ChatGPT-powered Conversational Drug Editing Using Retrieval and Domain Feedback

Shengchao Liu [*1 2] **Jiongxiao Wang** [*3] **Yijin Yang** [3] **Chengpeng Wang** [4] **Ling Liu** [5] **Hongyu Guo** [† 6] **Chaowei Xiao** [† 3]

Abstract

Recent advancements in conversational large language models, such as ChatGPT, have demonstrated remarkable promise in various domains, including drug discovery. However, 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 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. Through 10 case studies, we further demonstrate that ChatDrug can identify the key substructures for manipulation, generating diverse and valid suggestions for drug editing.

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 (Sullivan, 2019), including but not limited to virtual screening (Liu et al., 2018; Rohrer & Baumann, 2009), lead optimization (Irwin et al., 2022; Jin et al., 2020; Liu et al., 2022b; Wang et al., 2022), reaction and retrosynthesis (Bi et al., 2021; Gottipati et al., 2020). 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. On the other hand, significant advancements have been made in large language models (LLMs) (Brown et al., 2020; Devlin et al., 2018; Yang et al., 2019b), showcasing exceptional capabilities in understanding human knowledge and exhibiting promising reasoning abilities (Huang et al., 2022; Zhou et al., 2022).

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 is a robust foundation for drug-related tasks. Second, conversational LLMs possess outstanding abilities in fast adaptation and generalization. This adaptability and generalization capacity holds immense potential for addressing complex drug discovery challenges and generating valuable insights. 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 (Mihalić & Trinajstić, 1992), and traditional solutions relying on domain experts for manual editing can be subjective or biased (Drews, 2000; Gomez, 2018). Recent works (Liu et al., 2022a; 2023c) 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. ChatDrug naturally adopts the following potentials of conversational LLMs. 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. To fully verify the effectiveness of ChatDrug, we introduce 39 editing tasks over three common drugs: 14 for small molecules, 11 for peptides, and 2 for proteins. 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

[*] Equal contribution [1] Mila-Québec Artificial Intelligence Institute [2] Université de Montréal [3] Arizona State University [4] University of Illinois Urbana-Champaign [5] Princeton University [6] National Research Council Canada. Correspondence to: Shengchao Liu <liusheng@mila.quebec>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023. Copyright 2023 by the author(s).

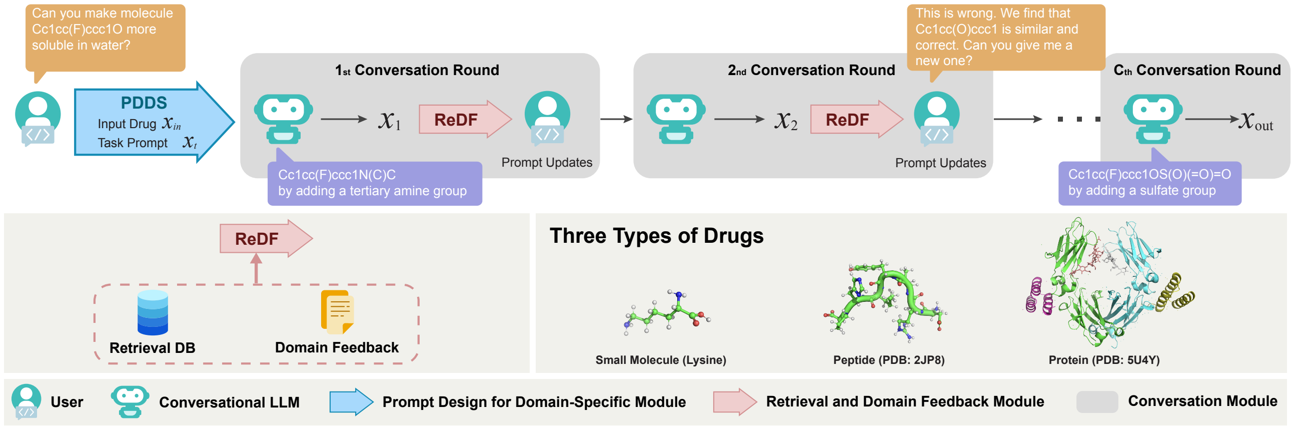

Can you make molecule Cc1cc(F)ccc1O more soluble in water?
PDDS
Input Drug x_{in}
Task Prompt x_t
1st Conversation Round
x_1
ReDF
Prompt Updates
Cc1cc(F)ccc1N(C)C by adding a tertiary amine group
2nd Conversation Round
x_2
ReDF
Prompt Updates
This is wrong. We find that Cc1cc(O)ccc1 is similar and correct. Can you give me a new one?
Cth Conversation Round
x_{out}
Cc1cc(F)ccc1OS(O)(=O)=O by adding a sulfate group
ReDF
Retrieval DB
Domain Feedback
Three Types of Drugs
Small Molecule (Lysine)
Peptide (PDB: 2JP8)
Protein (PDB: 5U4Y)
User
Conversational LLM
Prompt Design for Domain-Specific Module
Retrieval and Domain Feedback Module
Conversation Module

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.

studies, illustrating that ChatDrug can successfully identify the important substructures for each type of drug.

2. 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.

Data Structure of Drugs. In this paper, we would like to explore the three most common drugs: small molecules (Jayatunga et al., 2022), proteins (Frokjaer & Otzen, 2005), and peptides (Craik et al., 2013). Small molecules use SMILES strings (Weininger, 1988) and molecular graphs (Duvenaud et al., 2015; Kearnes et al., 2016; Liu et al., 2019). 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. Peptides are short chains of amino acids and can be viewed as a special type of protein. The three data structures are demonstrated in Figure 1.

Drug Editing and Problem Formulation. 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 $\boldsymbol{x}_{\text{in}}$, and a target or desired property in the textual description is also known as the *text prompt* $\boldsymbol{x}_t$ in literature (Liu et al., 2023a; Raffel et al., 2020). Then the goal is to optimize the drug:

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

Then an evaluation metric $E(\boldsymbol{x}_{\text{in}}, \boldsymbol{x}_{\text{out}}; \boldsymbol{x}_t) \in \{\text{True}, \text{False}\}$ 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*.

2.1. 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 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 in Equation (1). Here the text prompts $\boldsymbol{x}_t$ should be specifically designed to enable the generalization for domain-specific tasks with computationally feasible metrics. 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.

