b>48.65<math>\pm</math>3.39</b>
	0.1	5.27 $\pm$ 0.24	3.97 $\pm$ 0.10	4.44 $\pm$ 0.58	6.33 $\pm$ 2.09	8.81 $\pm$ 0.82	14.06 $\pm$ 3.18	<b>19.37<math>\pm</math>5.54</b>
104 <i>less like a drug</i>	0	36.96 $\pm$ 2.25	35.17 $\pm$ 2.61	39.99 $\pm$ 0.57	71.33 $\pm$ 0.85	58.59 $\pm$ 1.01	77.62 $\pm$ 2.80	70.75 $\pm$ 2.92
	0.1	6.16 $\pm$ 1.87	5.26 $\pm$ 0.95	7.56 $\pm$ 0.29	27.67 $\pm$ 3.79	37.56 $\pm$ 1.76	<b>54.22 <math>\pm</math> 3.12</b>	30.99 $\pm$ 2.66
105 <i>higher permeability</i>	0	25.23 $\pm$ 2.13	21.36 $\pm$ 0.79	21.98 $\pm$ 3.77	22.00 $\pm$ 0.82	57.74 $\pm$ 0.60	<b>59.84 <math>\pm</math> 0.78</b>	56.56 $\pm$ 1.84
	10	17.41 $\pm$ 1.43	14.52 $\pm$ 0.80	14.66 $\pm$ 2.13	6.17 $\pm$ 0.62	47.51 $\pm$ 1.88	<b>50.42 <math>\pm</math> 2.73</b>	43.08 $\pm$ 2.95
106 <i>lower permeability</i>	0	16.79 $\pm$ 2.54	15.48 $\pm$ 2.40	17.10 $\pm$ 1.14	28.83 $\pm$ 1.25	34.13 $\pm$ 0.59	31.76 $\pm$ 0.97	<b>77.35<math>\pm</math>1.98</b>
	10	11.02 $\pm$ 0.71	10.62 $\pm$ 1.86	12.01 $\pm$ 1.01	15.17 $\pm$ 1.03	26.48 $\pm$ 0.97	19.76 $\pm$ 1.31	<b>66.69<math>\pm</math>2.74</b>
107 <i>more hydrogen bond acceptors</i>	0	12.64 $\pm$ 1.64	10.85 $\pm$ 2.29	11.78 $\pm$ 0.15	21.17 $\pm$ 3.09	54.01 $\pm$ 5.26	37.35 $\pm$ 0.79	<b>95.35<math>\pm</math>0.62</b>
	1	0.69 $\pm$ 0.01	0.90 $\pm$ 0.84	0.67 $\pm$ 0.01	1.83 $\pm$ 0.47	27.33 $\pm$ 2.62	16.13 $\pm$ 2.87	<b>72.60<math>\pm</math>2.51</b>
108 <i>more hydrogen bond donors</i>	0	2.97 $\pm$ 0.61	3.97 $\pm$ 0.55	6.23 $\pm$ 0.66	19.50 $\pm$ 2.86	28.55 $\pm$ 0.76	60.97 $\pm$ 5.09	<b>96.54<math>\pm</math>1.31</b>
	1	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	1.33 $\pm$ 0.24	7.69 $\pm$ 0.56	32.35 $\pm$ 2.57	<b>76.43<math>\pm</math>3.32</b>

**Table 2.** Results on 12 multi-objective small molecule editing, and the evaluation is the hit ratio of the property change. For ChatDrug, we report the mean and std of five random seeds. The best results are marked in **bold**.

Two Target Properties	$\Delta$	Random	PCA	High Variance	GS-Mutate	MoleculeSTM (SMILES)	MoleculeSTM (Graph)	ChatDrug (Ours)
201 <i>more soluble in water and more hydrogen bond acceptors</i>	0 – 0	9.88 $\pm$ 1.03	8.64 $\pm$ 2.06	9.09 $\pm$ 1.25	14.00 $\pm$ 2.48	27.87 $\pm$ 3.86	27.43 $\pm$ 3.41	<b>79.62<math>\pm</math>0.64</b>
	0.5 – 1	0.23 $\pm$ 0.33	0.45 $\pm$ 0.64	0.22 $\pm$ 0.31	0.67 $\pm$ 0.62	8.80 $\pm$ 0.04	11.10 $\pm$ 1.80	<b>49.64<math>\pm</math>2.66</b>
202 <i>less soluble in water and more hydrogen bond acceptors</i>	0 – 0	2.99 $\pm$ 0.38	2.00 $\pm$ 0.58	2.45 $\pm$ 0.67	7.17 $\pm$ 0.85	8.55 $\pm$ 2.75	8.21 $\pm$ 0.81	<b>51.59<math>\pm</math>3.79</b>
	0.5 – 1	0.45 $\pm$ 0.32	0.00 $\pm$ 0.00	0.22 $\pm$ 0.31	0.17 $\pm$ 0.24	2.93 $\pm$ 0.30	0.00 $\pm$ 0.00	<b>24.92<math>\pm</math>4.85</b>
203 <i>more soluble in water and more hydrogen bond donors</i>	0 – 0	2.28 $\pm$ 1.15	2.23 $\pm$ 1.16	4.44 $\pm$ 0.58	13.83 $\pm$ 2.95	33.51 $\pm$ 4.08	49.23 $\pm$ 1.71	<b>89.34<math>\pm</math>0.96</b>
	0.5 – 1	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	9.98 $\pm$ 1.03	23.94 $\pm$ 1.09	<b>53.64<math>\pm</math>5.81</b>
204 <i>less insoluble in water and more hydrogen bond donors</i>	0 – 0	0.69 $\pm$ 0.58	1.96 $\pm$ 0.87	1.79 $\pm$ 0.66	5.67 $\pm$ 0.62	17.03 $\pm$ 2.75	14.42 $\pm$ 3.43	<b>39.90<math>\pm</math>3.86</b>
	0.5 – 1	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	2.59 $\pm$ 1.14	3.84 $\pm$ 0.71	<b>24.19<math>\pm</math>2.19</b>
205 <i>more soluble in water and higher permeability</i>	0 – 0	5.06 $\pm$ 1.21	3.53 $\pm$ 0.38	4.88 $\pm$ 2.21	8.17 $\pm$ 1.03	35.69 $\pm$ 3.19	<b>39.74 <math>\pm</math> 2.26</b>	12.85 $\pm$ 2.68
	0.5 – 10	1.16 $\pm$ 0.68	0.67 $\pm$ 0.55	0.66 $\pm$ 0.54	0.00 $\pm$ 0.00	19.15 $\pm$ 0.73	<b>22.66 <math>\pm</math> 1.90</b>	10.44 $\pm$ 5.75
206 <i>more soluble in water and lower permeability</i>	0 – 0	12.17 $\pm$ 1.05	10.43 $\pm$ 2.88	13.08 $\pm$ 2.28	19.83 $\pm$ 2.46	44.35 $\pm$ 0.68	30.87 $\pm$ 0.62	<b>65.33<math>\pm</math>2.16</b>
	0.5 – 10	6.20 $\pm$ 0.64	6.23 $\pm$ 2.31	6.67 $\pm$ 0.53	4.83 $\pm$ 0.85	28.67 $\pm$ 2.22	20.06 $\pm$ 1.26	<b>52.9<math>\pm</math>2.23</b>

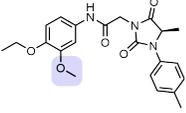
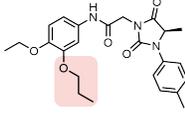
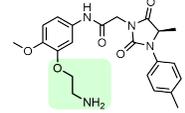
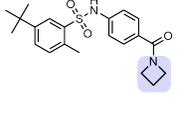
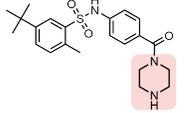
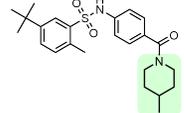
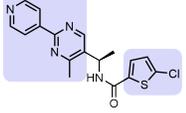
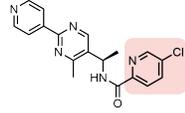
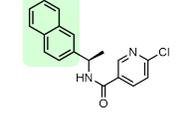
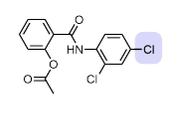
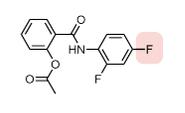
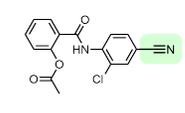
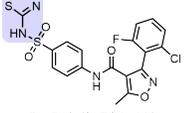
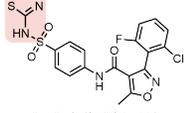
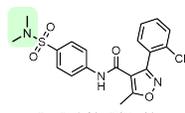
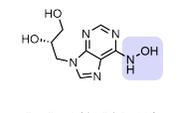
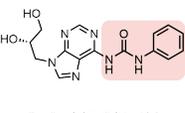
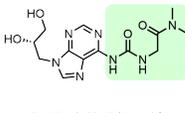
the difference between input and output properties is above the threshold. **Baselines:** The baselines are from [27], based on MegaMolBART [5], a pretrained auto-regressive model. Baselines include Random, PCA, High-Variance, GS-Mutate, and MoleculeSTM with SMILES or Graph as the molecule representation.

**Observation.** We illustrate the descriptions and the single- and multi-objective editing results in Tables 1 and 2, respectively. The threshold  $\Delta$  for each specific task is specified in Table 1; for multi-objective editing tasks in Table 2, the threshold  $\Delta$  has two values corresponding to the two tasks. We further conduct an ablation study on the thresholds of ReDF in Section 4.6. We can observe that ChatDrug can reach the best performance on 22 out of 28 tasks, 20 of which possess over 20% hit ratio than the second-best method. Table 3 visualizes examples of 6 molecule editing tasks where ChatDrug successfully generates output molecules  $\mathbf{x}_{\text{out}}$  with desirable property change, while the output of the first conversation round  $\mathbf{x}_1$  fail. For example, in Table 3a,  $\mathbf{x}_1$  converts a methyl group to a propyl which incorrectly yields a less soluble molecule. Through conversational guidance, ChatDrug changes its output  $\mathbf{x}_{\text{out}}$  to an aminoethyl group, successfully fulfilling the task. In Table 3f,  $\mathbf{x}_1$  installs a phenyl urea to the molecule, which brings lower permeability as requested but makes the molecule less soluble. In contrast, ChatDrug is able to replace the hydrophobic aromatic substituent with a hydrophilic amide in  $\mathbf{x}_{\text{out}}$ , consistent with the requirement of higher solubility in water.

## 4.2 Text-guided Immunogenic Binding Peptide Editing

The second task is text-guided immunogenic binding peptide editing. Immunogenic peptides are promising therapeutic targets for the personalized vaccine, which triggers a person’s immune system, *e.g.*, CD8+ T cells, to fight diseases [52, 53]. Immunogenic peptides are typically degraded from intracellular antigens. To activate CD8+ T cell immune responses, these

**Table 3.** Visualization of six small molecule editing tasks. The blue regions, red regions, and green regions correspond to the edited substructures in the input molecule  $x_{in}$ , intermediate molecule  $x_1$  for the 1st conversation round, and the output molecule  $x_{out}$ , respectively.

