

# GeLLM<sup>3</sup>O: Generalizing Large Language Models for Multi-property Molecule Optimization

Vishal Dey<sup>1\*</sup>, Xiao Hu<sup>1\*</sup>, Xia Ning<sup>1,2,3,4</sup>

<sup>1</sup> Department of Computer Science and Engineering, The Ohio State University, USA

<sup>2</sup> Translational Data Analytics Institute, The Ohio State University, USA

<sup>3</sup> Department of Biomedical Informatics, The Ohio State University, USA

<sup>4</sup> College of Pharmacy, The Ohio State University, USA

Correspondence: [ning.104@osu.edu](mailto:ning.104@osu.edu)

## Abstract

Despite recent advancements, most computational methods for molecule optimization are constrained to single- or double-property optimization tasks and suffer from poor scalability and generalizability to novel optimization tasks. Meanwhile, Large Language Models (LLMs) demonstrate remarkable out-of-domain generalizability to novel tasks. To demonstrate LLMs' potential for molecule optimization, we introduce MuMOInstruct, the first high-quality instruction-tuning dataset specifically focused on complex multi-property molecule optimization tasks. Leveraging MuMOInstruct, we develop GeLLM<sup>3</sup>O<sub>s</sub>, a series of instruction-tuned LLMs for molecule optimization. Extensive evaluations across 5 in-domain and 5 out-of-domain tasks demonstrate that GeLLM<sup>3</sup>O<sub>s</sub> consistently outperform state-of-the-art baselines. GeLLM<sup>3</sup>O<sub>s</sub> also exhibit outstanding zero-shot generalization to unseen tasks, significantly outperforming powerful closed-source LLMs. Such strong generalizability demonstrates the tremendous potential of GeLLM<sup>3</sup>O<sub>s</sub> as foundational models for molecule optimization, thereby tackling novel optimization tasks without resource-intensive retraining. MuMOInstruct, models, and code are accessible through <https://github.com/ninglab/GeLLMO>.

## 1 Introduction

Drug discovery is a costly and time-consuming process, costing over \$2 billion and a decade (Sertkaya et al., 2024). One of the most critical stages (Hughes et al., 2011) in this process is lead optimization (Sun et al., 2022a), where a molecule with promising bioactivity against a drug target is optimized into a lead molecule by improving multiple molecular properties simultaneously. For example, a hit molecule to treat schizophrenia is optimized such that it can permeate the blood-brain

barrier (Pollak et al., 2018) to reach the DRD2 target (Seeman, 2006) in the brain, while balancing lipophilicity, solubility and toxicity. Improving all these properties together requires balancing multiple trade-offs (Nicolaou and Brown, 2013) and conflicting objectives (Nicolotti et al., 2011), making multi-property optimization extremely challenging.

Most computational methods (Gao et al., 2022) for molecule optimization focus on single- or double-property tasks, leaving multi-property optimization tasks largely unexplored. Existing approaches (Kim et al., 2024; Yang et al., 2021) use predefined fitness and reward functions (Luukkonen et al., 2023), respectively, to model property trade-offs. However, designing such functions for each task demands significant effort and domain expertise. In contrast, other methods (Chen et al., 2021; Wu et al., 2024) obviate the need for such functions, but depend on scarce task-specific data, limiting their scalability and adaptability. Additionally, existing methods lack generalization to unseen tasks, hindering their practical applicability to emerging therapeutic requirements.

Large language models (LLMs) (Naveed et al., 2023) have demonstrated remarkable generalization to unseen tasks across diverse domains (Chang et al., 2024) recently. However, their potential in challenging, multi-property molecule optimization tasks remains largely unidentified. To fully identify LLMs' potential, we introduce MuMOInstruct, the first high-quality instruction-tuning dataset specifically focused on complex, multi-property tasks, each aiming to improve at least 3 properties simultaneously. This is in stark contrast to existing instruction-tuning datasets (Ye et al., 2025) that are limited to single- and double-property tasks.

Leveraging MuMOInstruct, we develop a series of Generalizable LLMs for Multi-property Molecule Optimization, denoted as GeLLM<sup>3</sup>O<sub>s</sub>, by instruction-tuning general-purpose LLMs. Task-specific GeLLM<sup>3</sup>O<sub>s</sub> are fine-tuned on individual

\*Equal Contribution

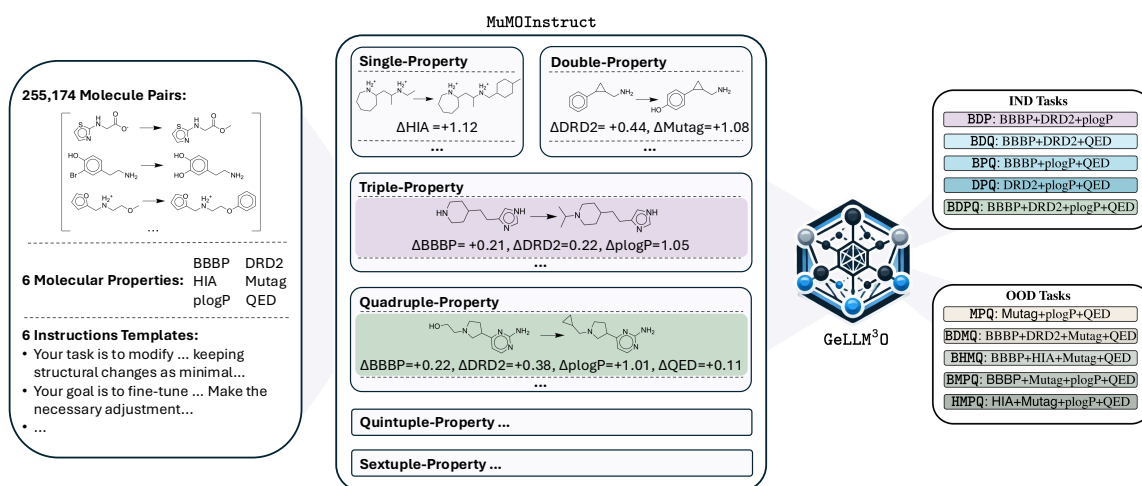


Figure 1: Overview of MuMOInstruct and GeLLM<sup>3</sup>O

tasks, learning precise optimization tailored to specific therapeutic contexts. Generalist GeLLM<sup>3</sup>O<sub>s</sub>, on the other hand, are fine-tuned on multiple tasks which enables them to learn and reason property trade-offs across diverse therapeutic contexts. Moreover, fine-tuning using diverse tasks and instructions enables generalist GeLLM<sup>3</sup>O<sub>s</sub> to effectively handle unseen tasks and instructions.