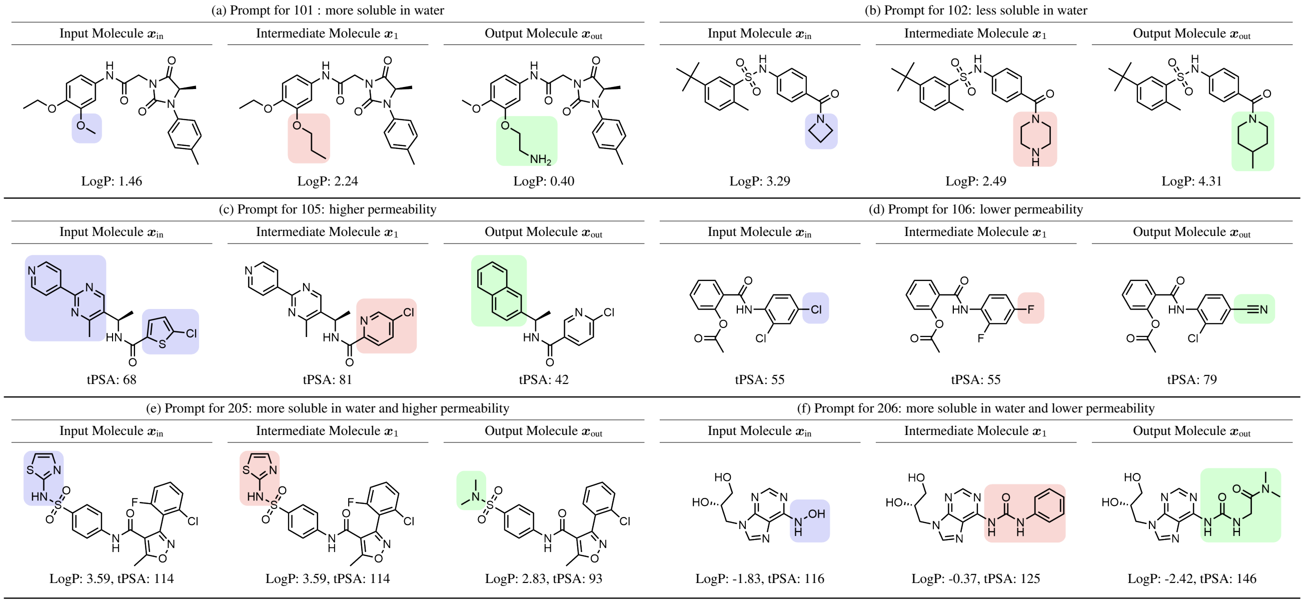
2.2. ReDF Module

To better utilize the domain knowledge, we propose an important module: the ReDF (retrieval and domain feedback) module. For each input drug $\boldsymbol{x}_{\text{in}}$ and prompt $\boldsymbol{x}_t$, we have a candidate drug $\tilde{\boldsymbol{x}}$, which does not satisfy the desired property change in $\boldsymbol{x}_t$. 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 2.3). Based on these, ReDF will return a drug $\boldsymbol{x}_R$ satisfying:

$$\boldsymbol{x}_R = \underset{\boldsymbol{x}'_R \in \text{Retrieval DB}}{\arg\max} \langle \tilde{\boldsymbol{x}}, \boldsymbol{x}'_R \rangle \wedge D(\boldsymbol{x}_{\text{in}}, \boldsymbol{x}'_R; \boldsymbol{x}_t), \tag{2}$$

where $D(\cdot, \cdot; \cdot)$ is the domain feedback function, and $\langle \tilde{\boldsymbol{x}}, \boldsymbol{x}'_R \rangle$ is the similarity function. We use Tanimoto similarity (Bajusz et al., 2015) for small molecules and Levenshtein distance for peptides and proteins. Notice that here we take $D(\cdot, \cdot; \cdot)$ the same as evaluation metric $E(\cdot, \cdot; \cdot)$. Then the ReDF module injects $\boldsymbol{x}_R$ into a new prompt, *e.g.*, the updated prompt is *"Your provided sequence [$\tilde{\boldsymbol{x}}$] is not correct. We find a sequence [$\boldsymbol{x}_R$] which is correct and similar to the molecule you provided. Can you give me a new molecule?"*

Table 1. Visualization of six small molecule editing tasks. The blue regions , red regions , and green regions correspond to the edited substructures in the input molecule $\boldsymbol{x}_{\text{in}}$, intermediate molecule $\boldsymbol{x}_1$ for the 1st conversation round, and the output molecule $\boldsymbol{x}_{\text{out}}$.


(a) Prompt for 101 : more soluble in water
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 1.46
LogP: 2.24
LogP: 0.40
(b) Prompt for 102: less soluble in water
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 3.29
LogP: 2.49
LogP: 4.31
(c) Prompt for 105: higher permeability
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
tPSA: 68
tPSA: 81
tPSA: 42
(d) Prompt for 106: lower permeability
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
tPSA: 55
tPSA: 55
tPSA: 79
(e) Prompt for 205: more soluble in water and higher permeability
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 3.59, tPSA: 114
LogP: 3.59, tPSA: 114
LogP: 2.83, tPSA: 93
(f) Prompt for 206: more soluble in water and lower permeability
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: -1.83, tPSA: 116
LogP: -0.37, tPSA: 125
LogP: -2.42, tPSA: 146

2.3. Conversation Module

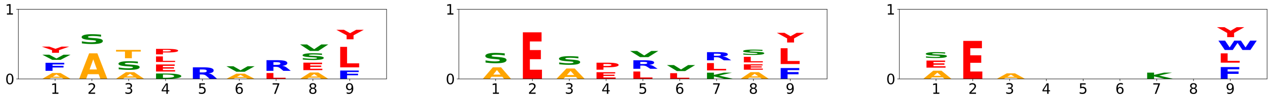
Another appealing attribute of conversational LLMs (like ChatGPT) is the interactive capability. This enables the LLMs to iteratively update the results by injecting prior knowledge. Inspired by this, we also consider adapting the conversational strategy for ChatDrug, which can naturally fit the ReDF module as described in in Section 2.2. Then concretely on this conversational strategy in ChatDrug, first suppose there are C conversation rounds, and we have an edited drug $\boldsymbol{x}_c$ for the conversation round c. If $\boldsymbol{x}_c$ satisfies our condition in the task prompt, then ChatDrug will exit. Otherwise, users will tell ChatDrug that $\boldsymbol{x}_c$ is wrong, and we need to retrieve another similar but correct drug from the retrieval DB using ReDF: $\boldsymbol{x}_R = \text{ReDF}(\boldsymbol{x}_{\text{in}}, \boldsymbol{x}_c)$, with $\tilde{\boldsymbol{x}} = \boldsymbol{x}_c$ in Equation (2).

3. Experiment

Specifications for ChatDrug. We verify the effectiveness of ChatDrug 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, 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 the appendix for the full list.

3.1. Text-guided Molecule Property Editing

We adopt 16 single-objective tasks and 12 multi-objective editing tasks from MoleculeSTM (Liu et al., 2022a). **Data**: Both the input molecules and retrieval DB are sampled from ZINC (Irwin et al., 2012): we sample 200 and 10K molecules (with SMILES strings) from ZINC as input molecules and retrieval DB, respectively. **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 using RDKit (Landrum et al., 2013). Another important argument is the threshold Δ: it is a successful hit if the difference between input and output properties is above the threshold. **Baselines**: The baselines are from (Liu et al., 2022a), based on MegaMolBART (Irwin et al., 2022), 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 2 and 3, respectively. The threshold Δ for each specific task is specified in Table 2; for multi-objective editing tasks in Table 3, the threshold Δ has two values corresponding to the two tasks. We can observe that ChatDrug can reach the best performance on 22 out of 14 tasks. Table 1 visualizes examples of 6 molecule editing tasks where ChatDrug successfully generates output molecules $\boldsymbol{x}_{\text{out}}$ with desirable property change, while the output of the first conversation round $\boldsymbol{x}_1$ fail. For example, in Table 1a, $\boldsymbol{x}_1$ converts a methyl group to a propyl which incorrectly yields a less soluble molecule. Through conversational guidance, ChatDrug changes its output $\boldsymbol{x}_{\text{out}}$ to an aminoethyl group, successfully fulfilling the task.