(a) Prompt for 101 : more soluble in water			(b) Prompt for 102: less soluble in water		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 LogP: 1.46	 LogP: 2.24	 LogP: 0.40	 LogP: 3.29	 LogP: 2.49	 LogP: 4.31
(c) Prompt for 105: higher permeability			(d) Prompt for 106: lower permeability		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 tPSA: 68	 tPSA: 81	 tPSA: 42	 tPSA: 55	 tPSA: 55	 tPSA: 79
(e) Prompt for 205: more soluble in water and higher permeability			(f) Prompt for 206: more soluble in water and lower permeability		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 LogP: 3.59, tPSA: 114	 LogP: 3.59, tPSA: 114	 LogP: 2.83, tPSA: 93	 LogP: -1.83, tPSA: 116	 LogP: -0.37, tPSA: 125	 LogP: -2.42, tPSA: 146

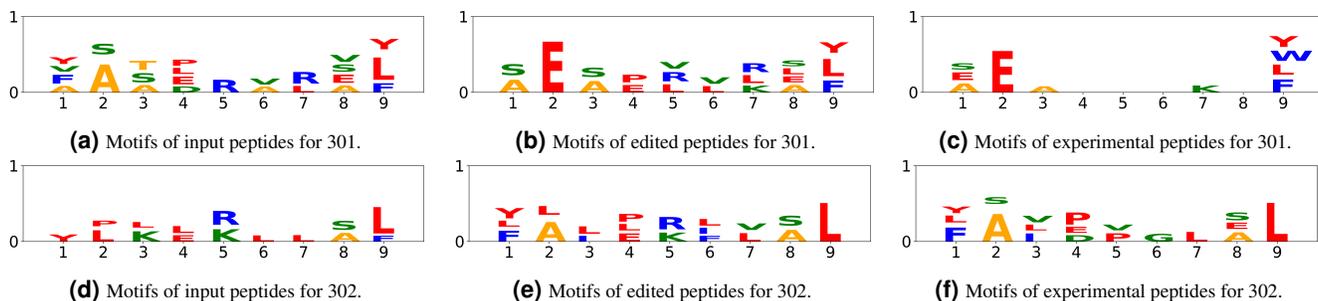
peptides must first bind to Major Histocompatibility Complex (MHC) proteins, forming peptide-MHC complexes which are then presented on the surface of infected or malignant cells to interact with the T cells. Although the peptide-MHC binding process is critical for immune response, it is highly specific, making editing known peptides to improve their binding affinity to specific MHC proteins a challenging yet crucial task for peptide engineering and discovery. Recall that peptides are typically short protein chains, with most peptides having less than 16 amino acids.

**Data:** In this experiment, we use the experimental dataset of peptide-MHC binding affinities [54]. This dataset contains 149 human MHC Class I proteins (alleles) and 309K peptides. We follow existing works [55] on using the 30 common MHC proteins (alleles) and we randomly pick one as the source allele and one or more alleles as the target alleles. Notice that for single-allele tasks, 30 MHC proteins can be further divided into 3 categories: HLA-A, HLA-B, and HLA-C; we make sure that the sampled source and target alleles are from different categories. Then we sample 500 peptides from the source allele types. For the retrieval DB, the experimental data of the target allele(s) are adopted. The sampled MHC types are further specified in Appendix E. **Prompt:** We expect the edited peptides can bind to the target MHC protein(s), so the prompt template is *We want a peptide that binds to [target allele]. We have a peptide [peptide sequence] that binds to [source allele], can you help modify it? The output peptide should be similar to the input peptide.* **Evaluation:** The actual bindings require wet-lab experiments, which are expensive and prohibited for large scaled evaluation. Following existing works [56, 55], we leverage the MHCflurry2.0 [54] as a pseudo-oracle to predict the peptide-MHC binding affinity. MHCflurry2.0 is the state-of-the-art method enabling accurate estimating of the binding affinity of peptides with MHC proteins. The success of the peptide editing needs to satisfy two conditions: (1) The output peptide should have a higher binding affinity with the target allele compared to the input peptide; (2) The binding affinity of the output peptide and target allele should be above a certain threshold. Here we take the threshold as one-half of the average binding affinity of experimental data on the target allele. **Baselines:** Since there is no existing approach for text-guided binding peptide editing, we use random mutation as the baseline, *i.e.*, conducting random mutation on the amino acid sequence of the input peptides.

**Observation.** We illustrate the single- and multi-objective editing results in Table 4. We can observe that ChatDrug reaches the best performance over all 9 tasks compared to the random mutation baselines. We further visualize peptides using position weight matrices (PWMs) in Figure 2. PWM has been widely used for the visualization of protein motifs (patterns), and it plots the distribution of each amino acid at the corresponding position. Thus, more important motifs with higher probabilities will be marked in higher alphabets. According to Figure 2, the edited or optimized peptides follow similar patterns to the experimental data presented. For instance, for task 301, the edited peptides can successfully upweight the alphabet E (glutamic acid) at position 2; similarly, for alphabet A at position 2 and L at position 9 for task 302. These results indicate that the binding motifs of the edited peptides are highly correlated with the real binding motifs derived from experimental data.

**Table 4.** Results on six single-objective and three multi-objective peptide editing tasks. Random Mutation- $R$  for  $R$  mutated positions. The evaluation is the hit ratio of the increased binding affinity score. The best results are marked in **bold**. Due to the space limitation, please check Appendix E for the text prompt of each task.

	single-objective editing						multi-objective editing		
	301	302	303	304	305	306	401	402	403
Random Mutation-1	1.80	14.40	1.80	1.80	12.00	5.60	3.20	0.80	0.40
Random Mutation-2	1.80	13.40	2.80	3.00	8.40	4.40	2.20	0.60	1.20
Random Mutation-3	1.80	9.40	2.40	4.20	9.00	3.80	3.00	0.60	0.80
ChatDrug	<b>58.60</b>	<b>69.34</b>	<b>58.52</b>	<b>55.11</b>	<b>64.40</b>	<b>62.73</b>	<b>53.71</b>	<b>41.45</b>	<b>54.71</b>



**Figure 2.** Visualization of two peptide editing tasks using PWM. The x-axis corresponds to the position index, while the y-axis corresponds to the distribution of each amino acid (in alphabets) at each position.

### 4.3 Text-guided Protein Secondary Structure Editing

Last but not least, we consider text-guided protein secondary structure editing (PSSE) [57]. For protein 1D sequence, it can fold into the 3D structure, as shown in Figure 1. Specifically, proteins possess four levels of structures, and secondary structures are fundamental building blocks, which are local folding patterns stabilized by hydrogen bonds. Typical secondary structures include  $\alpha$ -helix and  $\beta$ -sheet, consisting of  $\beta$ -strands. Here we are interested in two PSSE tasks, *i.e.*, using ChatDrug to edit protein sequences with more helix or strand structures after folding [11, 28].

**Data:** TAPE [58] is a benchmark for protein sequence property prediction, including the secondary structure prediction task. We take the test dataset and training dataset as the input proteins and retrieval DB, respectively. **Prompt:** *For an input protein sequence [protein sequence], can you modify it with more helix/strand structures?* **Baselines:** Same with peptide editing, we adopt random mutation as baselines. **Evaluation.** For evaluation, we adopt the state-of-the-art pretrained secondary structure prediction model, *i.e.*, ProteinCLAP-EBM-NCE model from ProteinDT [26]. The hit condition is if the output protein sequences have more secondary structures than the input sequences.

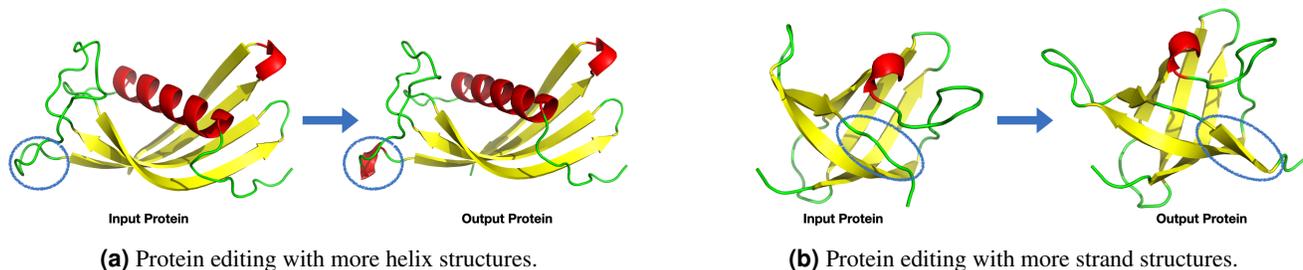
**Observation.** Because we only consider two types of secondary structures in PSSE, the tasks are single-objective tasks. As shown in Table 5, we can tell the large performance gain by ChatDrug. We further visualize cases on how ChatDrug successfully edits the proteins with more helix/strand structures. We adopt pretrained ESMFold [28] for protein folding (protein sequence to protein structure prediction) and then plot the protein structures using PyMOL [59]. We show two examples in Figure 3. As circled in the blue regions in Figures 3a and 3b, the edited proteins possess more helix structures and strand structures, respectively. More visualization can be found in Appendix G.

### 4.4 Ablation Study on Comparison with Zero-shot and In-context Learning

There are two important modules for ChatDrug: conversation for result refinement and the ReDF for knowledge retrieval. Thus in this ablation study, we would like to explore the effect of such two modules. The first case is zero-shot prediction. It is indeed ChatDrug with  $c = 0$ , *i.e.*, without conversation or ReDF. On the other hand, in-context learning (ICL) can be treated as ChatDrug equipped with the ReDF module but without any conversational round. Concretely, the retrieved drug is  $\mathbf{x}_R = \text{ReDF}(\mathbf{x}_{in}, \mathbf{x}_{in})$ , with  $\tilde{\mathbf{x}} = \mathbf{x}_{in}$  in Equation (2). The text prompt for zero-shot and ICL are “Can you edit the molecule

**Table 5.** Results on two protein editing tasks. Random Mutation- $R$  for  $R$  mutated positions. The evaluation is the hit ratio of increased secondary structures accordingly. The best results are marked in **bold**.

	501 more helix	502 more strand
Random Mutation-1	18.32	17.35
Random Mutation-2	24.95	19.69
Random Mutation-3	26.90	21.44
ChatDrug	<b>34.79</b>	<b>51.38</b>



**Figure 3.** Visualization of two protein editing tasks. For the protein secondary structures, the  $\alpha$ -helix is marked in red, and  $\beta$ -sheet is marked in yellow. The edited regions before and after ChatDrug are marked in blue circles.

**Table 6.** Ablation studies on comparison with in-context learning (ICL) and conversation rounds on molecule editing. The threshold is the loose threshold with  $\Delta = 0$ , and the random seed is 0.

	C	101	102	103	104	105	106	107	108	201	202	203	204	205	206
ICL (few-shot)		52.11	75.45	37.76	46.23	30.64	42.86	54.97	69.81	59.88	39.86	53.45	49.36	37.42	42.77
ChatDrug	C = 0 (zero-shot)	78.26	71.35	16.15	32.12	16.04	8.33	59.41	63.16	43.09	0.52	54.49	0.53	2.11	22.22
	C = 1	89.56	93.64	48.35	61.62	47.93	56.97	90.00	93.08	72.29	36.26	86.14	30.00	9.44	54.14
	C = 2	93.37	97.11	52.81	67.93	55.76	78.40	95.57	98.10	80.37	48.52	90.18	39.88	12.72	67.23
	C = 3	96.11	97.69	55.11	75.54	59.51	87.65	98.09	98.73	83.75	60.49	92.02	50.32	15.48	76.74
	C = 4	96.67	97.69	59.20	78.14	63.35	94.41	98.09	98.73	86.79	68.32	94.41	57.42	22.36	80.00
	C = 5	97.22	97.69	59.77	83.06	65.84	95.03	99.36	98.73	89.17	70.19	94.41	63.40	25.32	81.55

**Table 7.** Ablation studies on thresholds in domain feedback function  $D$  with two conversational rounds. The evaluation function  $E$  uses the strict threshold. We report the mean of five seeds, and stds are in Appendix H.

	101	102	103	104	105	106	107	108	201	202	203	204	205	206
loose threshold	80.73	41.00	11.23	16.94	33.16	53.59	14.96	21.93	20.14	7.96	17.93	5.79	3.66	41.04
strict threshold	88.67	70.08	19.37	30.99	43.08	66.69	72.60	76.43	49.64	24.92	53.64	24.19	10.44	52.9

$[\mathbf{x}_{in}]$  to be more soluble in water?” and “We know that  $[\mathbf{x}_R]$  is similar to  $[\mathbf{x}_{in}]$  and is more soluble in water. Can you edit the molecule  $[\mathbf{x}_{in}]$  to be more soluble in water?” The results can be found in Table 6. As we can see, both ChatDrug and ICL are better than the zero-shot prediction, and conversational refinement performs best on all 14 tasks.