All GeLLM<sup>3</sup>O models are extensively evaluated against strong general-purpose LLMs, state-of-the-art foundational LLMs for chemistry and task-specific non-LLMs across 5 in-domain (IND) and 5 out-of-domain (OOD) tasks. Our experimental results demonstrate the following key findings:

(1) Both task-specific and generalist GeLLM<sup>3</sup>O<sub>s</sub> significantly outperform state-of-the-art baselines, including powerful closed-source LLMs, across all IND and OOD tasks, with significant improvements of up to 186.6% over the best baselines.

(2) Compared to task-specific GeLLM<sup>3</sup>O<sub>s</sub>, generalist GeLLM<sup>3</sup>O<sub>s</sub> excel on 3 out of 5 IND tasks and demonstrate competitive performance on the other 2 tasks, with remarkable gains of up to 91.3% on more complex tasks, such as BDPQ.

(3) Generalist GeLLM<sup>3</sup>O-P(6) models show superior generalization to OOD tasks outperforming strong baselines by as much as 159.9%.

Figure 1 presents the overall scheme of GeLLM<sup>3</sup>O. To the best of our knowledge, MuMOInstruct is the first large-scale, high-quality instruction-tuning dataset specifically designed for multi-property molecule optimization. Notably, the strong generalization ability of our generalist GeLLM<sup>3</sup>O<sub>s</sub> demonstrates their tremendous potential to accelerate drug discovery by tackling novel optimization tasks without resource-intensive retraining. Dataset,

models, and code are accessible through <https://github.com/ninglab/GeLLM0>.

## 2 Related Work

Various computational approaches have been developed for molecule optimization (You et al., 2018; Blaschke et al., 2020; Xie et al., 2021; Bung et al., 2022; Sun et al., 2022b). For example, Modof (Chen et al., 2021), MIMOSA (Fu et al., 2021), and f-RAG (Lee et al., 2024) perform substructure modifications over molecular graphs. Chemformer (Irwin et al., 2022) and Prompt-MolOpt (Wu et al., 2024) treat optimization as a translation over SMILES (Weininger, 1988) strings, and learn the required modification from molecule pairs. GraphGA (Jensen, 2019) and MolLeo (Wang et al., 2025) leverage genetic algorithms to evolve molecules via genetic algorithm. These methods (Kim et al., 2024; Yang et al., 2021) often require designing non-trivial fitness or reward functions to capture nuanced trade-offs among multiple properties (Niu et al., 2024). Moreover, such methods tends to generate molecules with entirely new scaffolds, limiting their applicability *in vitro* optimization.

Recently, LLMs (Chang et al., 2024) have emerged as a promising option for molecule optimization. For example, ChatDrug (Liu et al., 2024) and Re3DF (Le and Chawla, 2024) leverage LLMs to optimize a molecule iteratively through multi-turn dialogues. DrugAssist (Ye et al., 2025) instruction-tuned Llama2-7B-Chat (Touvron et al., 2023) on each optimization task. While these approaches offer flexible task formulation through natural language, they still face several limitations. ChatDrug incurs high costs due to multiple API

Table 1: Comparison among instruction-tuning datasets for molecular optimization

Comparison	MolOpt-Instructions (Ye et al., 2025)	OpenMolIns (Li et al., 2025)	MuMOInstruct (ours)
Realistic tasks	✗	✗	✓
Multi-prop opt	✓	✗	✓
# properties	5	3	6
# molecule pairs	1,029,949	1,200,000	255,174
# Total tasks	8	3	63
# Train $\geq$ 3 prop	0	0	42
# Eval $\geq$ 3 prop	0	0	10
# IND	0	0	5
# OOD	0	0	5

calls, and instruction-tuning in DrugAssist relies on task-specific data, limiting scalability and adaptability to more complex multi-property tasks.

### 3 MuMOInstruct Dataset

**Comparison among MolOpt-Instructions and MuMOInstruct:** To address these gaps, we introduce MuMOInstruct, the first instruction-tuning dataset specifically focusing on realistic multi-property optimization tasks. Different from MolOpt-Instructions (Ye et al., 2025), which focuses on single- and double-property tasks, MuMOInstruct emphasizes on tasks with at least 3 properties for evaluating LLMs in in-domain and out-of-domain settings. Table 1 highlights the notable differences between the two datasets.

**Problem Definition:** A molecule optimization task is to transform a hit  $M_x$  – a molecule exhibiting initial bio-activity against a therapeutic target into a lead molecule  $M_y$  – an improved molecule for drug development, through structural modification over  $M_x$ , such that (1)  $M_y$  is structurally similar to  $M_x$  (similarity constraint), and (2)  $M_y$  is better than  $M_x$  in terms of all desired properties of interest (property constraint). The desirability of a property is determined by the therapeutic goal, where improved properties indicate more suitable toward a successful drug candidate. For example, for drug candidates targeting the central nervous system (CNS), higher blood-brain barrier permeability (BBBP) is desired to allow the molecules to reach the brain or spinal cord, whereas for those targeting the peripheral nervous system (PNS), lower BBBP is desired instead to prevent the drugs from damaging the CNS. Under the property constraint, the molecule pair  $(M_x, M_y)$  is represented as  $(M_x \prec_{\Delta_p} M_y)_{\forall p \in P}$ , indicating that  $M_y$  is better than  $M_x$  on each property  $p$  of all the desired properties  $P$  with a property-specific

difference  $\Delta_p$ .