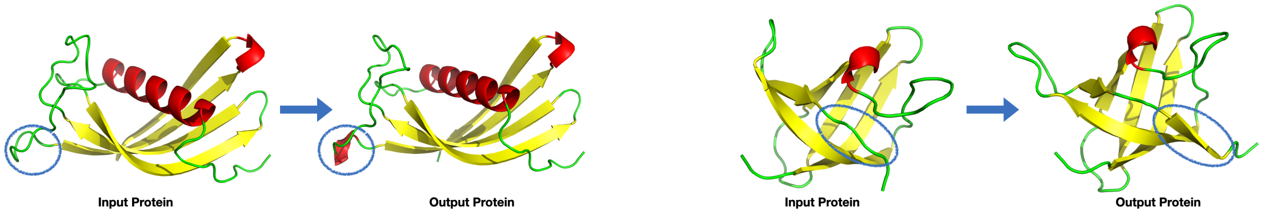


(a) Motifs of input peptides for 301.

(b) Motifs of edited peptides for 301.

(c) Motifs of experimental peptides for 301.

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.



(a) Protein editing with more helix structures.

(b) Protein editing with more strand structures.

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

3.2. 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. To activate CD8+ T cell immune responses, the immunogenic peptides must first bind to Major Histocompatibility Complex (MHC) proteins. **Data**: In this experiment, we use the experimental dataset of peptide-MHC binding affinities (O'Donnell et al., 2020). We follow existing works (Chen et al., 2023) 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. Then we sample 500 peptides from the source allele types. For the retrieval DB, the experimental data of the target allele(s) are adopted. **Evaluation**: The actual bindings require wet-lab experiments, which are expensive and prohibited for large scaled evaluation. Following existing works (Chen et al., 2021; 2023), we leverage the MHCflurry2.0 (O'Donnell et al., 2020) as a pseudo-oracle to predict the peptide-MHC binding affinity. 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. **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. The edited peptides follow similar patterns to the experimental data. For instance, for task 301, the edited peptides can successfully upweight the alphabet E (glutamic acid) at position 2.

3.3. Text-guided Protein Secondary Structure Editing

Last but not least, we consider text-guided protein secondary structure editing (PSSE) (Klausen et al., 2019). 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 α-helix and β-sheet, consisting of β-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 (Jumper et al., 2021; Lin et al., 2022). **Data**: TAPE (Rao et al., 2019) 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. **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 (Liu et al., 2023c). 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 (Lin et al., 2022) for protein folding (protein sequence to protein structure prediction) and then plot the protein structures using PyMOL (Schrödinger & DeLano). We show two examples in Figure 3. As circled in the blue regions in Figures 3(a) and 3(b), the edited proteins possess more helix structures and strand structures, respectively. More visualization can be found in Appendix G.

Broader impact

This work studies how to enable ChatGPT for drug editing tasks. We want to emphasize that drug editing (lead optimization or protein design) is generally objective but requires wet lab testing for the most rigorous model assessment, and we would like to leave this for future exploration.

References

Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13, 2015.

Bi, H., Wang, H., Shi, C., Coley, C. W., Tang, J., and Guo, H. Non-autoregressive electron redistribution modeling for reaction prediction. In *ICML*, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chen, Z., Min, M. R., and Ning, X. Ranking-based convolutional neural network models for peptide-mhc binding prediction. *Frontiers in molecular biosciences*, 2021.

Chen, Z., Zhang, B., Guo, H., Emani, P., Clancy, T., Jiang, C., Gerstein, M., Ning, X., Cheng, C., and Min, M. R. Binding peptide generation for mhc class i proteins with deep reinforcement learning. *Bioinformatics*, 39(2): btad055, 2023.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Craik, D. J., Fairlie, D. P., Liras, S., and Price, D. The future of peptide-based drugs. *Chemical biology & drug design*, 81(1):136–147, 2013.

Demirel, M. F., Liu, S., Garg, S., Shi, Z., and Liang, Y. Attentive walk-aggregating graph neural networks. *Transactions on Machine Learning Research*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Drews, J. Drug discovery: a historical perspective. *Science*, 287(5460):1960–1964, 2000.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

Edwards, C., Zhai, C., and Ji, H. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 595–607, 2021.

Edwards, C., Lai, T., Ros, K., Honke, G., and Ji, H. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

Frokjaer, S. and Otzen, D. E. Protein drug stability: a formulation challenge. *Nature reviews drug discovery*, 4 (4):298–306, 2005.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Gomez, L. Decision making in medicinal chemistry: The power of our intuition. *ACS Medicinal Chemistry Letters*, 9(10):956–958, 2018.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, pp. 3668–3679. PMLR, 2020.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., and Han, J. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

Ingraham, J., Baranov, M., Costello, Z., Frappier, V., Ismail, A., Tie, S., Wang, W., Xue, V., Obermeyer, F., Beam, A., et al. Illuminating protein space with a programmable generative model. *bioRxiv*, 2022. doi: 10.1101/2022.12.01.518682.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

Irwin, R., Dimitriadis, S., He, J., and Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P. S., et al. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.

Jayatunga, M. K., Xie, W., Ruder, L., Schulze, U., and Meier, C. Ai in small-molecule drug discovery: A coming wave. *Nat. Rev. Drug Discov*, 21:175–176, 2022.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pp. 4839–4848. PMLR, 2020.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30:595–608, 2016.

Klausen, M. S., Jespersen, M. C., Nielsen, H., Jensen, K. K., Jurtz, V. I., Soenderby, C. K., Sommer, M. O. A., Winther, O., Nielsen, M., Petersen, B., et al. NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

Landrum, G. et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Liu, S., Alnammi, M., Ericksen, S. S., Voter, A. F., Ananiev, G. E., Keck, J. L., Hoffmann, F. M., Wildman, S. A., and Gitter, A. Practical model selection for prospective virtual screening. *Journal of chemical information and modeling*, 59(1):282–293, 2018.

Liu, S., Demirel, M. F., and Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.

Liu, S., Nie, W., Wang, C., Lu, J., Qiao, Z., Liu, L., Tang, J., Xiao, C., and Anandkumar, A. Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789*, 2022a.

Liu, S., Wang, C., Nie, W., Wang, H., Lu, J., Zhou, B., and Tang, J. GraphCG: Unsupervised discovery of steerable factors in graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022b. URL `https://openreview.net/forum?id=BhR44NzeK_1`.

Liu, S., Du, W., Li, Y., Li, Z., Zheng, Z., Duan, C., Ma, Z., Yaghi, O., Anandkumar, A., Borgs, C., et al. Symmetry-informed geometric representation for molecules, proteins, and crystalline materials. *arXiv preprint arXiv:2306.09375*, 2023b.