#### 4.5 Ablation Study on the Number of Conversation Rounds in ChatDrug

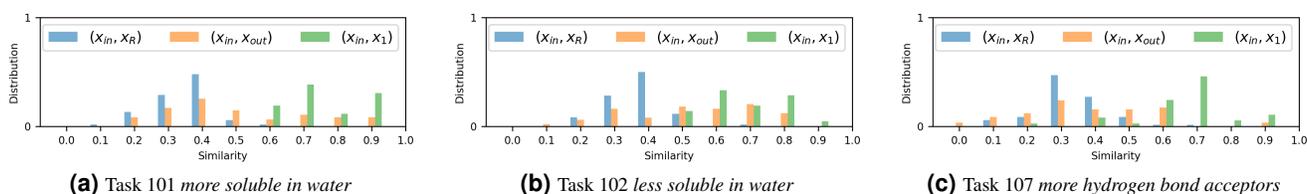
In ChatDrug, the number of conversation rounds is an important hyperparameter. Here we conduct an ablation study on small molecules to test its effectiveness. The results are in Table 6. For molecule editing tasks tested here, the performance of ChatDrug tends to converge after  $C = 2$  conversation rounds. This motivates us taking  $C = 2$  for the main results in Tables 1, 2, 4 and 5.

#### 4.6 Ablation Study on the Thresholds in Feedback Condition Function

In ChatDrug, another important factor is the domain feedback function  $D(\cdot, \cdot; \cdot)$ . For molecule editing, we discuss two thresholds when evaluating with  $E(\cdot, \cdot; \cdot)$ . One is  $\Delta = 0$  (loose condition), and the other is  $\Delta > 0$  (strict condition), where the  $\Delta$  value is different for each task. Here we conduct ablation studies on two conditions for feedback function  $D$ . The results are in Table 7, and the observation is that ChatDrug with the stricter threshold in feedback condition can lead to higher accuracy by a large margin. Note that for each task in Tables 1 and 2, we keep the same threshold for  $D$  and  $E$ .

#### 4.7 Ablation Study on the Similarity Between Input and Output Drugs

We plot the distribution of similarities between input molecules  $\mathbf{x}_{in}$  and retrieval  $\mathbf{x}_R$ , intermediate  $\mathbf{x}_1$ , and output molecules  $\mathbf{x}_{out}$  using ChatDrug. The similarity distributions of three tasks are in Figure 4, and more results are in Appendix G. One interesting observation is that the similarities between  $\mathbf{x}_{in}$  and intermediate molecules are quite  $\mathbf{x}_1$ , but the hit ratio is the lowest among the three. Then we plot the similarity  $\mathbf{x}_{in}$  and  $\mathbf{x}_R$ , where the similarities are comparatively low, yet the hit ratio is the highest. This reveals that the ReDF module is able to explore the chemical space to search for more optimal solutions. Then by utilizing such retrieved information and repeating  $C$  conversational rounds, ChatDrug will do a trade-off between the similarity with input molecules  $\mathbf{x}_{in}$  and knowledge explorations, which ultimately leads to more promising results as in  $\mathbf{x}_{out}$ .



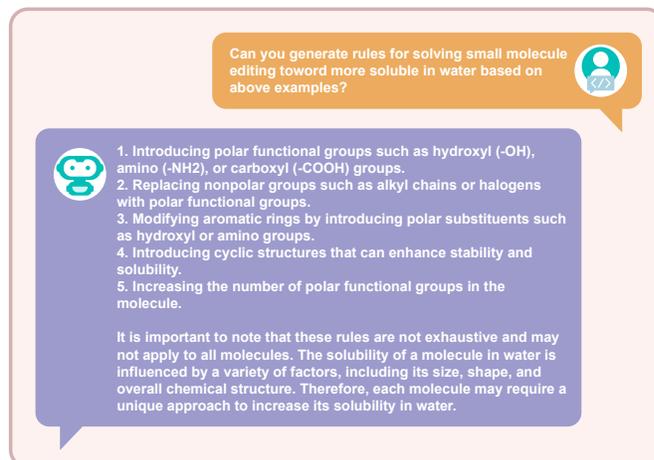
**Figure 4.** Similarity distribution between input molecules  $x_{in}$  and retrieval  $x_R$ , intermediate  $x_1$ , and output molecules  $x_{out}$ . We pick up three tasks on small molecules for visualization, and more results are in Appendix G.

#### 4.8 Why ChatDrug Works? Knowledge Extraction

We are also interested in understanding how ChatDrug can work. As shown in Figure 5, we illustrate a case study on small molecule editing. It can be observed that ChatDrug can do knowledge extraction: for a specific task on editing molecules to be more soluble in water, ChatDrug can extract the reasonings and summarize them into five rules. This gives us the confidence that the success of ChatDrug is its ability of domain interpretation. We conduct further ablation studies like knowledge extraction without the context as a control experiment in Appendix H.

Although ChatDrug can extract domain-specific information for the editing tasks, we do notice a minor issue: the **redundancy** among knowledge. As shown in Figure 5, the extracted rules 1, 3, and 5 are all centered on introducing polar functional groups for solubility in water, despite from slightly different angles. In Appendix H, we continue to explore how ChatDrug can play a positive role in knowledge summarization, aiming to alleviate this knowledge redundancy issue.

**Figure 5.** Knowledge extraction of ChatDrug.



## 5 Limitation and Conclusion

In this work, we present ChatDrug, a framework that utilizes ChatGPT for drug editing tasks. We build up a benchmark on 39 tasks over three main types of drugs: small molecules, peptides, and proteins. Empirical results have verified the effectiveness of ChatDrug on these drug editing tasks, and the visual analysis further qualitatively illustrates how ChatDrug can modify the key substructures for the target properties. Thus, we posit that using conversational LLMs for drug editing is a promising direction for both the machine learning and drug discovery communities.

Meanwhile, ChatDrug also possesses certain limitations. One limitation is that ChatDrug is not good at understanding the complex structures of drugs, *i.e.*, the 3D geometries. This may require a more profound utilization of geometric modeling. Another limitation is that ChatDrug requires certain conversational rounds to reach strong performance. An ideal solution is to reduce such computational costs using the knowledge summarization ability of ChatGPT, and we leave this for future work.

## References

- Sullivan, T. A tough road: cost to develop one new drug is \$2.6 billion; approval rate for drugs entering clinical development is less than 12%. *Policy & Medicine* (2019).
- Rohrer, S. G. & Baumann, K. Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *J. Chem. Inf. Model.* **49**, 169–184, DOI: [10.1021/ci8002649](https://doi.org/10.1021/ci8002649) (2009). PMID: 19161251, <https://doi.org/10.1021/ci8002649>.
- Liu, S. *et al.* Practical model selection for prospective virtual screening. *J. chemical information modeling* **59**, 282–293 (2018).
- Jin, W., Barzilay, R. & Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, 4839–4848 (PMLR, 2020).
- Irwin, R., Dimitriadis, S., He, J. & Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Mach. Learn. Sci. Technol.* **3**, 015022 (2022).
- Wang, Z. *et al.* Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126* (2022).

7. Liu, S. *et al.* GraphCG: Unsupervised discovery of steerable factors in graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning* (2022).
8. Gottipati, S. K. *et al.* Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, 3668–3679 (PMLR, 2020).
9. Shi, C., Xu, M., Guo, H., Zhang, M. & Tang, J. A graph to graphs framework for retrosynthesis prediction. In *ICML* (2020).
10. Bi, H. *et al.* Non-autoregressive electron redistribution modeling for reaction prediction. In *ICML* (2021).
11. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
12. Hsu, C. *et al.* Learning inverse folding from millions of predicted structures. *bioRxiv* (2022).
13. Brown, T. B. *et al.* Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
14. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
15. Yang, Z. *et al.* Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* (2019).
16. Huang, J. *et al.* Large language models can self-improve. *arXiv preprint arXiv:2210.11610* (2022).
17. Zhou, Y. *et al.* Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910* (2022).
18. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. & Iwasawa, Y. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916* (2022).
19. Xie, S. M., Raghunathan, A., Liang, P. & Ma, T. An explanation of in-context learning as implicit bayesian inference. *ICLR* (2021).
20. Mihalić, Z. & Trinajstić, N. A graph-theoretical approach to structure-property relationships. *J. Chem. Educ.* **69**, 701 (1992).
21. Ertl, P., Altmann, E. & McKenna, J. M. The most common functional groups in bioactive molecules and how their popularity has evolved over time. *J. medicinal chemistry* **63**, 8408–8418 (2020).
22. Böhm, H.-J., Flohr, A. & Stahl, M. Scaffold hopping. *Drug discovery today: Technol.* **1**, 217–224 (2004).
23. Hu, Y., Stumpfe, D. & Bajorath, J. Recent advances in scaffold hopping: miniperspective. *J. medicinal chemistry* **60**, 1238–1246 (2017).
24. Drews, J. Drug discovery: a historical perspective. *Science* **287**, 1960–1964 (2000).
25. Gomez, L. Decision making in medicinal chemistry: The power of our intuition. *ACS Medicinal Chem. Lett.* **9**, 956–958 (2018).
26. Liu, S. *et al.* A text-guided protein design framework. *arXiv preprint arXiv:2302.04611* (2023).
27. Liu, S. *et al.* Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789* (2022).
28. Lin, Z. *et al.* Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv* (2022).
29. Wishart, D. S. *et al.* Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research* **36**, D901–D906 (2008).
30. Drug definition. <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/drug>. Accessed: 2023-04-22.
31. Jayatunga, M. K., Xie, W., Ruder, L., Schulze, U. & Meier, C. Ai in small-molecule drug discovery: A coming wave. *Nat. Rev. Drug Discov* **21**, 175–176 (2022).
32. Frokjaer, S. & Otzen, D. E. Protein drug stability: a formulation challenge. *Nat. reviews drug discovery* **4**, 298–306 (2005).
33. Craik, D. J., Fairlie, D. P., Liras, S. & Price, D. The future of peptide-based drugs. *Chem. biology & drug design* **81**, 136–147 (2013).
34. Weininger, D. Smiles, a chemical language and information system. I. introduction to methodology and encoding rules. *J. chemical information computer sciences* **28**, 31–36 (1988).
35. Duvenaud, D. K. *et al.* Convolutional networks on graphs for learning molecular fingerprints. *Adv. neural information processing systems* **28** (2015).

36. Kearnes, S., McCloskey, K., Berndl, M., Pande, V. & Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. computer-aided molecular design* **30**, 595–608 (2016).
37. Liu, S., Demirel, M. F. & Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Adv. neural information processing systems* **32** (2019).
38. Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *The J. Mach. Learn. Res.* **21**, 5485–5551 (2020).
39. Liu, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**, 1–35 (2023).
40. Han, X., Zhao, W., Ding, N., Liu, Z. & Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **3**, 182–192 (2022).
41. Hu, S. *et al.* Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035* (2021).
42. Webson, A. & Pavlick, E. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247* (2021).
43. Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E. & Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980* (2020).
44. He, J., Kryściński, W., McCann, B., Rajani, N. & Xiong, C. Ctrlsum: Towards generic controllable text summarization. *arXiv preprint arXiv:2012.04281* (2020).
45. Schick, T. & Schütze, H. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926* (2020).
46. Dou, Z.-Y., Liu, P., Hayashi, H., Jiang, Z. & Neubig, G. Gsum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014* (2020).
47. Bajusz, D., Rácz, A. & Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. cheminformatics* **7**, 1–13 (2015).
48. Dong, Q. *et al.* A survey for in-context learning. *arXiv preprint arXiv:2301.00234* (2022).
49. Min, S. *et al.* Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837* (2022).
50. Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S. & Coleman, R. G. Zinc: a free tool to discover chemistry for biology. *J. chemical information modeling* **52**, 1757–1768 (2012).
51. Landrum, G. *et al.* RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling (2013).
52. Craiu, A., Akopian, T., Goldberg, A. & Rock, K. L. Two distinct proteolytic processes in the generation of a major histocompatibility complex class i-presented peptide. *Proc. Natl. Acad. Sci.* **94**, 10850–10855 (1997).
53. Hennecke, J. & Wiley, D. C. T cell receptor–mhc interactions up close. *Cell* **104**, 1–4 (2001).
54. O'Donnell, T. J., Rubinsteyn, A. & Laserson, U. Mhcflurry 2.0: improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. *Cell systems* **11**, 42–48 (2020).
55. Chen, Z. *et al.* Binding peptide generation for mhc class i proteins with deep reinforcement learning. *Bioinformatics* **39**, btad055 (2023).
56. Chen, Z., Min, M. R. & Ning, X. Ranking-based convolutional neural network models for peptide-mhc binding prediction. *Front. molecular biosciences* (2021).
57. Klausen, M. S. *et al.* NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins: Struct. Funct. Bioinforma.* **87**, 520–527 (2019).
58. Rao, R. *et al.* Evaluating protein transfer learning with TAPE. *Adv. neural information processing systems* **32** (2019).
59. Schrödinger, L. & DeLano, W. Pymol.
60. Zeng, Z., Yao, Y., Liu, Z. & Sun, M. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nat. communications* **13**, 862 (2022).
61. Edwards, C., Lai, T., Ros, K., Honke, G. & Ji, H. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817* (2022).