In this paper, we introduce MuMOInstruct, the first high-quality instruction-tuning dataset to evaluate models in molecule optimization tasks. Our design is based on four key principles: (1) **Pair-wise optimization:** MuMOInstruct contains a comprehensive set of molecule pairs satisfying the similarity constraint (Tanimoto similarity  $> 0.6$ ) and property constraint over multiple desired properties. Such molecule pairs enable opportunities for molecule optimization models to learn the association between the structural differences and the property improvement among the pairing molecules, and apply such associations for new lead optimization. (2) **Comprehensive coverage:** MuMOInstruct covers more molecular properties, and extends beyond single- and double-property tasks in existing molecular optimization benchmarks. It introduces multi-property optimization tasks that require simultaneous improvement of at least 3 properties, thereby representing complex pharmacological trade-offs in lead optimization. (3) **Real-world relevance:** The tasks in MuMOInstruct are carefully constructed to represent realistic challenges in lead optimization by combining molecular properties key to drug development. For instance, one of the tasks aims to improve intestinal adsorption, toxicity and BBBP – key properties for optimizing orally bioavailable CNS drugs. (4) **Diverse instructions:** MuMOInstruct provides diverse natural language instructions, each describing the optimization task using different phrasings. This prevents LLMs instruction-tuned on MuMOInstruct from overfitting to a specific phrasing and thus, enables them to generalize to unseen instructions, which is crucial in practice to allow different descriptions on optimization tasks.

#### 3.1 Overview of MuMOInstruct Tasks

MuMOInstruct comprises 63 tasks, with 42 tasks aiming to improve at least 3 properties simultaneously, out of which 10 tasks are further divided into IND and OOD tasks (Section 3.5). All tasks in MuMOInstruct are systematically designed by considering combinations from 6 molecular properties: (i) **Penalized LogP (plogP)** representing lipophilicity, balancing permeability, solubility, and metabolic stability – higher plogP is generally desired in drug development; (ii) **Quantitative Estimate of Drug-Likeness (QED)** assessing overall drug-likeness, incorporating multiple molecular

Table 2: Summary of MuMOInstruct Tasks for Evaluation

Type	Task	Properties ( $\Delta_p$ )						#Train	#Val	#Test	#Mols	Cat
		BBBP <sup>†</sup> (0.2)	DRD2 <sup>†</sup> (0.2)	HIA <sup>†</sup> (0.1)	Mutag <sup>†</sup> (0.1)	plogP <sup>†</sup> (1.0)	QED <sup>†</sup> (0.1)					
IND	BDP	✓	✓	-	-	✓	-	2,064	230	500	2,449	CS
	BDQ	✓	✓	-	-	-	✓	4,472	497	500	4,614	CS
	BPQ	✓	-	-	-	✓	✓	4,048	450	500	6,953	CS
	DPQ	-	✓	-	-	✓	✓	2,114	235	500	2,589	CS
	BDPQ	✓	✓	-	-	✓	✓	624	70	500	802	CS
OOD	MPQ	-	-	-	✓	✓	✓	3,132	349	500	5,384	GT
	BDMQ	✓	✓	-	✓	-	✓	601	67	500	791	CS
	BHMQ	✓	-	✓	✓	-	✓	191	22	118	333	CS
	BMPQ	✓	-	-	✓	✓	✓	373	42	191	690	CS
	HMPQ	-	-	✓	✓	✓	✓	234	26	96	417	GT

"#Train", "#Val", "#Test", "#Mols" denote the number of training pairs, validation pairs, test samples, and unique molecules in each task, respectively. "Type" indicates task types, including IND and OOD tasks. "Cat" indicates task category.

attributes related to molecular weight, lipophilicity, and solubility – higher QED indicates better drug-likeness; (iii) **Blood-Brain Barrier Permeability (BBBP)** which refers to the ability of a drug to cross the blood-brain barrier – higher BBBP is desired for CNS drug candidates; (iv) **Mutagenicity (Mutag)** indicating the likelihood of a drug causing genetic mutations – lower Mutag scores are desired to reduce toxicity; (v) **Human Intestinal Absorption (HIA)** which reflects a drug’s ability to be absorbed through the gastrointestinal tract – higher HIA is desired for orally administered drugs; and (vi) **Dopamine Receptor D2 (DRD2) binding affinity** representing the ability of drugs to target dopaminergic pathways – higher DRD2 scores are desired for drugs targeting the DRD2 receptor.

We focus on these properties because they are critical in influencing the pharmacokinetics and safety profiles of molecules, thereby contributing to the development of successful drug candidates. Additionally, these properties are well-studied in the literature and are used in existing benchmarks (Gao et al., 2022). MuMOInstruct provides 10 evaluation tasks which are summarized in Table 2 with details in Appendix A. These tasks can be categorized into two groups based on their significance: (1) **General Drug-Likeness and Toxicity Optimization (GT)**: Focuses on widely studied molecular properties related to drug-likeness, absorption, and toxicity, that are general to any successful drug candidates. (2) **Context-specific Optimization (CS)**: Includes properties relevant to specific disease contexts and therapeutic requirements.

### 3.2 Creation of Task-Specific Training Pairs

We construct task-specific training pairs, where each pair  $(M_x, M_y)$  is sourced from the dataset provided by Chen et al. (2021), which consists of

255K molecule pairs derived from 331K molecules. Each pair differs at only one disconnection site, meaning  $M_x$  can be transformed to  $M_y$  by modifying exactly one fragment. Among these molecule pairs, we select those that satisfy all  $P$  property constraints for a given task optimizing  $P$  properties (i.e.,  $(M_x \prec_{\Delta_p} M_y)_{\forall p \in P}$ ). This ensures that the hit molecule  $M_x$  in each pair requires substantial optimizations, making the selected pairs suitable to model realistic optimization tasks.

### 3.3 Creation of Task-Specific Test Set

We construct a test set by randomly sampling an initial pool of 250K molecules from the ZINC database (Sterling and Irwin, 2015) – a collection of commercially available drug-like molecules – that are not included in the training set. Out of this pool, we select a molecule into the test set of a task which has a property worse than the median among all  $M_x$  in the training pairs (i.e., median property scores of  $M_x$  denoted as  $MPS_{\text{train}}$ ) for each desired property. This provides a task-specific, data-driven selection criteria that is robust to outliers. Additional criteria to exclude outliers are detailed in Appendix A.2. After applying these steps to the initial pool of 250K molecules, we randomly select at most 500 molecules into the test set for each task, with possible overlap across tasks. Table A1 presents the task-specific data set characteristics.

### 3.4 Quality Control

We implement multiple quality control measures as detailed in Appendix A.3. We remove duplicate molecules based on canonicalized SMILES strings. For each molecule, we compute empirical property scores using well-established tools: ADMET-AI (Swanson et al., 2024) and the official implementation provided by You et al. (2018).