Liu, S., Zhu, Y., Lu, J., Xu, Z., Nie, W., Gitter, A., Xiao, C., Tang, J., Guo, H., and Anandkumar, A. A text-guided protein design framework. *arXiv preprint arXiv:2302.04611*, 2023c.

Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Mihalić, Z. and Trinajstić, N. A graph-theoretical approach to structure-property relationships. *Journal of Chemical Education*, 69(9):701, 1992.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 1045–1048. Makuhari, 2010.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

O'Donnell, T. J., Rubinsteyn, A., and Laserson, U. Mhcflurry 2.0: improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. *Cell systems*, 11(1):42–48, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems*, 32, 2019.

Rohrer, S. G. and Baumann, K. Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *Journal of Chemical Information and Modeling*, 49(2):169–184, 2009. doi: 10.1021/ci8002649. URL https://doi.org/10.1021/ci8002649. PMID: 19161251.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.

Schrödinger, L. and DeLano, W. Pymol. URL http://www.pymol.org/pymol.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Schütt, K. T., Unke, O. T., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

Su, B., Du, D., Yang, Z., Zhou, Y., Li, J., Rao, A., Sun, H., Lu, Z., and Wen, J.-R. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*, 2022.

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.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Nie, W., Qiao, Z., Xiao, C., Baraniuk, R., and Anandkumar, A. Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126*, 2022.

Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019a.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019b.

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

Zhang, Y., Chen, Q., Zhang, Y., Wei, Z., Gao, Y., Peng, J., Huang, Z., Sun, W., and Huang, X.-J. Automatic term name generation for gene ontology: task and dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4705–4710, 2020.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

Zhu, X., Sobhani, P., and Guo, H. Long short-term memory over recursive structures. In *ICML*, 2015.

A. Main Results

Due to the space limitation, we leave the main results in this section.

Table 2. Results on eight 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	Δ	Random	PCA	High Variance	GS-Mutate	MoleculeSTM (SMILES)	MoleculeSTM (Graph)	ChatDrug (Ours)
101 *more soluble in water*	0	35.33 ± 1.31	33.80 ± 3.63	33.52 ± 3.75	52.00 ± 0.41	61.87 ± 2.67	67.86 ± 3.46	**94.13±1.04**
	0.5	11.04 ± 2.40	10.66 ± 3.24	10.86 ± 2.56	14.67 ± 0.62	49.02 ± 1.84	54.44 ± 3.99	**88.67±0.95**
102 *less soluble in water*	0	43.36 ± 3.06	39.36 ± 2.55	42.89 ± 2.36	47.50 ± 0.41	52.71 ± 1.67	64.79 ± 2.76	**96.86±1.10**
	0.5	19.75 ± 1.56	15.12 ± 2.93	18.22 ± 0.33	12.50 ± 0.82	30.47 ± 3.26	47.09 ± 3.42	**70.08±3.44**
103 *more like a drug*	0	38.06 ± 2.57	33.99 ± 3.72	36.20 ± 4.34	28.00 ± 0.71	36.52 ± 2.46	39.97 ± 4.32	**48.65±3.39**
	0.1	5.27 ± 0.24	3.97 ± 0.10	4.44 ± 0.58	6.33 ± 2.09	8.81 ± 0.82	14.06 ± 3.18	**19.37±5.54**
104 *less like a drug*	0	36.96 ± 2.25	35.17 ± 2.61	39.99 ± 0.57	71.33 ± 0.85	58.59 ± 1.01	**77.62 ± 2.80**	70.75±2.92
	0.1	6.16 ± 1.87	5.26 ± 0.95	7.56 ± 0.29	27.67 ± 3.79	37.56 ± 1.76	**54.22 ± 3.12**	30.99±2.66
105 *higher permeability*	0	25.23 ± 2.13	21.36 ± 0.79	21.98 ± 3.77	22.00 ± 0.82	57.74 ± 0.60	**59.84 ± 0.78**	56.56±1.84
	10	17.41 ± 1.43	14.52 ± 0.80	14.66 ± 2.13	6.17 ± 0.62	47.51 ± 1.88	**50.42 ± 2.73**	43.08±2.95
106 *lower permeability*	0	16.79 ± 2.54	15.48 ± 2.40	17.10 ± 1.14	28.83 ± 1.25	34.13 ± 0.59	31.76 ± 0.97	**77.35±1.98**
	10	11.02 ± 0.71	10.62 ± 1.86	12.01 ± 1.01	15.17 ± 1.03	26.48 ± 0.97	19.76 ± 1.31	**66.69±2.74**
107 *more hydrogen bond acceptors*	0	12.64 ± 1.64	10.85 ± 2.29	11.78 ± 0.15	21.17 ± 3.09	54.01 ± 5.26	37.35 ± 0.79	**95.35±0.62**
	1	0.69 ± 0.01	0.90 ± 0.84	0.67 ± 0.01	1.83 ± 0.47	27.33 ± 2.62	16.13 ± 2.87	**72.60±2.51**
108 *more hydrogen bond donors*	0	2.97 ± 0.61	3.97 ± 0.55	6.23 ± 0.66	19.50 ± 2.86	28.55 ± 0.76	60.97 ± 5.09	**96.54±1.31**
	1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.33 ± 0.24	7.69 ± 0.56	32.35 ± 2.57	**76.43±3.32**

Table 3. Results on six 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	Δ	Random	PCA	High Variance	GS-Mutate	MoleculeSTM (SMILES)	MoleculeSTM (Graph)	ChatDrug (Ours)
201 *more soluble in water* and	0 – 0	9.88 ± 1.03	8.64 ± 2.06	9.09 ± 1.25	14.00 ± 2.48	27.87 ± 3.86	27.43 ± 3.41	**79.62±0.64**
more hydrogen bond acceptors	0.5 – 1	0.23 ± 0.33	0.45 ± 0.64	0.22 ± 0.31	0.67 ± 0.62	8.80 ± 0.04	11.10 ± 1.80	**49.64±2.66**
202 *less soluble in water* and	0 – 0	2.99 ± 0.38	2.00 ± 0.58	2.45 ± 0.67	7.17 ± 0.85	8.55 ± 2.75	8.21 ± 0.81	**51.59±3.79**
more hydrogen bond acceptors	0.5 – 1	0.45 ± 0.32	0.00 ± 0.00	0.22 ± 0.31	0.17 ± 0.24	2.93 ± 0.30	0.00 ± 0.00	**24.92±4.85**
203 *more soluble in water* and	0 – 0	2.28 ± 1.15	2.23 ± 1.16	4.44 ± 0.58	13.83 ± 2.95	33.51 ± 4.08	49.23 ± 1.71	**89.34±0.96**
more hydrogen bond donors	0.5 – 1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	9.98 ± 1.03	23.94 ± 1.09	**53.64±5.81**
204 *less insoluble in water* and	0 – 0	0.69 ± 0.58	1.96 ± 0.87	1.79 ± 0.66	5.67 ± 0.62	17.03 ± 2.75	14.42 ± 3.43	**39.90±3.86**
more hydrogen bond donors	0.5 – 1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	2.59 ± 1.14	3.84 ± 0.71	**24.19±2.19**
205 *more soluble in water* and	0 – 0	5.06 ± 1.21	3.53 ± 0.38	4.88 ± 2.21	8.17 ± 1.03	35.69 ± 3.19	**39.74 ± 2.26**	12.85±2.68
higher permeability	0.5 – 10	1.16 ± 0.68	0.67 ± 0.55	0.66 ± 0.54	0.00 ± 0.00	19.15 ± 0.73	**22.66 ± 1.90**	10.44±5.75
206 *more soluble in water* and	0 – 0	12.17 ± 1.05	10.43 ± 2.88	13.08 ± 2.28	19.83 ± 2.46	44.35 ± 0.68	30.87 ± 0.62	**65.33±2.16**
lower permeability	0.5 – 10	6.20 ± 0.64	6.23 ± 2.31	6.67 ± 0.53	4.83 ± 0.85	28.67 ± 2.22	20.06 ± 1.26	**52.9±2.23**

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.