62. Edwards, C., Zhai, C. & Ji, H. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 595–607 (2021).
63. Taylor, R. *et al.* Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).
64. Su, B. *et al.* A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481* (2022).
65. Bran, A. M., Cox, S., White, A. D. & Schwaller, P. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376* (2023).
66. Boiko, D. A., MacKnight, R. & Gomes, G. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint arXiv:2304.05332* (2023).
67. Radford, A. *et al.* Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763 (PMLR, 2021).
68. Nichol, A. *et al.* Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).
69. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
70. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D. & Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2085–2094 (2021).
71. Fan, L. *et al.* Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853* (2022).
72. Mikolov, T., Karafát, M., Burget, L., Cernocký, J. & Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, vol. 2, 1045–1048 (Makuhari, 2010).
73. Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Process.* **45**, 2673–2681 (1997).
74. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).
75. Zhu, X., Sobhani, P. & Guo, H. Long short-term memory over recursive structures. In *ICML* (2015).
76. Cho, K. *et al.* Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
77. Vaswani, A. *et al.* Attention is all you need. *Adv. neural information processing systems* **30** (2017).
78. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. *et al.* Improving language understanding by generative pre-training. *OpenAI blog* (2018).
79. Radford, A. *et al.* Language models are unsupervised multitask learners. *OpenAI blog* **1**, 9 (2019).
80. Chung, H. W. *et al.* Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).
81. Iyer, S. *et al.* Opt-impl: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017* (2022).
82. Ouyang, L. *et al.* Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **35**, 27730–27744 (2022).
83. Demirel, M. F., Liu, S., Garg, S., Shi, Z. & Liang, Y. Attentive walk-aggregating graph neural networks. *Transactions on Mach. Learn. Res.* (2021).
84. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272 (PMLR, 2017).
85. Yang, K. *et al.* Analyzing learned molecular representations for property prediction. *J. chemical information modeling* **59**, 3370–3388 (2019).
86. Thomas, N. *et al.* Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219* (2018).
87. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. Schnet—a deep learning architecture for molecules and materials. *The J. Chem. Phys.* **148**, 241722 (2018).

88. Satorras, V. G., Hoogeboom, E. & Welling, M. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844* (2021).
89. Schütt, K. T., Unke, O. T. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150* (2021).
90. Madani, A. *et al.* Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497* (2020).
91. Zhang, Y. *et al.* Automatic term name generation for gene ontology: task and dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4705–4710 (2020).
92. Ingraham, J. *et al.* Illuminating protein space with a programmable generative model. *bioRxiv* DOI: [10.1101/2022.12.01.518682](https://doi.org/10.1101/2022.12.01.518682) (2022).

## A Examples of ChatDrug

Here we show three concrete examples of the editing tasks on small molecules, peptides, and proteins.

The chat interface shows a user asking for five molecules similar to O=C(NC[C@H]1CCCO1)c1ccccc1N1CCCC1=O with increased water solubility and more hydrogen bond donors. The AI provides five SMILES strings. The user points out an error in the first string, and the AI provides a corrected one.

User: Can you make molecule O=C(NC[C@H]1CCCO1)c1ccccc1N1CCCC1=O more soluble in water and more hydrogen bond donors? The output molecule should be similar to the input molecule. Give me five molecules in SMILES only and list them using bullet points. No explanation is needed.

AI: Sure, here are five molecules in SMILES that are similar to the input molecule but have increased solubility in water and more hydrogen bond donors:

- O=C(NC[C@H]1CCCO1)c1ccccc1N1CCOCC1=O
- O=C(NC[C@H]1CCCO1)c1ccccc1N1CCNCC1=O
- O=C(NC[C@H]1CCCO1)c1ccccc1N1CC(O)CC1=O
- O=C(NC[C@H]1CCCO1)c1ccccc1N1CC(O)C1=O
- O=C(NC[C@H]1CCCO1)c1ccccc1N1CC(O)(C)C1=O

User: Your provided sequence O=C(NC[C@H]1CCCO1)c1ccccc1N1CCOCC1=O is not correct. We find a sequence CCOC(=O)Nc1ccccc1C(=O)NC[C@H]1CCCO1 which is correct and similar to the molecule you provided. Can you give me a new molecule?

AI: I apologize for the mistake. Here is a new molecule that is similar to the input molecule but has increased solubility in water and more hydrogen bond donors:

- O=C(NC[C@H]1CCCO1)c1ccccc1N1CC(O)(C)C1=O

User: This is correct.

Figure 6. Example of small molecule editing (task 203).

The chat interface shows a user asking for a modified peptide sequence similar to AAASHFFSL that binds to HLA-B\*44:02. The AI provides the sequence AEAYHFFIL. The user confirms it is correct.

User: We want a peptide that binds to HLA-B\*44:02. We have a peptide AAASHFFNL that binds to HLA-C\*16:01, can you help modify it? The output peptide should be similar to input peptide. Please provide the possible modified peptide sequence only. No explanation is needed.

AI: AAASHFFSL

User: Your provided sequence AAASHFFSL is not correct. We find a sequence AEAYHFFIL which is correct and similar to the peptide you provided. Can you give me a new peptide?

AI: Sure, based on the sequence AEAYHFFIL, a possible modified peptide sequence that may bind to HLA-B\*44:02 could be AEAYHFFSL.

User: This is correct.

Figure 7. Example of peptide editing (task 301).

We have a protein GAPVPVDENDEGLQRALQFAMAEYN-RASNDKYSSRRVVRVISAKRQLVSGIKYLQVEIGRT-TCPKSSGDLQSCFHDPEMAKYTTCTFVVYSIPWLNQIKLLESKCQ. Can you update modify it by making more amino acids into the helix structure (secondary structure)? The input and output protein sequences should be similar but different. No explanation is needed.

Yes. I can modify the protein sequence to increase the helix structure. One way to do this is by introducing more amino acids with helix-forming properties into the sequence. The changes made are as follows:

- Replaced the amino acid "E" at position 7 with "L", which has a higher propensity for helix formation.
- Replaced the amino acid "A" at position 10 with "L", which also has a higher propensity for helix formation.
- Replaced the amino acid "N" at position 14 with "L", which has a high propensity for helix formation.
- ...
- ...

The resulting protein sequence is:  
 GAPVPVLLNLGLQLLLQFLLEYNRASNDKYLLRVLLVISLLQLVS-GIKYLQVEIGRT-TCPKLLGDLQLCLFHDLPKLYTTCTFVVYSIPWLNQLLESKCQ

This sequence should have an increased tendency to form alpha-helices due to the higher proportion of helix-forming amino acids.

This is correct

**Figure 8.** Example of protein editing (task 501).

## B More Discussions on ChatDrug

In this section, we will discuss three aspects of ChatDrug and the general conversational LLMs for drug discovery: the scope, the main attributes, and the key challenges and guidelines when using ChatDrug.

**Scope of ChatDrug.** The first natural question raised is *What are the feasible drug discovery tasks for ChatDrug?* To answer this question, we need to reiterate the conversational LLM’s feasibility for drug discovery tasks. There have been a series of works [60, 61, 62, 63, 64] exploring the LLMs for small molecule and protein discovery, ranging from molecule representation to text-to-molecule generation. These are important machine learning tasks, revealing domain data’s (conditional) distribution learning ability. However, there exists another important task in real scenarios: *drug editing* (a.k.a. *lead optimization* or *protein design* in domain applications). This is a routine task in pharmaceutical companies, and it aims at updating the molecule’s substructures [20], related to certain key tactics in drug discovery like functional group change [21] and scaffold hopping [22, 23]. Traditional solutions relying on domain experts for manual editing can be subjective or biased [24, 25]. To alleviate this issue, multi-modal models with LLMs provide a promising solution, and recent works [26, 27] have started to explore text-guided drug editing or controllable drug generation. However, these works are domain-specific (*e.g.*, only for small molecules or proteins) and do not possess conversational potentials like ChatGPT. In contrast, ChatDrug possesses outstanding generalization abilities to various domain tasks and enables conversational refinement in drug editing tasks.<sup>2</sup>

**Attributes of ChatDrug.** We conclude three fundamental attributes of ChatDrug: open vocabulary, compositionality, and inspiration. (1) Open vocabulary means ChatDrug is not limited to a fixed set of pre-defined drug-related annotations. Instead, it can generalize to novel drug concepts due to the unbound vocabulary depicted by the natural language. (2) Compositionality means we can express a complex concept by decomposing it into simple concepts. (3) Inspiration means the position of ChatDrug is to inspire domain experts with knowledge exploration but not replace them. A concrete example illustrating these three attributes is *multi-objective lead optimization*. We can use natural language to guide us to generate an entirely new attribute of a molecule (open vocabulary); meanwhile, the new attribute is composed of multiple simple attributes, such as binding to a new protein and high permeability (compositionality). Finally, such an optimized molecule may not be directly used for real scenarios, but it can provide insights for domain experts in drug design (inspiration).

**Challenges and Guidelines when Using ChatDrug.** Now that we have decided to narrow ChatDrug to the drug editing tasks, we need to scrutinize more details before deploying it. With careful reconsidering, we summarize two main challenges that we need to keep in mind. (1) ChatDrug can do better in fuzzy searching than exact searching in drug editing tasks. Drug editing tasks, or drug controllable generation, can cover various topics. However, one critical difference between ChatDrug and other LLMs in vision tasks is that ChatDrug or drug discovery is **a scientific problem** while the image and video [67, 68, 69, 70, 71] generation is more of **an artistic endeavor**. Namely, for text prompts like “I want to add an isobutyl group on the 3-position of the aromatic ring in Aspirin”, domain experts can do this precisely, and thus the impact of ChatDrug is limited here. However, for other tasks like “I want to modify this molecule to be more soluble in the water”, the results are not deterministic, and this is where LLMs can act as a more useful tool to inspire the domain experts. These two types of text prompts are called exact searching and fuzzy searching, respectively. We conclude that ChatDrug is more beneficial for the fuzzy searching problem. (2) ChatDrug relies on the pretrained LLMs, initially pretrained on a large-scale and universal corpus. Thus, there is a noticeable domain shift when applying them to domain-specific tasks. However, as will be shown in Sections 3 and 4, the existing LLMs illustrate the interpretation ability of the domain knowledge. Though such interpretation is preliminary, we believe that ChatDrug is an inspiring and promising direction for future usage in both communities.

---

<sup>2</sup>We acknowledge that there have been certain parallel works [65, 66] exploring conversational LLMs on reaction and synthesis tasks.