Additionally, we provide 6 distinctly phrased (i.e., diverse) instructions for each task (Appendix A.4). To evaluate LLMs’ instruction understanding and generalizability to unseen instructions, we hold out one instruction for each task during training.

### 3.5 IND and OOD Tasks

To distinctly assess the capabilities of instruction-tuned LLMs on both familiar and novel optimization tasks, we categorize our tasks into two groups:

**In-Distribution (IND) tasks:** IND tasks only have property combinations that are used during training. Success in these tasks provides insights on the models’ ability to handle optimizations they are specifically trained on.

**Out-of-Distribution (OOD) tasks:** OOD tasks include novel property combinations that are not used during training. Note that OOD is defined in terms of property combinations and not individual properties, that is, each property is included as part of training tasks. OOD tasks evaluate the models’ ability to generalize to novel optimization tasks without task-specific retraining. This generalizability is crucial in real-world lead optimization, where new therapeutic requirements frequently arise.

## 4 GeLLM<sup>3</sup>O Models

We introduce GeLLM<sup>3</sup>O, a series of general-purpose LLMs instruction-tuned over MuMOInstruct. Through instruction tuning, GeLLM<sup>3</sup>O implicitly learns chemical semantics, structure-property relationships (SPR) (Hansch, 1969) and associations between structural differences expressed in molecule pairs and the desired property improvement expressed via natural language instruction. GeLLM<sup>3</sup>O applies this knowledge to perform structural modifications on a given molecule and generate better molecules with improved properties. Notably, unlike existing methods that rely on carefully designed reward or fitness functions to balance trade-offs among multiple properties (Niu et al., 2024), GeLLM<sup>3</sup>O captures such trade-offs implicitly by learning from multiple paired molecules across diverse multi-property objectives.

We develop both task-specific and generalist GeLLM<sup>3</sup>O models. Task-specific models are trained on a single optimization task, and thus benefit from dedicated training tailored to that specific task. In contrast, generalist models are trained across multiple optimization tasks simultaneously. This multi-task

training enables cross-task knowledge transfer, allowing the generalist GeLLM<sup>3</sup>O to leverage shared chemical knowledge on SPR and multi-property trade-offs across all possible property combinations. Thus, the generalist GeLLM<sup>3</sup>O represents a step toward a foundational model for molecule optimization, capable of handling diverse tasks without task-specific retraining.

We develop a series of generalist GeLLM<sup>3</sup>O models trained on the power sets of 3, 4, and 6 properties, denoted as GeLLM<sup>3</sup>O-P(3), GeLLM<sup>3</sup>O-P(4), and GeLLM<sup>3</sup>O-P(6), respectively. Importantly, the training data for GeLLM<sup>3</sup>O-P(6) excludes all property combinations that appear in the OOD tasks. This ensures that OOD tasks represent unseen optimization scenarios during inference. To train these models, we fine-tune 2 general-purpose LLMs: Mistral-7B-Instruct-v0.3 (AI, 2023) and Llama3.1-8B-Instruct (Grattafiori et al., 2024) by applying LoRA (Hu et al., 2022) adapters to all projection layers and the language modeling head. We perform 0-shot evaluations (i.e., without in-context examples) for all GeLLM<sup>3</sup>O models in all tasks. For each test molecule, we generate 20 molecules via beam search decoding, with the number of beams set to 20. Training details are provided in Appendix B.1.

## 5 Experimental Setup

### 5.1 Baselines

We compare GeLLM<sup>3</sup>O models against 3 categories of baseline models: (1) general-purpose LLMs: Mistral-7B Instruct-v0.3 (AI, 2023), Llama-3.1 8B-Instruct (Touvron et al., 2023), Claude-3.5 (Anthropic, 2024), and GPT-4o (OpenAI, 2024); (2) foundational LLMs for chemistry: an Intern-LM2 with 7B parameters tuned on chemistry and molecular tasks, denoted as ChemLLM (Zhang et al., 2024), and LLaSMol tuned on Mistral-7B, denoted as LLaSMol<sub>Mistral</sub> (Yu et al., 2024), and (3) task-specific non-LLMs: Prompt-MolOpt (Wu et al., 2024). Similarly to GeLLM<sup>3</sup>O models, we generate 20 molecules for each input molecule for all baselines. For LLM baselines that support beam search decoding, we use the same generation strategy as for GeLLM<sup>3</sup>O models. Experimental setups are detailed in Appendix B.2. Prompt templates for LLMs are in Appendix C. Discussions on Prompt-MolOpt and DeepSeek-R1 are in Appendix E and F, respectively.

## 5.2 Evaluation Metrics

We employ multiple evaluation metrics (detailed in Appendix B.3) for a holistic comparison. For brevity and clarity, we present the results only in terms of: **(1) Success Rate (SR)** which is the proportion of input molecules that are successfully optimized with improvement in all desired properties; **(2) Similarity with input (Sim)** which denotes the average Tanimoto similarity (Bajusz et al., 2015) between the optimized and the corresponding input molecule; and **(3) Relative Improvement (RI)** representing the average change in each desired property relative to its initial value in the input molecule. Formally, for a task improving  $\mathcal{P}$  properties, RI is computed as the average of relative change ( $\text{RI}_p$ ) in each property  $p \in \mathcal{P}$  as:

$$\text{RI} = \frac{\sum_{p \in \mathcal{P}} \text{RI}_p}{|\mathcal{P}|},$$

where  $\text{RI}_p$  is computed as:

$$\text{RI}_p = \frac{\mathbb{D}[p](p(M_y) - p(M_x))}{p(M_x)},$$

where  $\mathbb{D}[p]$  is an indicator function denoting whether higher scores of  $p$  is desirable; and  $p(M_x)$  and  $p(M_y)$  denote the score of property  $p$  in the input molecule  $M_x$  and generated molecule  $M_y$ , respectively. Higher SR, Sim, and RI are desirable, indicating more successful optimizations.