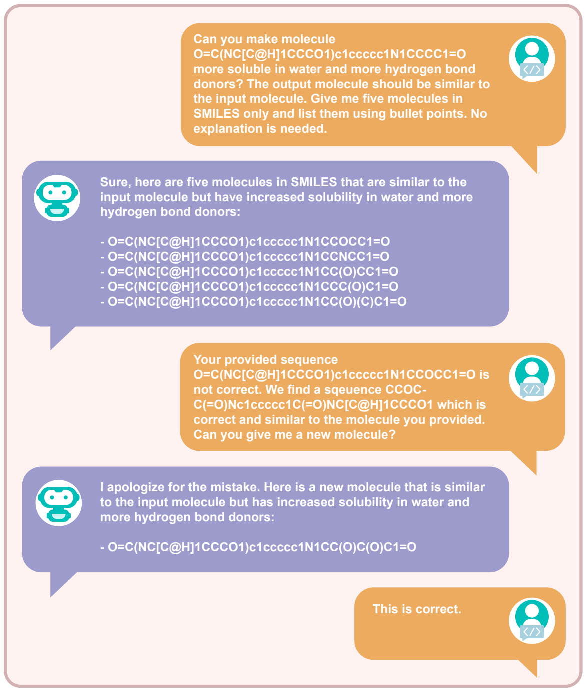
	single-objective editing						multi-objective editing		
	301	302	303	304	305	306	401	402	403
Random Mutation-1	1.80	14.40	1.80	1.80	12.00	5.60	3.20	0.80	0.40
Random Mutation-2	1.80	13.40	2.80	3.00	8.40	4.40	2.20	0.60	1.20
Random Mutation-3	1.80	9.40	2.40	4.20	9.00	3.80	3.00	0.60	0.80
ChatDrug	**58.60**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

B. Examples of ChatDrug

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

Table 5. Results on protein editing tasks. Random Mutation-R for R mutated positions. The evaluation is the hit ratio of increased secondary structures. 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	**34.79**	**51.38**



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.

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)c1ccccc1N1CCC(O)C1=O
- O=C(NC[C@H]1CCCO1)c1ccccc1N1CC(O)(C)C1=O

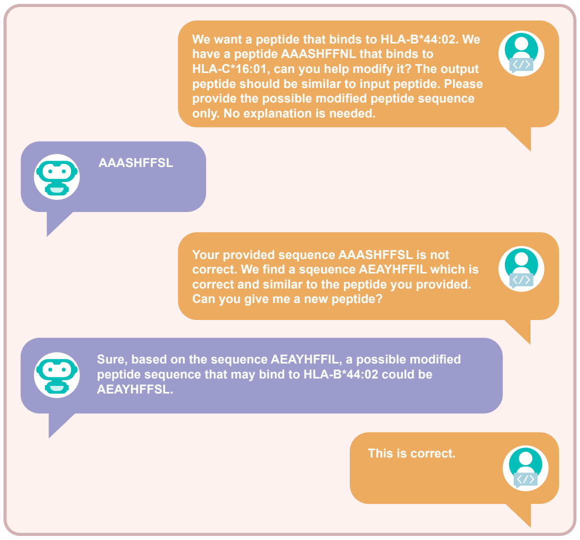
Your provided sequence O=C(NC[C@H]1CCCO1)c1ccccc1N1CCOCC1=O is not correct. We find a sqeuence CCOC-C(=O)Nc1ccccc1C(=O)NC[C@H]1CCCO1 which is correct and similar to the molecule you provided. Can you give me a new molecule?

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(O)C1=O

This is correct.

Figure 4. Example of small molecule editing (task 203).



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.

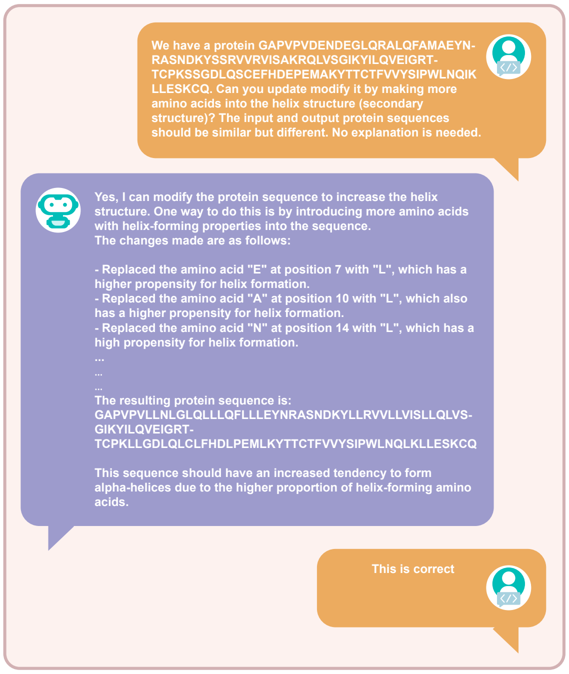
AAASHFFSL

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

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

This is correct.

Figure 5. Example of peptide editing (task 301).



We have a protein GAPVPVDENDEGLQRALQFAMAEYNRASNDKYSSRVVRVISAKRQLVSGIKYILQVEIGRTTCPKSSGDLQSCEFHDEPEMAKYTTCTFVVYSIPWLNQIK
LLESKCQ. 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:
GAPVPVLLNLGLQLLLQFLLLEYNRASNDKYLLRVVLLVISLLQLVSGIKYILQVEIGRTTCPKLLGDLQLCLFHDLPEMLKYTTCTFVVYSIPWLNQLKLLESKCQ

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 6. Example of protein editing (task 501).