## C Related Work

### C.1 Large Language Models

Large language models (LLMs), which predict subsequent words in a sentence, have facilitated the generation of human-like text. Initially, neural language models, such as Recurrent Neural Networks (RNNs) [72, 73], Long Short-Term Memory (LSTM) [74, 75], and Gated Recurrent Units (GRU) [76], were developed. These models processed text sequentially, allowing them to capture some contextual nuances. However, they struggled with long-range dependencies and computational efficiency. This challenge paved the way for the transformative architecture of Transformers [77], equipped with an attention mechanism. Transformers revolutionized the handling of long-range dependencies, offering a significant improvement over RNNs and LSTMs by enabling parallel computation across sentences. The introduction of the Transformer architecture marked a significant shift in NLP, laying the foundation for influential models. It enables the development of BERT [14], T5 [38], Generative Pre-trained Transformer (GPT) [78] and so on. GPT-3 [13], for example, has 175 billion parameters and can generate human-like text that is almost indistinguishable from human writing. Despite the advancements, large models such as GPT-2 [79], GPT-3 [13], T5 [38], BERT [14] faced difficulties in consistently producing desired outputs, specifically in adhering to natural language instructions and executing real-world tasks. This gap led to the exploration of instruction-tuning methods, aiming to enhance the zero-shot and few-shot generalization capabilities of LLMs. Instruction-tuned counterparts, such as ChatGPT, FLAN-T5 [80], FLANPaLM [80], and OPT-IML [81], were born from this endeavor. Among these, ChatGPT stands out. It was initially trained on a substantial internet text corpus, followed by a unique fine-tuning process: AI trainers simulated a range of conversational scenarios, assuming both user and AI assistant roles. Reinforcement learning from human feedback (RLHF) [82] was later incorporated to further boost the system's performance. In this paper, we aim to leverage the large language model to explore its functionality in the drug editing domain.

### C.2 Multi-modal Modeling for Small Molecule Discovery

Small molecules can be roughly categorized into two big modalities [60, 27]: the **internal chemical structure** and **external description**. The internal chemical structure refers to the molecule's structure information, *e.g.*, 1D sequence (SMILES) [34], 2D molecular graph [83, 84, 35, 85], and 3D geometric graph [86, 87, 88, 89]. On the other hand, the external description depicts the high-level information of molecules, *e.g.*, the molecule's binding affinity with potential targets, and the functionalities of molecules.

Recently, a research line has been starting to bridge the gap between such two modalities. KV-PLM [60] first applies the joint masking auto-encoding on the SMILES string and biomedical textual description. Text2Mol [62] conducts contrastive learning between molecular graph and text data for retrieval tasks between modalities. MolT5 [61] does the translation between SMILES and textual annotation of molecules in a mutual way. MoMu [64] also conducts contrastive learning while it considers both the retrieval and molecule captioning and text-to-molecule tasks. MoleculeSTM [27] proposes a larger molecule-text dataset and highlights the text-guided molecule editing tasks. Such tasks reveal the potential of LLMs for more realistic drug discovery tasks.

### C.3 Multi-modal Modeling for Peptide and Protein Discovery

There have also been several works exploring multi-modal modeling for protein discovery. ProGen [90] is a text-to-sequence protein design framework, but it is fixed to a predefined set of texts, which can be treated with indices. Thus it is not open-vocabulary and lacks the generalization ability to novel textual descriptions. Besides, the predefined texts and indices cannot sufficiently describe the protein functions [91]. ProteinDT [26] is a recent work that addresses this issue with free-text protein design. A parallel work is Chroma [92], and it conducts text-guided protein editing on the backbone structure instead of the sequence.

## D Data Specification

Drugs like small molecules and proteins can have multiple modalities. Specifically, small molecules can be naturally represented as 1D sequence, 2D molecular graph, and 3D geometric graph, biological knowledge graph, and textual description. The first three data structures capture the internal chemical structure information, while the last two data structures provide a higher-level view of the molecule's functionalities (*e.g.*, the molecule's interactions with other proteins or diseases.).

There are 20 amino acids in nature, as listed below:

**Table 8.** 20 amino acids and the corresponding abbreviations.

Amino Acid	Alphabet
Isoleucine	I
Valine	V
Leucine	L
Phenylalanine	F
Cysteine	C
Methionine	M
Alanine	A
Glycine	G
Threonine	T
Serine	S
Tryptophan	W
Tyrosine	Y
Proline	P
Histidine	H
Asparagine	N
Asparatic acid	D
Glutamine	Q
Glutamic acid	E
Lysine	K
Arginine	R

## E Task Specification

Here we present all the task specifications and prompts used in our experiments.

- We list the template of prompts of two stages of PDDS and ReDF in Tables 9, 11 and 13 for small molecules, peptides, and proteins, respectively.
- We list the corresponding task requirement and allele type information in Tables 10, 12 and 14.
- We further list the prompts of in-context learning in Table 15 for reference.

**Table 9.** Prompt for small molecule editing. The task requirement can be found in Table 10.

Task	Module	Prompt
1xx (101-108)	PDDS	Can you make molecule [input SMILES] [task requirement 1]? The output molecule should be similar to the input molecule. Give me five molecules in SMILES only and list them using bullet points. No explanation is needed.
	ReDF	Your provided sequence [output SMILES] is not correct. We find a sequence [retrieved SMILES] which is correct and similar to the molecule you provided. Can you give me a new molecule?
2xx (201-206)	PDDS	Can you make molecule [input SMILES] [task requirement 1] and [task requirement 2]? The output molecule should be similar to the input molecule. Give me five molecules in SMILES only and list them using bullet points. No explanation is needed.
	ReDF	Your provided sequence [output SMILES] is not correct. We find a sequence [retrieved SMILES] which is correct and similar to the molecule you provided. Can you give me a new molecule?

**Table 10.** Task requirement for small molecule editing, corresponding to Table 9.

Task ID	Task Requirement 1	Task Requirement 2
101	more soluble in water	None
103	more like a drug	None
104	less like a drug	None
105	higher permeability	None
106	lower permeability	None
107	more hydrogen bond acceptors	None
108	more hydrogen bond donors	None
201	more soluble in water	more hydrogen bond acceptors
202	less soluble in water	more hydrogen bond acceptors
203	more soluble in water	more hydrogen bond donors
204	less soluble in water	more hydrogen bond donors
205	more soluble in water	higher permeability
206	more soluble in water	lower permeability

**Table 11.** Prompt for peptide editing. The source allele target type and target allele type can be found in Table 12.

Task	Stage	Prompt
3xx (301-306)	PDDS	We want a peptide that binds to [target allele type 1]. We have a peptide [input peptide] that binds to [source allele type], can you help modify it? The output peptide should be similar to input peptide. Please provide the possible modified peptide sequence only. No explanation is needed.
	ReDF	Your provided sequence [output peptide] is not correct. We find a sequence [retrieved peptide] which is correct and similar to the peptide you provided. Can you give me a new peptide?
4xx (401-403)	PDDS	We want a peptide that binds to [target allele type 1] and [target allele type 2]. We have a peptide [input peptide] that binds to [source allele type], can you help modify it? The output peptide should be similar to input peptide. Please provide the possible modified peptide sequence only. No explanation is needed.
	ReDF	Your provided sequence [output peptide] is not correct. We find a sequence [retrieved peptide] which is correct and similar to the peptide you provided. Can you give me a new peptide?

**Table 12.** Target allele type and source allele type for peptide editing, corresponding to Table 11

Task ID	Source Allele Type	Target Allele Type 1	Target Allele Type 2
301	HLA-C*16:01	HLA-B*44:02	None
302	HLA-B*08:01	HLA-C*03:03	None
303	HLA-C*12:02	HLA-B*40:01	None
304	HLA-A*11:01	HLA-B*08:01	None
305	HLA-A*24:02	HLA-B*08:01	None
306	HLA-C*12:02	HLA-B*40:02	None
401	HLA-A*29:02	HLA-B*08:01	HLA-C*15:02
402	HLA-A*03:01	HLA-B*40:02	HLA-C*14:02
403	HLA-C*14:02	HLA-B*08:01	HLA-A*11:01

**Table 13.** Prompt of Conversation Module for protein editing. The task requirement can be found in Table 14.

Task ID	Prompt	
5xx (501-502)	PDDS	We have a protein [input protein]. Can you update modify it by [task requirement]? The input and output protein sequences should be similar but different. No explanation is needed.
	ReDF	Your provided sequence [output protein] is not correct. We find a sequence [retrieved protein] which is correct and similar to the protein you provided. Can you give me a new protein?

**Table 14.** Task requirement for protein editing, corresponding to Table 13.

Task ID	Task Requirement
501	making more amino acids into the helix structure (secondary structure)
502	making more amino acids into the strand structure (secondary structure)

**Table 15.** Prompt of in-context learning.

Task	Prompt
1xx (101-108)	Can you make molecule [input SMILES] [task requirement]? The output molecule should be similar to the input molecule. We have known that similar molecule [retrieved SMILES] is one of the correct answers. Give me another five molecules in SMILES only and list them using bullet points. No explanation is needed.
2xx (201-208)	Can you make molecule [input SMILES] [task requirement 1] and [ask requirement 2]? The output molecule should be similar to the input molecule. We have known that similar molecule [retrieved SMILES] is one of the correct answers. Give me another five molecules in SMILES only and list them using bullet points. No explanation is needed.
3xx (301-306)	We want a peptide that binds to [target allele type]. We have a peptide [input peptide] that binds to [source allele type], can you help modify it? The output peptide should be similar to input peptide. We have known that similar peptide [retrieved peptide] is one of the correct answers. Please provide another possible modified peptide sequence only. No explanation is needed.
4xx (401-403)	We want a peptide that binds to [target allele type 1] and [target allele type 2]. We have a peptide [input peptide] that binds to [source allele type], can you help modify it? The output peptide should be similar to input peptide. We have known that similar peptide [retrieved peptide] is one of the correct answers. Please provide another possible modified peptide sequence only. No explanation is needed.
5xx (501-502)	We have a protein [input protein]. Can you update modify it by [task requirement]? The input and output protein sequences should be similar but different. We have known that similar protein [retrieved protein] is one of the correct answers. Please provide another possible modified protein only. No explanation is needed.

## F Implementation and Hyperparameters

### F.1 ChatGPT Settings

We implement our experiments with ChatGPT through OpenAI API. Specifically, we utilize the model *gpt-3.5-turbo* under *ChatCompletion* function, which is the standard approach for deploying ChatGPT. To facilitate the replication of our experiments, we set the *temperature* to 0, ensuring deterministic output. Additionally, we observe that ChatGPT often generates repeated sequences or fails to stop generating sequences for chemistry-related questions. To mitigate this issue, we set the *frequency\_penalty* to 0.2. Moreover, for improved adaptation to different domains, it is advisable to incorporate a system role prompt within ChatGPT. In our case, we utilize the following prompt: "You are an expert in the field of molecular chemistry."

### F.2 Experiments Threshold for Small Molecule Editing

Following MoleculeSTM [27], in our small molecule editing experiments, we utilize two different threshold settings: a loose threshold and a strict threshold. For the main results in Tables 1 and 2, we keep the same threshold for domain feedback function *D* and evaluation function *E*. The threshold  $\Delta$  used for each small molecule editing task is shown in Table 16, which holds for both functions.

**Table 16.** Threshold  $\Delta$  for each small molecule editing task,  $\Delta_1$  and  $\Delta_2$  represent the threshold of task requirement 1 and task requirement 2, respectively.

Task ID	Loose Threshold		Strict Threshold	
	$\Delta_1$	$\Delta_2$	$\Delta_1$	$\Delta_2$
101	0	–	0.5	–
102	0	–	0.5	–
103	0	–	0.1	–
104	0	–	0.1	–
105	0	–	10	–
106	0	–	10	–
107	0	–	1	–
108	0	–	1	–
201	0	0	0.5	1
202	0	0	0.5	1
203	0	0	0.5	1
204	0	0	0.5	1
205	0	0	0.5	10
206	0	0	0.5	10

### F.3 Experiments Threshold for Peptide Editing

For the peptide editing task, as mentioned in Section 4, we take the threshold as one-half of the average binding affinity of experimental data on the target allele. The original average binding affinity of each experimental data can be found in the source code.

### F.4 Evaluation Metric

We evaluate the performance of ChatDrug by hit ratio, which is computed by the following equation:

$$\text{Hit Ratio} = \frac{\text{Number of Success Sequence Editing}}{\text{Number of Valid Sequence Editing}} \quad (3)$$

One point we need to highlight is that if ChatDrug returns an invalid sequence, we would just skip and do not consider it in computing the hit ratio. That is why we use "Number of Valid Sequence Editing" as the denominator here.