## 6 Experimental Results

**Main Findings:** Our experiments reveal the following findings: **(1)** Both task-specific and generalist GeLLM<sup>3</sup>Os consistently outperform general-purpose LLMs, foundational LLMs for chemistry, and task-specific non-LLMs across all IND (Section 6.1) and OOD tasks (Section 6.2), significantly improving SR by as much as 186.6% over the best baseline. **(2)** Compared to task-specific GeLLM<sup>3</sup>Os, generalist GeLLM<sup>3</sup>Os excel on 3 out of 5 IND tasks and demonstrate competitive performance on the other 2 tasks, with remarkable gains of 91.3% in SR on more complex tasks such as BDPQ (Section 6.1). **(3)** Generalist GeLLM<sup>3</sup>0-P(6) models exhibit strong 0-shot generalization to unseen (OOD) tasks and unseen instructions, significantly outperforming powerful general-purpose LLMs (Section 6.2). **(4)** All GeLLM<sup>3</sup>Os substantially outperform the best general-purpose LLM, Claude-3.5 (5-shot) and foundational LLM for chemistry, LLaSMol<sub>Mistral</sub>, across all IND and OOD tasks.

## 6.1 IND Evaluation

Table 3 shows the overall performance of GeLLM<sup>3</sup>Os and baselines across all 5 IND tasks. Detailed results for each task are in Appendix D.1.

**Overall Comparison:** Both task-specific and generalist GeLLM<sup>3</sup>Os significantly outperform all baselines across all IND tasks. Specifically, the generalist GeLLM<sup>3</sup>Os, GeLLM<sup>3</sup>0-P(4)<sub>Mistral</sub> and GeLLM<sup>3</sup>0-P(6)<sub>Mistral</sub>, achieve an average SR of 76.8% and 76.1%, respectively, across all 5 tasks – outperforming the best baseline by 113.2% and 108.8% on average. This is due to the ability of generalist GeLLM<sup>3</sup>Os to leverage knowledge synergistically by optimizing different property combinations, thereby effectively capturing shared chemical principles and property trade-offs. On the most challenging task, BDPQ with 4 properties, the generalist GeLLM<sup>3</sup>0 outperforms all baselines, task-specific GeLLM<sup>3</sup>Os and non-LLMs by as much as 176% in terms of SR, showcasing its strong ability to tackle complex tasks with limited training data. Detailed comparison between GeLLM<sup>3</sup>Os and task-specific non-LLMs are provided in Appendix D.1.

**Comparison between task-specific and generalist GeLLM<sup>3</sup>Os:** As shown in Table 3, generalist GeLLM<sup>3</sup>Os outperform task-specific GeLLM<sup>3</sup>Os on 3 out of 5 IND tasks, particularly with remarkable gains on more complex task BDPQ. On BDPQ the generalist GeLLM<sup>3</sup>0-P(4)<sub>Mistral</sub> achieves a substantial improvement of 91.3% in SR over the best task-specific GeLLM<sup>3</sup>0-4<sub>Mistral</sub>. This is likely due to BDPQ’s complexity in balancing four properties. By leveraging data from other tasks, the generalist GeLLM<sup>3</sup>Os can better capture the nuanced property trade-offs that task-specific GeLLM<sup>3</sup>Os can not with limited task-specific training data.

On the other hand, task-specific GeLLM<sup>3</sup>Os outperform generalist ones on 2 IND tasks (BDP and BDQ), where the properties BBBP and DRD2 are positively correlated (Pearson correlation of 0.6), and there is sufficient training data (2,064 pairs in BDP and 4,472 in BDQ). Notably, DRD2 targets the inhibition of a specific receptor, representing a more specialized therapeutic objective than fundamental molecular properties like pLogP or QED. Therefore, by focusing exclusively on such highly correlated properties and specific therapeutic requirements, task-specific GeLLM<sup>3</sup>Os learn more targeted structural modifications for each task. Nonetheless, this only leads to modest im-

Table 3: Overall Performance in IND Tasks

Model	BDP			BDQ			BPQ			DPQ			BDPQ		
	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>
<b>General-purpose LLMs</b>															
Mistral (0-shot)	6.60	<b>0.81</b>	0.68	3.00	<b>0.76</b>	0.53	15.80	0.73	0.51	2.20	<b>0.65</b>	0.41	3.20	0.77	0.87
Llama (0-shot)	22.00	0.73	0.74	2.20	0.64	0.53	28.40	0.64	0.72	2.60	0.62	0.32	5.20	<b>0.80</b>	0.62
Claude-3.5 (0-shot)	19.60	0.66	1.05	13.00	0.62	1.14	56.00	0.62	0.86	11.00	0.54	0.51	8.00	0.60	1.34
GPT-4o (0-shot)	7.80	0.69	0.90	2.00	0.69	0.62	36.40	0.73	0.42	2.80	0.57	0.50	1.80	0.71	0.39
Mistral (5-shot)	35.20	0.64	2.10	17.00	0.60	2.32	68.60	0.63	0.79	10.40	0.54	1.10	11.00	0.69	0.96
Llama (5-shot)	35.40	0.57	2.71	16.60	0.43	<b>5.70</b>	34.60	0.70	0.64	8.20	0.44	3.02	9.60	0.54	3.45
Claude-3.5 (5-shot)	35.40	0.50	2.43	29.40	0.43	3.80	76.80	0.53	1.24	<u>29.20</u>	<u>0.37</u>	<u>2.87</u>	<u>20.80</u>	<u>0.35</u>	<u>3.53</u>
GPT-4o (1-shot)	9.40	0.69	0.79	7.60	0.66	0.61	40.00	<b>0.75</b>	0.41	7.00	0.62	0.44	3.40	0.70	0.61
<b>Foundational LLMs for Chemistry</b>															
ChemLLM	0.20	0.17	1.20	1.00	0.55	0.82	4.80	0.29	0.96	0.60	0.28	0.42	0.00	n/a	n/a
LlaSMo1 <sub>Mistral</sub>	<u>43.60</u>	<u>0.62</u>	<u>1.09</u>	<u>31.40</u>	<u>0.66</u>	<u>0.93</u>	<u>86.00</u>	<u>0.58</u>	<u>0.84</u>	24.00	0.57	0.61	14.00	0.62	1.03
<b>Task-specific non-LLMs</b>															
Prompt-MolOpt	12.20	0.12	<b>7.46</b>	23.20	0.10	5.40	15.80	0.10	<u>1.50</u>	23.60	0.10	<b>5.46</b>	6.60	0.11	<b>5.36</b>
<b>Task-specific LLMs</b>															
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	84.80	0.47	4.30	87.00	0.47	5.61	93.00	0.46	1.49	62.80	0.37	3.87	-	-	-
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	<b>86.80</b>	<u>0.48</u>	<u>4.38</u>	<b>90.00</b>	<u>0.46</u>	<u>5.66</u>	94.00	<u>0.50</u>	1.38	60.60	0.44	3.76	-	-	-
GeLLM <sup>3</sup> O-4 <sub>Mistral</sub>	71.60	0.49	3.27	57.40	0.55	2.56	90.20	0.46	1.41	54.00	0.44	3.02	<u>30.00</u>	<u>0.48</u>	<u>3.44</u>
GeLLM <sup>3</sup> O-4 <sub>Llama</sub>	53.60	0.63	1.94	48.60	0.59	1.29	93.40	0.59	1.12	39.60	0.57	1.32	28.00	0.66	1.02
Impv-Task (%)	99.1	-22.6	301.8	186.6	-30.3	508.6	9.3	-13.8	64.3	115.1	0.0	34.8	44.2	37.1	-2.5
<b>Generalist LLMs</b>															
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	75.60	0.56	3.31	79.40	0.53	4.52	93.20	0.55	1.23	57.20	0.50	2.22	-	-	-
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	77.40	0.51	3.16	76.40	0.57	4.41	95.40	0.50	1.46	63.40	0.49	2.46	-	-	-
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	81.40	0.55	3.95	82.60	0.56	5.24	96.20	0.52	1.52	<b>66.60</b>	<u>0.53</u>	<u>2.41</u>	<b>57.40</b>	<u>0.52</u>	<u>3.04</u>
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	80.40	0.54	3.60	81.40	0.56	4.81	93.80	0.47	1.64	61.40	0.50	2.02	49.80	0.48	3.26
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>83.00</b>	<u>0.57</u>	<u>3.60</u>	<b>85.80</b>	<u>0.59</u>	<u>4.78</u>	<b>96.80</b>	<u>0.53</u>	1.48	60.80	0.54	2.16	54.00	0.54	3.09
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	77.00	0.53	3.73	79.60	0.56	5.05	95.00	0.47	<b>1.66</b>	57.00	0.49	2.50	52.20	0.49	3.48
Impv-Gen (%)	90.4	-8.1	230.3	173.2	-10.6	414.0	12.6	-8.6	76.2	128.1	43.2	-16.0	176.0	48.6	-13.9