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) (Mikolov et al., 2010; Schuster & Paliwal, 1997), Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997; Zhu et al., 2015), and Gated Recurrent Units (GRU) (Cho et al., 2014), 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 (Vaswani et al., 2017), 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 (Devlin et al., 2018), T5 (Raffel et al., 2020), Generative Pre-trained Transformer (GPT) () and so on. GPT-3 (Brown et al., 2020), 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 (Radford et al., 2019), GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2020), BERT (Devlin et al., 2018) 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 (Chung et al., 2022), FLANPaLM (Chung et al., 2022), and OPT-IML (Iyer et al., 2022), 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) (Ouyang et al., 2022) 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 (Liu et al., 2022a; Zeng et al., 2022): the **internal chemical structure** and **external description**. The internal chemical structure refers to the molecule's structure information, *e.g.*, 1D sequence (SMILES) (Weininger, 1988), 2D molecular graph (Demirel et al., 2021; Duvenaud et al., 2015; Gilmer et al., 2017; Yang et al., 2019a), and 3D geometric graph (Satorras et al., 2021; Schütt et al., 2018; 2021; Thomas et al., 2018). 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 (Zeng et al., 2022) first applies the joint masking auto-encoding on the SMILES string and biomedical textual description. Text2Mol (Edwards et al., 2021) conducts contrastive learning between molecular graph and text data for retrieval tasks between modalities. MolT5 (Edwards et al., 2022) does the translation between SMILES and textual annotation of molecules in a mutual way. MoMu (Su et al., 2022) also conducts contrastive learning while it considers both the retrieval and molecule captioning and text-to-molecule tasks. MoleculeSTM (Liu et al., 2022a) 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 (Madani et al., 2020) 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 (Zhang et al., 2020). ProteinDT (Liu et al., 2023c) is a recent work that addresses this issue with free-text protein design. A parallel work is Chroma (Ingraham et al., 2022), 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 6. 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 7, 9 and 11 for small molecules, peptides, and proteins, respectively.
- We list the corresponding task requirement and allele type information in Tables 8, 10 and 12.
- We further list the prompts of in-context learning in Table 13 for reference.

Table 7. Prompt for small molecule editing. The task requirement can be found in Table 8.

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 8. Task requirement for small molecule editing, corresponding to Table 7.

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 9. Prompt for peptide editing. The source allele target type and target allele type can be found in Table 10.

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 10. Target allele type and source allele type for peptide editing, corresponding to Table 9

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 11. Prompt of Conversation Module for protein editing. The task requirement can be found in Table 12.

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 12. Task requirement for protein editing, corresponding to Table 11.

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 13. 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 (Liu et al., 2022a), 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 2 and 3, we keep the same threshold for domain feedback function D and evaluation function E. The threshold Δ used for each small molecule editing task is shown in Table 14, which holds for both functions.

F.3. Experiments Threshold for Peptide Editing

For the peptide editing task, as mentioned in Section 3, 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:

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

Table 14. Threshold Δ for each small molecule editing task, Δ_1 and Δ_2 represent the threshold of task requirement 1 and task requirement 2, respectively.

Task ID	Loose Threshold		Strict Threshold	
	Δ_1	Δ_2	Δ_1	Δ_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

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 2 and 3, 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.
- 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.

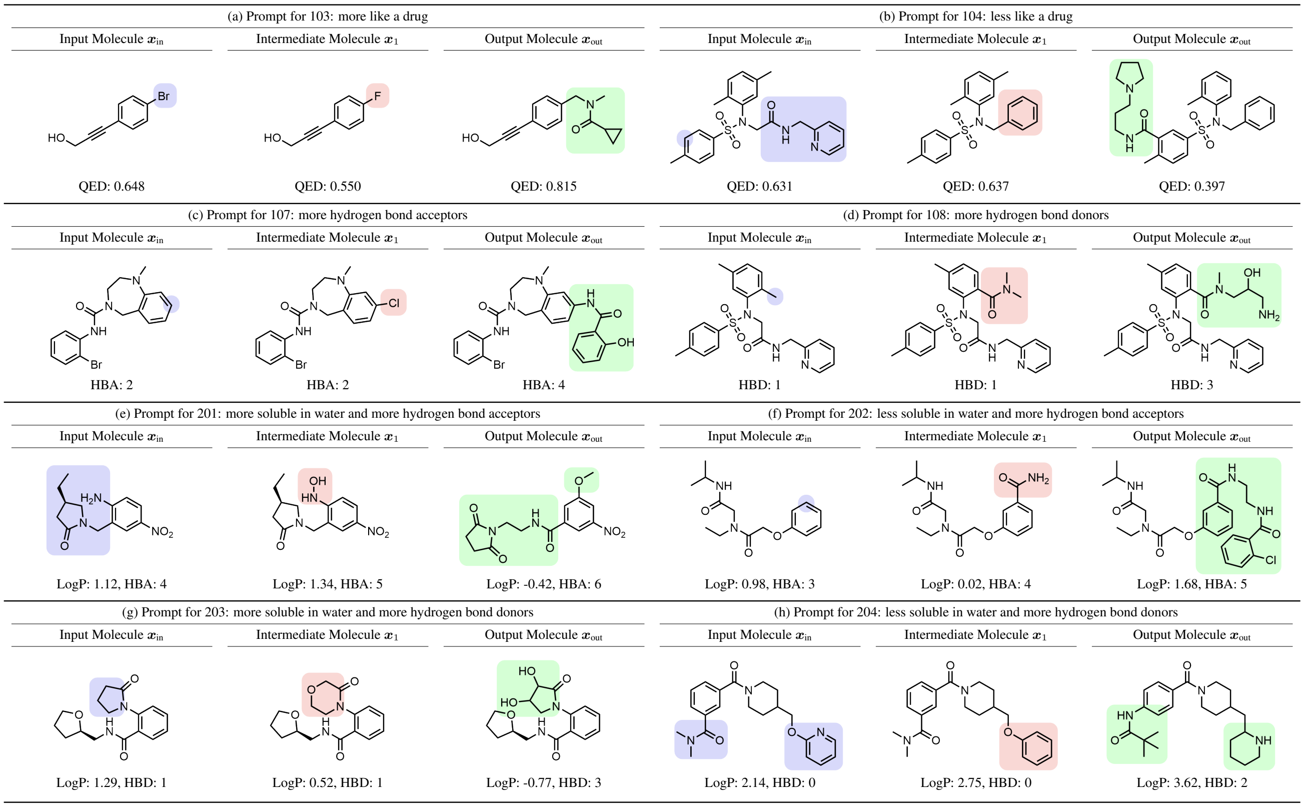
We want to specify that for all the qualitative analyses listed here, we are using $C = 2$ conversation rounds. Especially for small molecules, we consider random seed with 0 and the loose threshold, *i.e.*, $\Delta = 0$ for all tasks.

G.1. Small Molecules

Table 15 visualizes examples of 8 molecule editing tasks where ChatDrug successfully generates output molecules $\boldsymbol{x}_{\text{out}}$ with desirable property change, while the output of the first conversation round $\boldsymbol{x}_1$ fail. In Table 15a and b, $\boldsymbol{x}_{\text{out}}$ successfully adds the desirable fragments to alter the drug likeness of $\boldsymbol{x}_{\text{in}}$, while $\boldsymbol{x}_1$ does so in the wrong direction. In Table 15c, $\boldsymbol{x}_1$ 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 15d, the number of hydrogen bond donors (HBDs) remains in $\boldsymbol{x}_1$ but successfully increases in $\boldsymbol{x}_{\text{out}}$ via insertions of alcohols and amines.