In small molecule editing tasks, ChatDrug tends to return more than one sequence in the PDDS module. Thus, we add a prompt "Give me five molecules in SMILES only and list them using bullet points." to unify the numbers and format of molecules returned by ChatDrug. In the experiments of the Conversation module, we always choose the first valid molecule as the beginning of the conversation. We further carry out an ablation study to explore the effect of using more molecules in the PDDS module.

## F.5 Randomness

The experiment results of the PDDS Module are entirely deterministic. Any randomness observed in ReDF Module and Conversation Module is due to the utilization of different seeds during the sampling of retrieval database DB from ZINC for molecule editing.

Specifically, for small molecule editing, we adopt seed 0,1,2,3,4 for main results in Tables 1 and 2, and seed 0 for the other ablation studies.

## F.6 Computational Resources

All of our experiments are conducted on a single NVIDIA RTX A6000 GPU. The GPU is only used for peptide and protein evaluation. The primary cost incurred during our experiments comes from the usage of the OpenAI API for ChatGPT, which amounted to less than \$100 in total.

## G Qualitative Analysis

In the main body, we provide 10 case studies and 3 similarity distributions to illustrate the effectiveness of ChatDrug for small molecule editing, peptide editing, and protein editing.

In this section, we provide additional case studies and similarity distributions as follows:

- We list 8 case studies on functional group change of small molecules in Appendix G.1.1.
- We list 14 similarity comparisons on small molecules in Appendix G.1.2.
- We list 9 motif updates for all 9 peptide editing tasks in Appendix G.2.
- We list 8 case studies on secondary structure change of proteins in Appendix G.3.

We want to specify that for all the qualitative analyses listed here, we are using  $C = 2$  conversation rounds. Especially for small molecules, we consider random seed with 0 and the loose threshold, *i.e.*,  $\Delta = 0$  for all tasks.

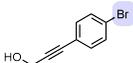
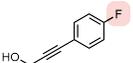
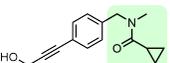
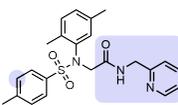
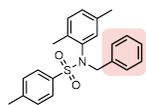
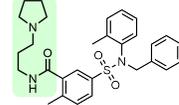
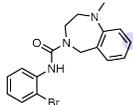
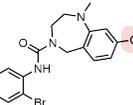
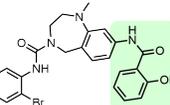
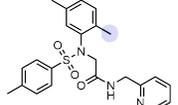
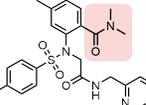
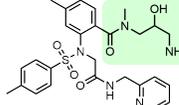
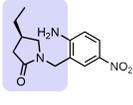
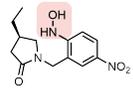
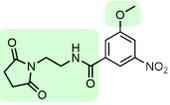
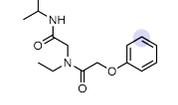
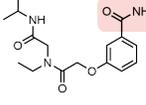
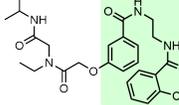
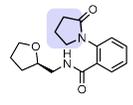
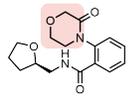
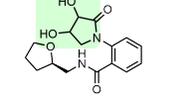
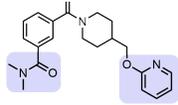
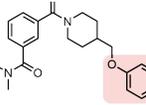
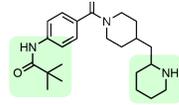
### G.1 Small Molecules

#### G.1.1 Functional Group Change on Small Molecules

Table 17 visualizes examples of 8 molecule editing tasks where ChatDrug successfully generates output molecules  $x_{out}$  with desirable property change, while the output of the first conversation round  $x_1$  fail. In Table 17a and b,  $x_{out}$  successfully adds the desirable fragments to alter the drug likeness of  $x_{in}$ , while  $x_1$  does so in the wrong direction. In Table 17c,  $x_1$  installs a chloride but maintains the same number of hydrogen bond acceptors (HBAs). In contrast, ChatDrug adds a salicylamide moiety that brings two more HBAs. Similarly, in Table 17d, the number of hydrogen bond donors (HBDs) remains in  $x_1$  but successfully increases in  $x_{out}$  via insertions of alcohols and amines.

In Table 17e and f, both cases of  $x_1$  are able to increase the number of HBAs as indicated in the prompt, but the water solubilities shift oppositely. The output molecules successfully fix the trend. In particular, hydrophobicity is appropriately employed in Table 17f to balance the additional polarity from HBAs, generating a less soluble molecule. In Table 17g and h, both cases of  $x_1$  satisfy the solubility requirement but not through the change of HBDs. In  $x_{out}$ , the problems are solved by having extra HBDs with further enhanced solubility changes.

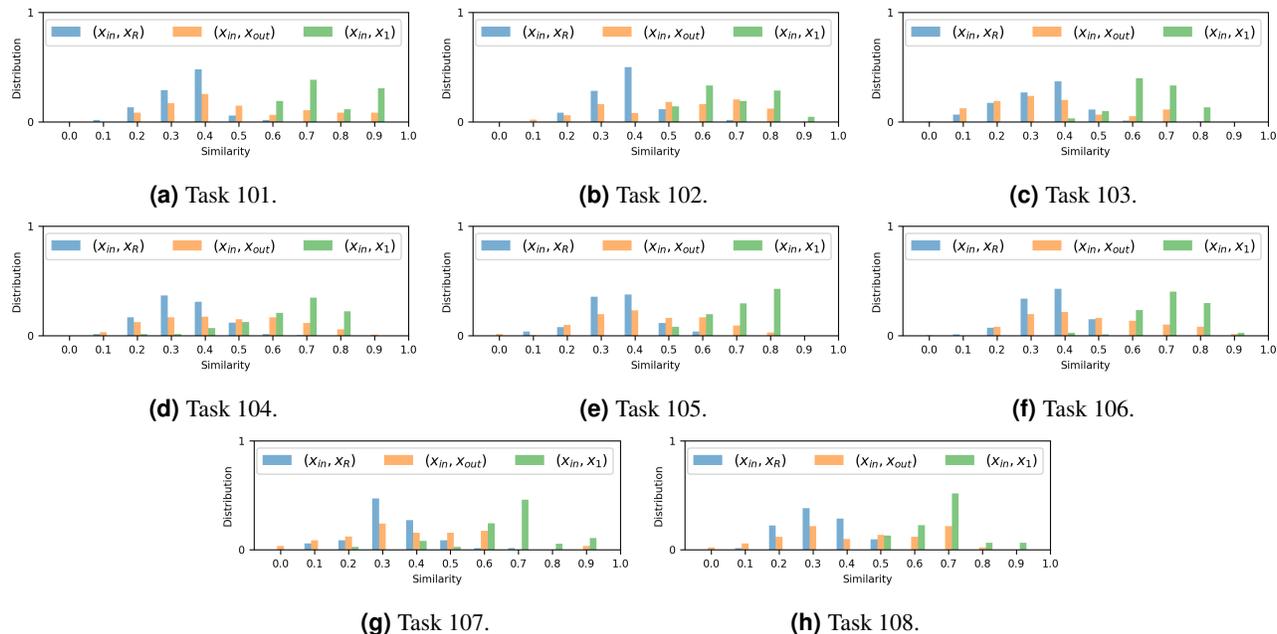
**Table 17.** Visualization of additional eight small molecule editing cases. The blue regions, red regions, and green regions correspond to the edited substructures in the input molecule  $x_{in}$ , intermediate molecule  $x_1$  in the 1st conversation round, and the output molecule  $x_{out}$ , respectively.

(a) Prompt for 103: more like a drug			(b) Prompt for 104: less like a drug		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 QED: 0.648	 QED: 0.550	 QED: 0.815	 QED: 0.631	 QED: 0.637	 QED: 0.397
(c) Prompt for 107: more hydrogen bond acceptors			(d) Prompt for 108: more hydrogen bond donors		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 HBA: 2	 HBA: 2	 HBA: 4	 HBD: 1	 HBD: 1	 HBD: 3
(e) Prompt for 201: more soluble in water and more hydrogen bond acceptors			(f) Prompt for 202: less soluble in water and more hydrogen bond acceptors		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 LogP: 1.12, HBA: 4	 LogP: 1.34, HBA: 5	 LogP: -0.42, HBA: 6	 LogP: 0.98, HBA: 3	 LogP: 0.02, HBA: 4	 LogP: 1.68, HBA: 5
(g) Prompt for 203: more soluble in water and more hydrogen bond donors			(h) Prompt for 204: less soluble in water and more hydrogen bond donors		
Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$	Input Molecule $x_{in}$	Intermediate Molecule $x_1$	Output Molecule $x_{out}$
 LogP: 1.29, HBD: 1	 LogP: 0.52, HBD: 1	 LogP: -0.77, HBD: 3	 LogP: 2.14, HBD: 0	 LogP: 2.75, HBD: 0	 LogP: 3.62, HBD: 2

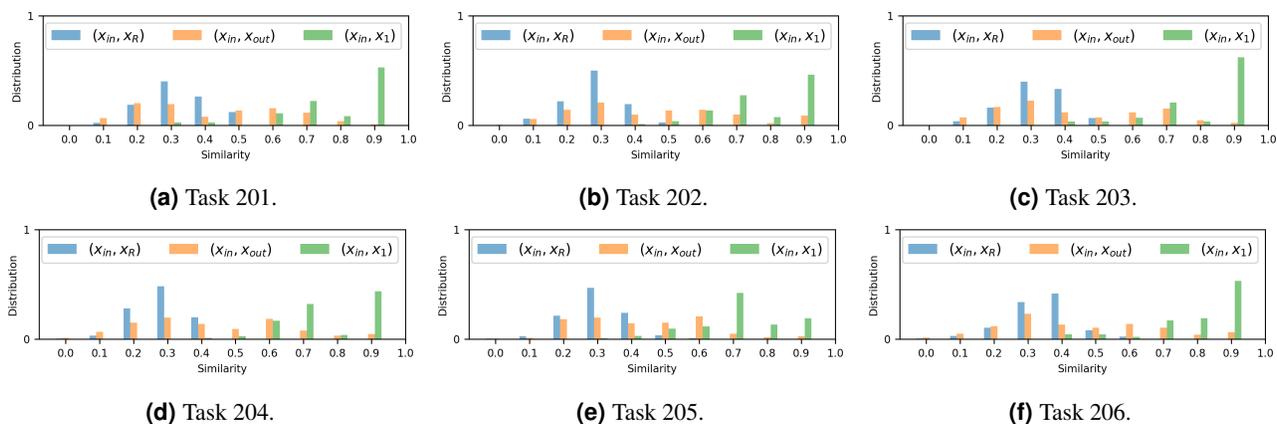
### G.1.2 Similarity Between Input, Intermediate, Retrieved, and Output Molecules

In Figure 4, we plot the distribution of similarities between input molecules  $x_{in}$  and retrieval  $x_R$ , intermediate  $x_1$ , and output molecules  $x_{out}$  using ChatDrug. Here we provide more results. The distributions of 8 single-objective small molecule editing tasks can be found in Figure 9, and 6 multi-objective small molecule editing tasks can be found in Figure 10.

As shown in Figures 9 and 10, the observation of similarity distribution satisfies for all 8 single-objective and 6 multi-objective tasks.



**Figure 9.** Similarity distribution between input molecules  $x_{in}$  and retrieval  $x_R$ , intermediate  $x_1$ , and output molecules  $x_{out}$ . Here we show the distribution of 8 single-objective editing tasks.

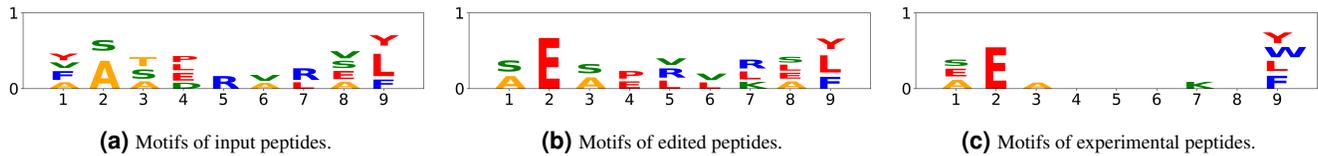


**Figure 10.** Similarity distribution between input molecules  $x_{in}$  and retrieval  $x_R$ , intermediate  $x_1$ , and output molecules  $x_{out}$ . Here we show the distribution of 6 multi-objective editing tasks.