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task. ‘-’ indicates cases where models are trained on 3 properties but the task has additional properties not included in the model training. ‘n/a’ indicates undefined metrics due to no optimized cases (i.e., 0% SR).

improvements of 4.6% and 4.9% over the generalist GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub>. These results collectively highlight the complementary strengths of generalist and task-specific GeLLM<sup>3</sup>O, with the generalist GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> offering a more scalable and foundational model for diverse optimization tasks.

**Comparison between GeLLM<sup>3</sup>O and general-purpose LLMs:** All GeLLM<sup>3</sup>O substantially outperform general-purpose LLMs by a large margin. For example, across 5 IND tasks, GeLLM<sup>3</sup>O-P(4)<sub>Mistral</sub> and GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> achieve a significant average improvement of 128.1% and 124.0% in SR over the best general-purpose LLM baseline, Claude-3.5 (5-shot). This remarkable performance can be attributed to the instruction tuning of GeLLM<sup>3</sup>O on molecule pairs, enabling it to learn modification strategies that general-purpose LLMs fail to acquire through in-context learning. General-purpose LLMs, particularly with 0-shot prompting, exhibit extremely low SR and relatively high Sim, meaning that the very few optimized molecules are highly similar to the

input. In contrast, GeLLM<sup>3</sup>O achieve substantially higher SR and RI, demonstrating more successful optimizations while maintaining reasonable similarity (e.g., Sim in [0.5, 0.6]).

### Comparison between GeLLM<sup>3</sup>O and foundational LLM for chemistry:

All GeLLM<sup>3</sup>O demonstrate significant improvement over the state-of-the-art foundational LLM for chemistry, LlaSMo1<sub>Mistral</sub>, on all IND tasks. For example, GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> outperforms LlaSMo1<sub>Mistral</sub> significantly by 186.6% on BDQ, and 99.1% on BDP, with up to a 5-fold relative improvement (e.g., RI of 508.6 on BDQ) in desired properties. Note that LlaSMo1 was not instruction-tuned on molecule optimization tasks. Thus, the performance gap suggests that the pre-trained chemistry knowledge in foundational LLMs, such as LlaSMo1, is not sufficient to solve tasks as specialized as molecule optimization. This highlights the importance of instruction tuning on MuMOInstruct for acquiring more specialized knowledge.

Table 4: Overall Performance in OOD Tasks

Model	MPQ			BDMQ			BHMQ			BMPQ			HMPQ		
	SR <sup>†</sup>	Sim <sup>†</sup>	RI <sup>†</sup>	SR <sup>†</sup>	Sim <sup>†</sup>	RI <sup>†</sup>	SR <sup>†</sup>	Sim <sup>†</sup>	RI <sup>†</sup>	SR <sup>†</sup>	Sim <sup>†</sup>	RI <sup>†</sup>	SR <sup>†</sup>	Sim <sup>†</sup>	RI <sup>†</sup>
<b>General-purpose LLMs</b>															
Mistral (0-shot)	11.20	0.57	0.48	1.20	0.68	0.37	12.71	0.73	1.90	12.57	0.61	0.54	21.88	<b>0.72</b>	0.72
Llama (0-shot)	25.80	0.44	0.61	1.20	<b>0.76</b>	0.30	11.02	0.74	0.68	16.75	0.51	0.57	15.62	0.47	0.60
Claude-3.5 (0-shot)	17.40	0.49	0.52	15.00	0.57	0.87	38.98	0.51	2.35	44.50	0.55	0.85	38.54	0.54	1.01
GPT-4o (0-shot)	19.40	<b>0.61</b>	0.35	1.60	0.67	0.18	17.80	0.66	0.65	25.13	0.67	0.35	20.83	0.67	0.37
Mistral (5-shot)	59.60	0.54	0.57	20.40	0.59	1.65	34.75	0.70	1.31	49.21	0.62	0.73	46.88	0.66	0.91
Llama (5-shot)	34.80	0.57	0.53	16.80	0.39	<u>3.22</u>	36.44	0.67	1.13	31.94	0.66	0.60	33.33	0.68	0.61
Claude-3.5 (5-shot)	50.60	0.49	<u>0.71</u>	<u>30.40</u>	0.49	<u>2.32</u>	52.54	0.48	<u>2.52</u>	52.36	0.46	<u>1.08</u>	<u>65.62</u>	0.48	<u>1.32</u>
GPT-4o (1-shot)	21.40	0.60	0.48	6.20	0.70	0.75	14.41	<b>0.75</b>	0.67	24.08	<b>0.68</b>	0.45	25.00	0.70	0.54
<b>Foundational LLMs for Chemistry</b>															
ChemLLM	6.20	0.31	0.61	0.00	n/a	n/a	1.69	0.53	0.74	5.24	0.24	0.72	3.12	0.58	0.51
LlaSMo1 <sub>Mistral</sub>	<u>76.40</u>	0.55	0.53	28.20	0.66	0.52	<u>53.39</u>	0.62	1.14	<u>64.92</u>	0.58	0.57	53.12	0.62	0.70
<b>Generalist LLMs</b>															
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>95.20</b>	0.53	0.85	<b>79.00</b>	0.56	3.10	86.44	0.54	2.58	91.10	0.53	1.06	91.67	0.55	1.42
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	93.60	0.48	<b>0.91</b>	74.20	0.55	<b>3.25</b>	<b>93.22</b>	0.49	<b>3.57</b>	<b>95.29</b>	0.49	<b>1.20</b>	<b>97.92</b>	0.46	<b>1.76</b>
Impv-Gen (%)	24.6	-3.6	60.4	159.9	14.3	33.6	74.6	-21.0	213.2	46.8	-15.5	110.5	49.2	-4.2	33.3