In Table 15e and f, both cases of $\boldsymbol{x}_1$ 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, hydrophibicity is appropriately employed in Table 15f to balance the additional polarity from HBAs, generating a less soluble molecule. In Table 15g and h, both cases of $\boldsymbol{x}_1$ satisfy the solubility requirement but not through the change of HBDs. In $\boldsymbol{x}_{\text{out}}$, the problems are solved by having extra HBDs with further enhanced solubility changes.

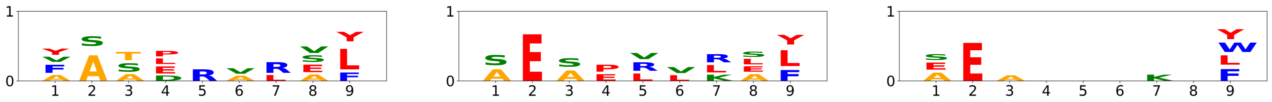
Table 15. Visualization of additional eight small molecule editing cases. The blue regions, red regions, and green regions correspond to the edited substructures in the input molecule $\boldsymbol{x}_{\text{in}}$, intermediate molecule $\boldsymbol{x}_1$ in the 1st conversation round, and the output molecule $\boldsymbol{x}_{\text{out}}$, respectively.


(a) Prompt for 103: more like a drug
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
QED: 0.648
QED: 0.550
QED: 0.815
(b) Prompt for 104: less like a drug
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
QED: 0.631
QED: 0.637
QED: 0.397
(c) Prompt for 107: more hydrogen bond acceptors
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
HBA: 2
HBA: 2
HBA: 4
(d) Prompt for 108: more hydrogen bond donors
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
HBD: 1
HBD: 1
HBD: 3
(e) Prompt for 201: more soluble in water and more hydrogen bond acceptors
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 1.12, HBA: 4
LogP: 1.34, HBA: 5
LogP: -0.42, HBA: 6
(f) Prompt for 202: less soluble in water and more hydrogen bond acceptors
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
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
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 1.29, HBD: 1
LogP: 0.52, HBD: 1
LogP: -0.77, HBD: 3
(h) Prompt for 204: less soluble in water and more hydrogen bond donors
Input Molecule $\boldsymbol{x}_{\text{in}}$
Intermediate Molecule $\boldsymbol{x}_1$
Output Molecule $\boldsymbol{x}_{\text{out}}$
LogP: 2.14, HBD: 0
LogP: 2.75, HBD: 0
LogP: 3.62, HBD: 2

G.2. Peptide

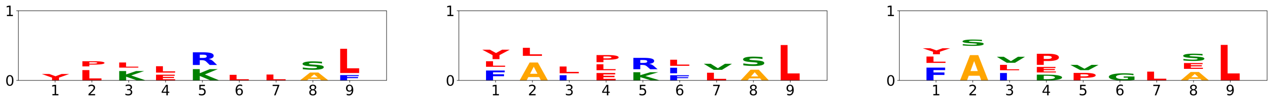
In the main body, we have illustrated how the motif of peptides changes for two peptide editing tasks. Here we show all 6 single-objective editing tasks in Figures 7 to 12.

- For task 301 in Figure 7, ChatDrug can successfully upweight E (Glutamic acid) at position 2.
- For task 302 in Figure 8, ChatDrug can successfully upweight A (Alanine) at position 2, and L (Leucine) at position 9.
- For task 303 in Figure 9, ChatDrug can successfully upweight E (Glutamic acid) at position 2, and L (Leucine) at position 9.
- For task 304 in Figure 10, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9.
- For task 305 in Figure 11, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9.
- For task 306 in Figure 12, ChatDrug can successfully upweight E (Glutamic acid) at position 2, and L (Leucine) at position 9.



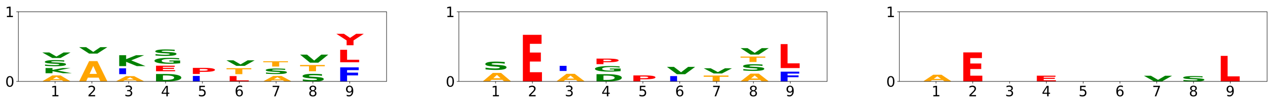
(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 7. Visualization for peptide editing for task 301.



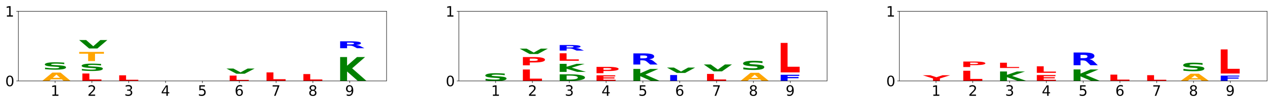
(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 8. Visualization for peptide editing for task 302.



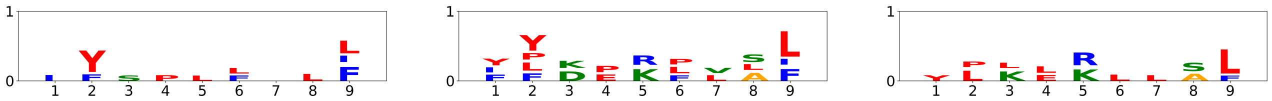
(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 9. Visualization for peptide editing for task 303.



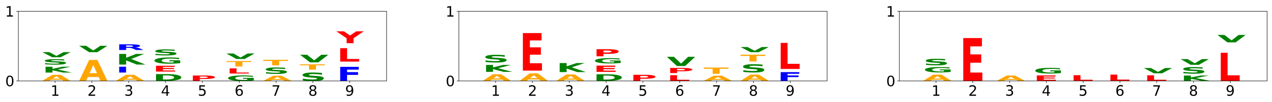
(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 10. Visualization for peptide editing for task 304.



(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 11. Visualization for peptide editing for task 305.

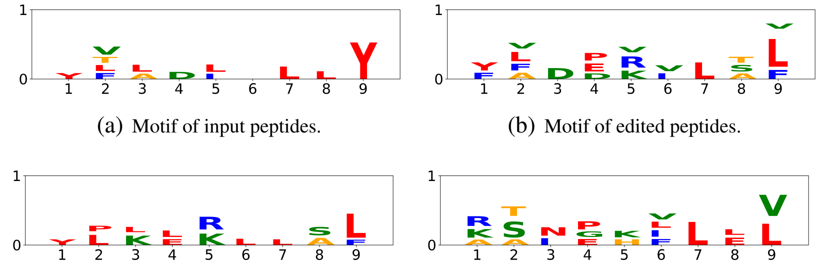


(a) Motif of input peptides. (b) Motif of edited peptides. (c) Motif of experimental peptides.

Figure 12. Visualization for peptide editing for task 306.

Here we show all 3 multi-objective editing tasks in Figures 13 to 15. 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 13, 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 14, 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 15, ChatDrug can successfully upweight R (Arginine) and K (Lysine) at position 5, and L (Leucine) at position 9 for target allele type 1.