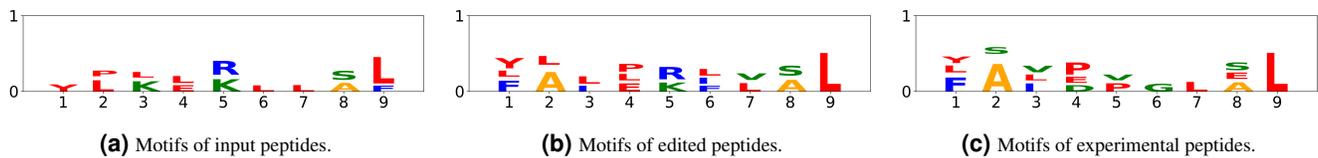
## G.2 Peptide

In the main body, we have illustrated how the motif of peptides changes for two peptide editing tasks. Here we show all 6 single-objective editing tasks in Figures 11 to 16.

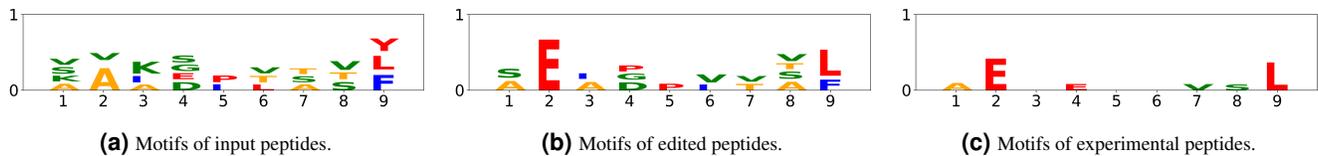
- For task 301 in Figure 11, ChatDrug can successfully upweight E (Glutamic acid) at position 2.
- For task 302 in Figure 12, ChatDrug can successfully upweight A (Alanine) at position 2, and L (Leucine) at position 9.
- For task 303 in Figure 13, ChatDrug can successfully upweight E (Glutamic acid) at position 2, and L (Leucine) at position 9.
- For task 304 in Figure 14, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9.
- For task 305 in Figure 15, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9.
- For task 306 in Figure 16, ChatDrug can successfully upweight E (Glutamic acid) at position 2, and L (Leucine) at position 9.



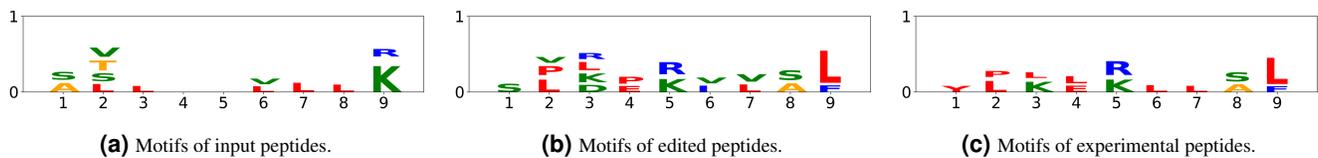
**Figure 11.** Visualization for peptide editing for task 301, higher binding affinity to HLA-B\*44:02.



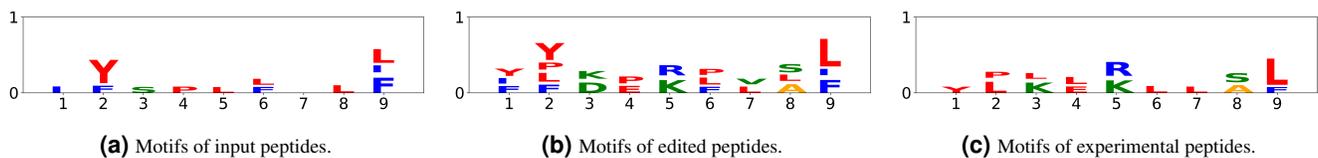
**Figure 12.** Visualization for peptide editing for task 302, higher binding affinity to HLA-C\*03:03.



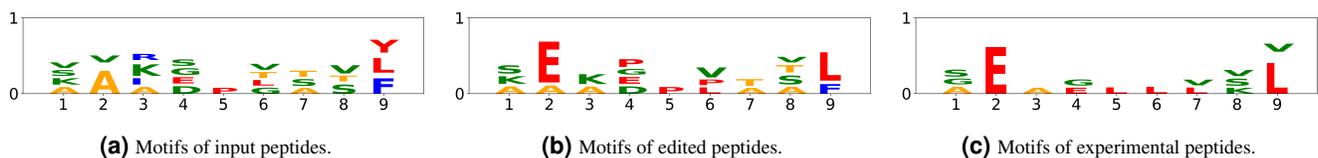
**Figure 13.** Visualization for peptide editing for task 303, higher binding affinity to HLA-B\*40:01.



**Figure 14.** Visualization for peptide editing for task 304, higher binding affinity to HLA-B\*08:01.



**Figure 15.** Visualization for peptide editing for task 305, higher binding affinity to HLA-B\*08:01.

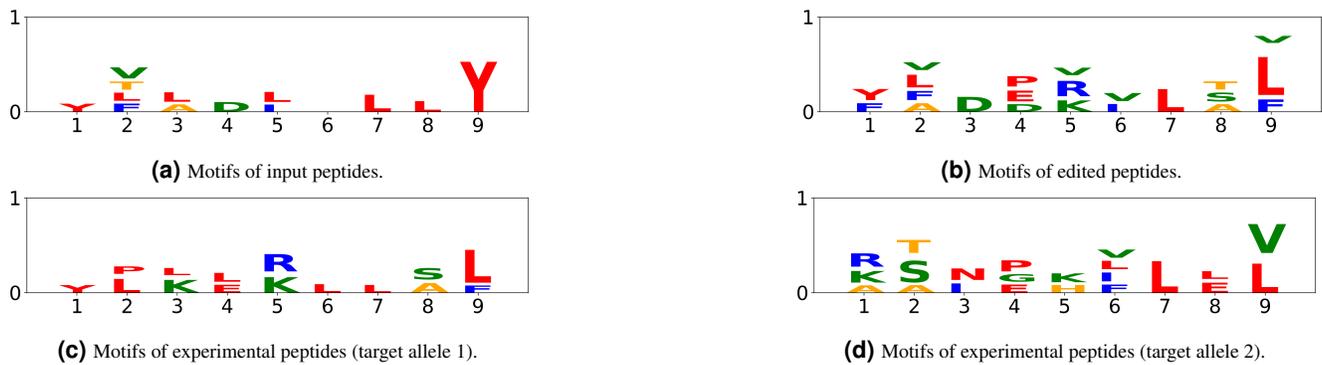


**Figure 16.** Visualization for peptide editing for task 306, higher binding affinity to HLA-B\*40:02.

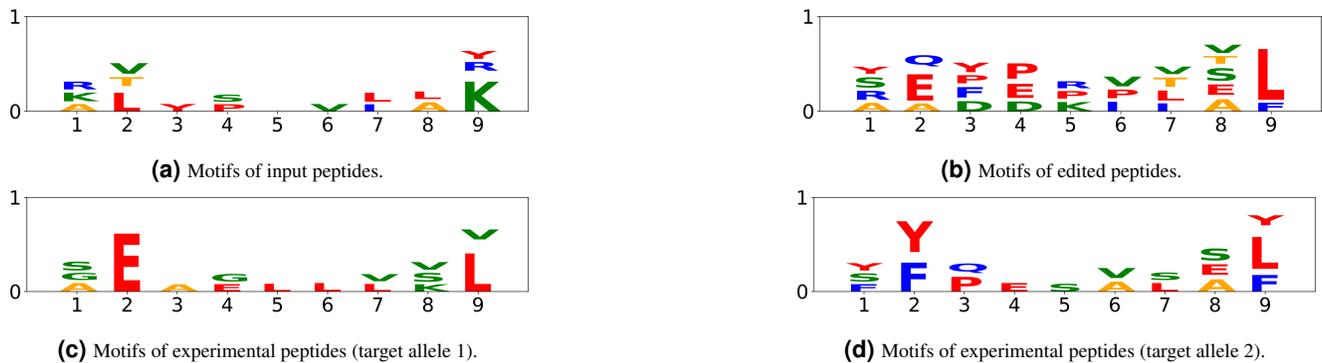
Here we show all 3 multi-objective editing tasks in Figures 17 to 19. Notice that here there are two target allele types, and we mark them as “target allele 1” and “target allele 2”.

- For task 401 in Figure 17, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) and F (Phenylalanine) at position 9 for target allele type 1. ChatDrug can also upweight L (Leucine) at position 7, and V (Valine) and L (Leucine) at position 9 for target allele type 2.
- For task 402 in Figure 18, ChatDrug can successfully upweight E (Glutamic acid) at position 2, and L (Leucine) at position 9 for target allele type 1. ChatDrug can also upweight F (Phenylalanine) and L (Leucine) at position 9 for target allele type 2.
- For task 403 in Figure 19, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9 for target allele type 1.

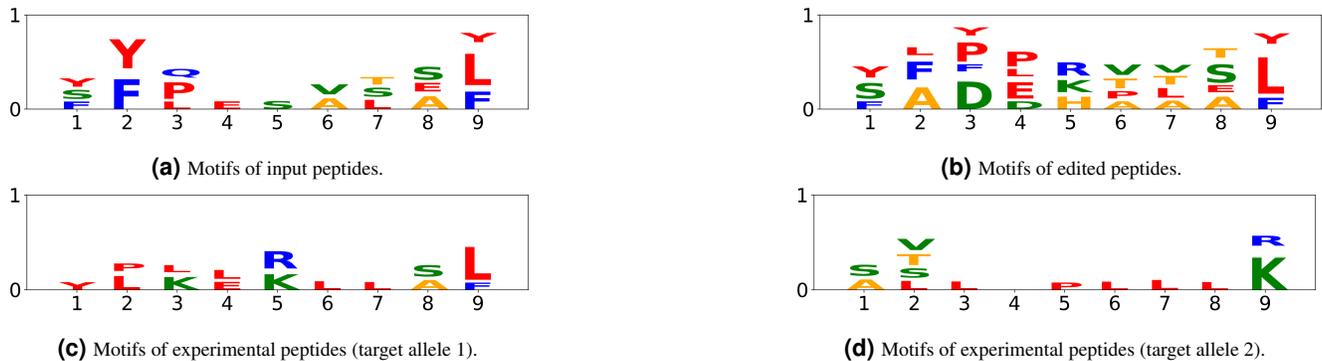
**Potential issue on conflicts among target allele types.** One potential challenge is that for multi-objective editing, the target allele types could have conflicting motifs, like the two target alleles for task 403. We leave this for future exploration.