The metrics, notations and formatting have the same meanings as those in Table 3.

**Comparison between GeLLM<sup>3</sup>O and Prompt-Mo1Opt:** GeLLM<sup>3</sup>O consistently outperform Prompt-Mo1Opt across all IND tasks, improving SR by as much as 770.0% on BDPQ task. This gain likely arises due to instruction-tuning which enables GeLLM<sup>3</sup>O to effectively learn modification strategies while leveraging their pre-trained general-purpose and chemical knowledge. In contrast, Prompt-Mo1Opt learns both chemical knowledge and task-specific knowledge from scratch (Appendix E.2), making it heavily reliant on limited task-specific training data and thus resulting in extremely low SR. Notably, the very few optimized cases from Prompt-Mo1Opt exhibit high RI but low Sim, indicating substantial property improvements with drastic structural changes. This suggests that Prompt-Mo1Opt tends to generate entirely new molecules, thus failing to retain the core scaffold – a key requirement in lead optimization.

**Comparison between GeLLM<sup>3</sup>O-3 and GeLLM<sup>3</sup>O-4:** GeLLM<sup>3</sup>O-3 is consistently better than GeLLM<sup>3</sup>O-4 in terms of SR across all 4 IND tasks with 3 properties. This performance gap can be attributed to GeLLM<sup>3</sup>O-4’s more constrained training setup, with fewer training pairs (e.g., 624 in BDPQ) and the added complexity of an additional property constraint. As a result, when evaluated on tasks with 3 properties, GeLLM<sup>3</sup>O-4 must adapt its knowledge learned from improving 4 properties, which can limit its effectiveness. In contrast, GeLLM<sup>3</sup>O-3 benefits from more focused task-specific training with larger datasets (e.g., 4,472 in BDQ), enabling better

performance.

## 6.2 OOD Evaluation

Table 4 presents the overall performance of GeLLM<sup>3</sup>O and baselines across all 5 OOD tasks (with detailed results in Appendix D.2). Note that OOD tasks involve novel property combinations excluded from training, making task-specific models and comparisons with GeLLM<sup>3</sup>O-P(3) and GeLLM<sup>3</sup>O-P(4) inapplicable. Generalist GeLLM<sup>3</sup>O demonstrate robust 0-shot generalization to OOD tasks, significantly outperforming all baselines. For instance, both GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> and GeLLM<sup>3</sup>O-P(6)<sub>Llama</sub> achieve very high SR of 88.7% and 90.8%, respectively, on average across all 5 tasks – outperforming strong baselines such as Claude-3.5 (5-shot) and LlaSMo1<sub>Mistral</sub> by as much as 159.9% on task BDMQ. By learning optimization strategies and property trade-offs across diverse tasks during training, generalist GeLLM<sup>3</sup>O develop a flexible understanding of modification strategies that can generalize to novel optimization tasks. This generalizability is crucial in practice, where the dynamic nature of therapeutic requirements requires one unified foundational model capable of handling novel and diverse optimization tasks without task-specific retraining.

## 6.3 Generalizability to Unseen Instructions

Table 5 presents the performance of task-specific GeLLM<sup>3</sup>O and generalist model, GeLLM<sup>3</sup>O-P(6), when prompted with a hold-out instruction and unseen property names (Appendix A.4). Overall,



Table 5: Overall Performance with Unseen Instructions in IND Tasks

Model	Instr	BDP			BDQ			BPQ			DPQ			BDPQ		
		SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>	SR <sup>↑</sup>	Sim <sup>↑</sup>	RI <sup>↑</sup>
<b>Task-specific LLMs</b>																
Mistral	seen	84.80	0.47	4.30	87.00	<b>0.47</b>	5.61	93.00	0.46	1.49	62.80	<b>0.37</b>	3.87	30.00	<b>0.48</b>	3.44
	unseen	<b>89.60</b>	0.45	<b>5.11</b>	87.40	0.44	<b>6.29</b>	93.00	0.45	1.48	64.20	0.35	3.95	<b>32.80</b>	0.45	<b>3.62</b>
Llama	seen	86.80	<b>0.48</b>	4.38	90.00	0.46	5.66	94.00	0.50	1.38	60.60	<b>0.44</b>	3.76	<b>28.00</b>	0.66	1.02
	unseen	85.40	0.44	<b>4.69</b>	90.40	0.46	5.68	93.80	0.49	1.42	63.60	0.39	<b>4.36</b>	24.20	0.64	<b>1.29</b>
<b>Generalist LLMs</b>																
-P(6) <sub>Mistral</sub>	seen	<b>83.00</b>	0.57	<b>3.60</b>	<b>85.80</b>	0.59	<b>4.78</b>	96.80	0.53	1.48	<b>60.80</b>	0.54	<b>2.16</b>	<b>54.00</b>	0.54	<b>3.09</b>
	unseen	75.80	0.59	3.15	80.40	0.59	4.54	96.20	0.54	1.42	54.60	0.55	1.99	49.80	<b>0.57</b>	2.81
-P(6) <sub>Llama</sub>	seen	<b>77.00</b>	0.53	<b>3.73</b>	<b>79.60</b>	0.56	<b>5.05</b>	95.00	0.47	1.66	<b>57.00</b>	0.49	<b>2.50</b>	<b>52.20</b>	0.49	3.48
	unseen	64.60	0.53	3.06	73.40	0.57	4.56	95.60	0.47	1.66	53.60	0.50	2.15	46.40	0.48	3.52