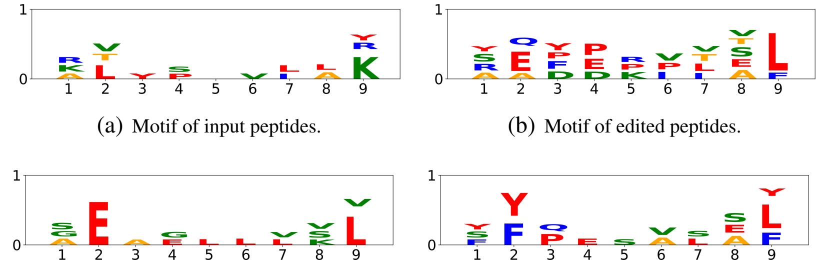
(a) Motif of input peptides.

(b) Motif of edited peptides.

(c) Motif of experimental peptides (target allele 1).

(d) Motif of experimental peptides (target allele 2).

Figure 13. Visualization for peptide editing for task 401.



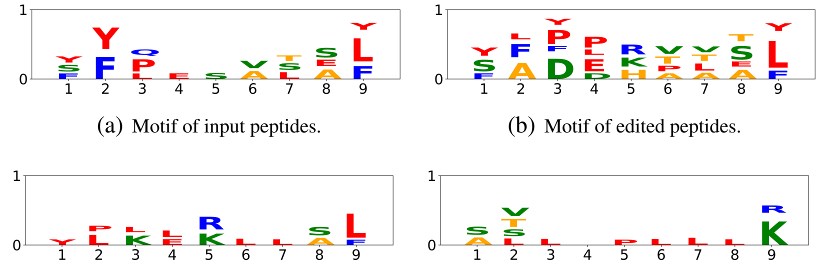
(a) Motif of input peptides.

(b) Motif of edited peptides.

(c) Motif of experimental peptides (target allele 1).

(d) Motif of experimental peptides (target allele 2).

Figure 14. Visualization for peptide editing for task 402.



(a) Motif of input peptides.

(b) Motif of edited peptides.

(c) Motif of experimental peptides (target allele 1).

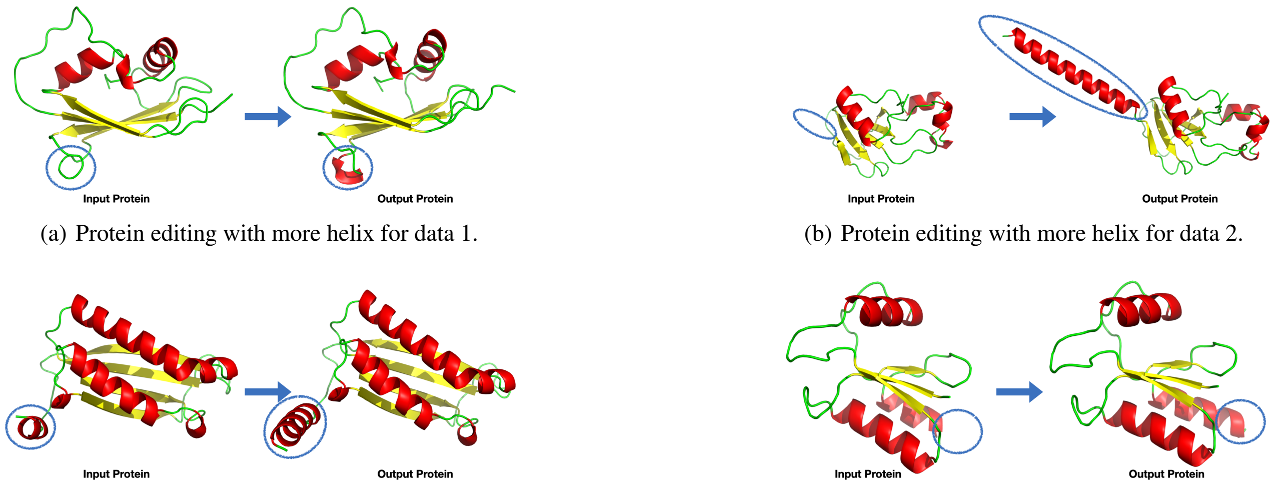
(d) Motif of experimental peptides (target allele 2).

Figure 15. Visualization for peptide editing for task 403.

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 (Lin et al., 2022) for protein folding (protein sequence to protein structure prediction) and then plot the protein structures using PyMOL (Schrödinger & DeLano). For all the protein structure visualizations, we mark α-helix structures and β-strand structures. The edited regions are highlighted in the blue circles.

Task 501: edit proteins with more helix structures.



(a) Protein editing with more helix for data 1.

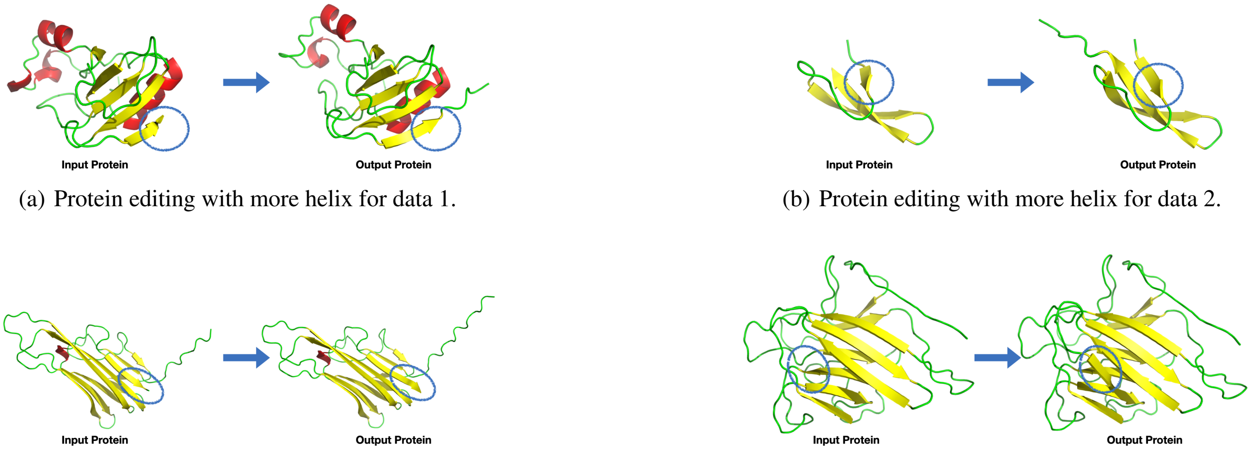
(b) Protein editing with more helix for data 2.

(c) Protein editing with more helix for data 3.

(d) Protein editing with more helix for data 4.

Figure 16. Protein editing with more α-helix structures.

Task 502: edit proteins with more strand structures.



(a) Protein editing with more helix for data 1.

(b) Protein editing with more helix for data 2.

(c) Protein editing with more helix for data 3.

(d) Protein editing with more helix for data 4.

Figure 17. Protein editing with more β-strand structures.

H. Limitation and Conclusion

In this work, we present ChatDrug, a framework that utilizes ChatGPT for drug editing tasks. We build up a benchmark on 27 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 (Liu et al., 2023b). 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.