**Figure 17.** Visualization for peptide editing for task 401, higher binding affinity to HLA-B\*08:01 and HLA-C\*15:02.



**Figure 18.** Visualization for peptide editing for task 402, higher binding affinity to HLA-B\*40:02 and HLA-C\*14:02.

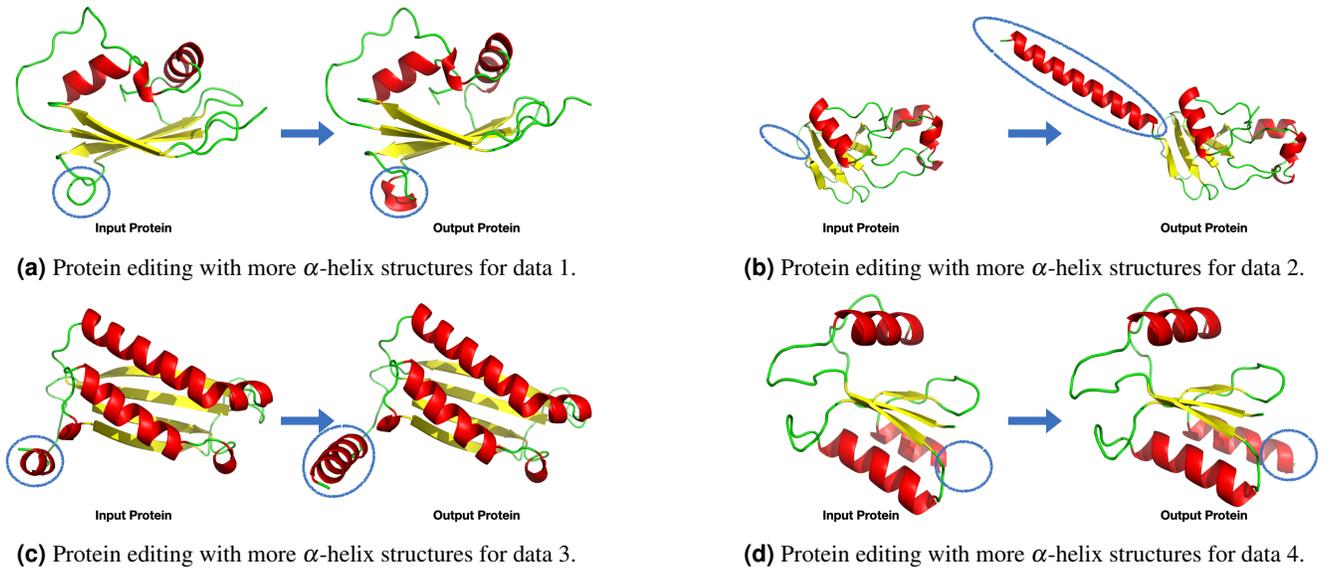


**Figure 19.** Visualization for peptide editing for task 403, higher binding affinity to HLA-B\*08:01 and HLA-A\*11:01.

### G.3 Protein

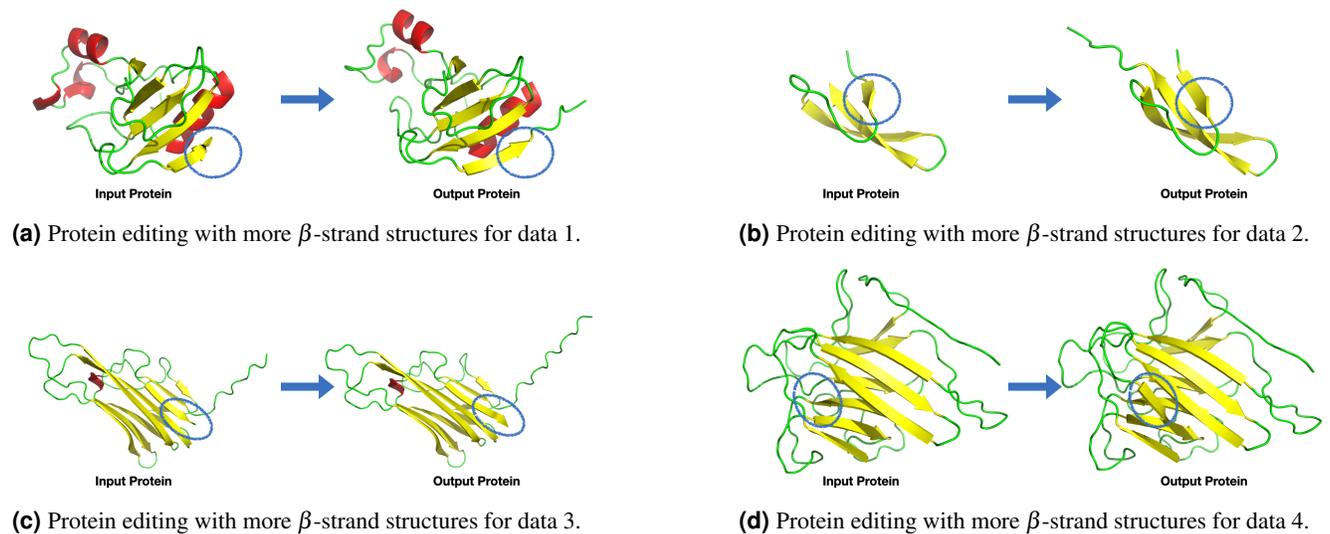
Recall that we consider two types of secondary structures for protein editing tasks. Both the inputs and outputs are protein sequences. Then we use ESMFold [28] for protein folding (protein sequence to protein structure prediction) and then plot the protein structures using PyMOL [59]. For all the protein structure visualizations, we mark  $\alpha$ -helix structures and  $\beta$ -strand structures. The edited regions are highlighted in the blue circles.

**Task 501: edit proteins with more helix structures.**



**Figure 20.** Protein editing with more  $\alpha$ -helix structures.

**Task 502: edit proteins with more strand structures.**



**Figure 21.** Protein editing with more  $\beta$ -strand structures.

## H Ablation Studies

### H.1 Zero-shot and In-context Learning for protein and peptide

In Section 4.4, we conduct an ablation study on small molecules to show the comparison between the zero-shot, in-context learning, and ChatDrug. Here we conduct the same ablation study on peptides and proteins as follows.

**Table 18.** Results on six single-objective and three multi-objective peptide editing tasks. Random Mutation- $R$  for  $R$  mutated positions. The evaluation is the hit ratio of the increased binding affinity score. The best results are marked in **bold**. Due to the space limitation, please check Appendix E for the text prompt of each task.

	single-objective editing						multi-objective editing		
	301	302	303	304	305	306	401	402	403
Random Mutation-1	1.80	14.40	1.80	1.80	12.00	5.60	3.20	0.80	0.40
Random Mutation-2	1.80	13.40	2.80	3.00	8.40	4.40	2.20	0.60	1.20
Random Mutation-3	1.80	9.40	2.40	4.20	9.00	3.80	3.00	0.60	0.80
In-context Learning (few-shot)	24.05	38.40	27.40	32.00	45.50	32.80	29.20	17.47	14.40
ChatDrug ( $C = 0$ , zero-shot)	1.60	16.80	2.40	8.22	15.00	8.02	5.41	2.00	1.20
ChatDrug ( $C = 2$ )	<b>58.60</b>	<b>69.34</b>	<b>58.52</b>	<b>55.11</b>	<b>64.40</b>	<b>62.73</b>	<b>53.71</b>	<b>41.45</b>	<b>54.71</b>

**Table 19.** Results on two protein editing tasks. Random Mutation- $R$  for  $R$  mutated positions. The evaluation is the hit ratio of increased secondary structures accordingly. The best results are marked in **bold**.

	501 more helix	502 more strand
Random Mutation-1	18.32	17.35
Random Mutation-2	24.95	19.69
Random Mutation-3	26.90	21.44
In-context Learning (few-shot)	<b>36.64</b>	44.47
ChatDrug ( $C = 0$ , zero-shot)	21.43	23.73
ChatDrug ( $C = 2$ )	34.79	<b>51.38</b>

## H.2 Ablation Study on the Thresholds in Feedback Condition Function

In the main body, we conduct an ablation study on the thresholds in the feedback condition function. Due to the space limitation, we only list the mean in Table 7. Here we list both the mean and standard deviation as follows.

**Table 20.** Ablation studies on single-objective small molecule editing and feedback condition  $D$  with five seeds and two conversational rounds. The evaluation metric  $E$  uses the strict threshold for each task.

	101	102	103	104	105	106	107	108
loose threshold	80.73±1.32	41.00±0.91	11.23±2.70	16.94±1.24	33.16±2.22	53.59±1.59	14.96±1.96	21.93±1.82
strict threshold	88.67±0.95	70.08±3.44	19.37±5.54	30.99±2.66	43.08±2.95	66.69±2.74	72.60±2.51	76.43±3.32

**Table 21.** Ablation studies on multi-objective small molecule editing and feedback condition  $D$  with five seeds and two conversational rounds. The evaluation metric  $E$  uses the strict threshold for each task.

	201	202	203	204	205	206
loose threshold	20.14±0.86	7.96±2.05	17.93±0.79	5.79±1.38	3.66±0.24	41.04±1.66
strict threshold	49.64±2.66	24.92±4.85	53.64±5.81	24.19±2.19	10.44±5.75	52.9±2.23

### H.3 Ablation Study on the Number of Request Answers in Zero-shot ChatDrug

Notice that in Table 9, we list five molecules (a.k.a. five trials) for each answer. In this subsection, we would like to conduct an ablation study to explore in the zero-shot setting of ChatDrug, *i.e.*, with the conversation round  $C = 0$ , if we can obtain higher performance using more trial numbers. This means that for each input small molecule, we have five edited small molecules; as long as one of them is a hit, then we say this is a successful hit. The results for 14 tasks with the loose threshold are listed below.

**Table 22.** Ablation studies on different trial numbers on single-objective molecule editing, with  $C = 0$  and seed is 0.

	loose condition $\Delta = 0$			strict condition $\Delta > 0$		
	trial = 1	trial = 3	trial = 5	trial = 1	trial = 3	trial = 5
101 <i>more soluble in water</i>	78.26	88.77	93.05	68.48	80.21	85.03
102 <i>less soluble in water</i>	71.35	89.95	93.12	24.16	74.60	78.84
103 <i>more like a drug</i>	16.15	45.64	53.81	2.08	4.62	7.11
104 <i>less like a drug</i>	32.12	68.37	75.00	2.07	17.86	31.12
105 <i>higher permeability</i>	16.04	27.98	33.16	9.63	18.13	22.28
106 <i>lower permeability</i>	8.33	34.04	57.67	5.56	24.47	42.86
107 <i>more hydrogen bond acceptors</i>	59.41	76.57	83.15	1.76	18.29	33.71
108 <i>more hydrogen bond donors</i>	63.16	85.23	89.77	5.85	19.89	32.39

**Table 23.** Ablation studies on different trial numbers on multi-objective molecule editing, with  $C = 0$  and seed is 0.

	loose condition $\Delta = 0$			strict condition $\Delta > 0$		
	trial = 1	trial = 3	trial = 5	trial = 1	trial = 3	trial = 5
201 <i>more soluble in water and more hydrogen bond acceptors</i>	43.09	69.89	75.40	6.08	22.04	34.22
202 <i>less soluble in water and more hydrogen bond acceptors</i>	0.52	13.47	31.44	0.00	0.52	2.06
203 <i>more soluble in water and more hydrogen bond donors</i>	54.49	79.67	81.97	6.18	18.13	35.52
204 <i>less insoluble in water and more hydrogen bond donors</i>	0.53	5.21	13.54	0.00	0.00	1.04
205 <i>more soluble in water and higher permeability</i>	2.11	4.15	5.18	1.05	1.04	1.04
206 <i>more soluble in water and lower permeability</i>	22.22	59.07	69.43	16.40	48.70	55.44

## H.4 Ablation Study on Reasoning and Knowledge Extraction of ChatDrug

In Figure 5, we show the knowledge extraction ability of ChatDrug. As a control experiment, here we test the knowledge extraction ability of ChatDrug without the examples illustrated in the conversation. As shown in Figure 22, the extracted knowledge is more general and less informative when we remove the examples.



**Figure 22.** Knowledge extraction directly without any conversations.

Then as discussed previously, the extracted information in ChatDrug contains redundant information. For example, the one listed in Figure 22 extracts three rules with overlap meanings. Then we conduct a further ablation study by forcing ChatDrug to extract three rules from the five original rules in Figure 5. We observe that ChatDrug successfully merges rule 1 and rule 5, both of which express the same concept that polar functional groups lead to good solubility. Interestingly, ChatDrug also merges rule 3 with rule 4. Though these 2 rules share the idea of ring structure, the essence of rule 3 is introducing polar substituents. Thus, rule 3 should be more relevant to rules 1 and 5. Nevertheless, it is clear that ChatDrug understands the domain knowledge and is capable of extracting and summarizing it. Overall, we would like to give positive feedback on ChatDrug.



**Figure 23.** Further knowledge extraction.

To sum up, we observe that ChatDrug can understand and extract the knowledge to some extent, though not perfectly. We believe this is a promising direction for future exploration.