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O is evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

task-specific GeLLM<sup>3</sup>O s retain their performance across all tasks, while, generalist GeLLM<sup>3</sup>O s exhibit a slight drop of 7% in SR on average. This minor drop is expected, since generalist GeLLM<sup>3</sup>O s trained on more property combinations, encounter the same property names more frequently during instruction-tuning. This may lead to subtle overfitting to specific names. Importantly, even with this minor performance drop, generalist GeLLM<sup>3</sup>O s still outperform all baselines by a large margin, (Section 6.1), highlighting their overall superiority. Detailed results are provided in Appendix D.3.

## 6.4 Case Studies

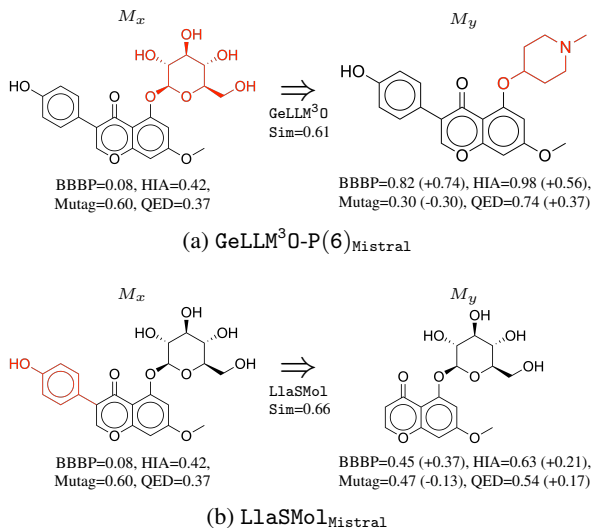


Figure 2: An optimization case on BHMQ. Modifications are highlighted in red.

Figure 2a shows a successful optimization for the OOD task BHMQ, where GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> improves all desired properties by replacing the sugar moiety in  $M_x$  with a nitrogen-containing heterocycle in  $M_y$  (highlighted fragments). The sugar

moiety, with multiple hydroxyl (-OH) groups, increases polarity and hydrogen bonding, limiting passive permeability and leading to low BBBP and HIA (Goetz et al., 2017; Mullard, 2018). Replacing this fragment with a nitrogen heterocycle reduces polarity and hydrogen bonding, leading to improved BBBP (+0.74) and HIA (+0.56). Moreover, hydroxyl-rich sugars in  $M_x$  are prone to oxidation and glycation, compromising stability and bioavailability (Twarda-Clapa et al., 2022; Chen et al., 2024). In contrast, the nitrogen heterocycle in  $M_y$  is a well-known motif for improving metabolic stability and bioavailability (Kerru et al., 2020; Ebenezer et al., 2022), leading to significant improvements in mutagenicity (-0.30) and QED (+0.37). In contrast, LLaSMol<sub>Mistral</sub> retains the sugar moiety and instead removes a phenol group (Figure 2b), leaving polarity and hydrogen bonding largely unaffected, resulting in limited improvements. Additional cases are in Appendix D.4.

## 7 Conclusion

In this work, we introduced MuMOInstruct, the first high-quality instruction-tuning dataset specifically focused on challenging multi-property optimization tasks. Leveraging MuMOInstruct, GeLLM<sup>3</sup>O s achieve state-of-the-art performance across all IND and OOD tasks, notably outperforming strong general-purpose LLMs and foundational LLMs for chemistry. Generalist GeLLM<sup>3</sup>O s demonstrated remarkable generalization to unseen tasks with an average SR of 90.9%, making them promising candidates for foundational models in molecule optimization. This highlights the potential of GeLLM<sup>3</sup>O s to adapt to diverse optimization tasks mirroring dynamic therapeutic requirements.

## 8 Limitations

Despite the strong performance of GeLLM<sup>3</sup>Os as demonstrated in our work, we acknowledge several limitations. **(1)** We did not explore scenarios where users specify precise property-specific improvement thresholds during inference, which could enhance the applicability of GeLLM<sup>3</sup>Os for highly customized therapeutic needs. **(2)** Our evaluations are limited to single-step optimization. We did not explore iterative refinement of generated molecules that could yield even better lead molecules over multiple steps. **(3)** Our construction of MuMOInstruct and evaluation leverages well-established empirical property predictors that are not experimentally validated, which may impact the accuracy of optimization outcomes. **(4)** MuMOInstruct encompasses 6 molecular properties that play a critical role in successful drug design. However, real-world lead optimization often involves additional, more specialized properties and complex trade-offs depending on specific therapeutic requirements. Addressing these limitations in future work could enhance GeLLM<sup>3</sup>Os’ applicability in practice. **(5)** While we evaluate generated molecules on synthetic accessibility via SAS, we do not validate the synthetic routes or retrosynthetic feasibility of generated molecules using wet-lab experiments. We acknowledge that such validation would provide stronger evidence, but such efforts are beyond the scope of this work.

## 9 Impact Statement

Our work introduces the first large-scale, high-quality instruction-tuning dataset, MuMOInstruct, specifically focused on molecule optimization tasks improving at least 3 properties simultaneously. By leveraging MuMOInstruct, we developed a series of instruction-tuned LLMs (GeLLM<sup>3</sup>Os). These models significantly outperform strong closed-source LLMs such as Claude-3.5 as well as foundational LLMs for chemistry on complex multi-property optimization tasks. To the best of our knowledge, our work is the first to introduce a generalist model training framework and a foundational model for molecule optimization. Notably, the robust zero-shot performance of our generalist GeLLM<sup>3</sup>Os demonstrates their potential as foundational models for molecule optimization, offering scalability and adaptability to diverse optimization scenarios.

**Broader Impacts:** The introduction of foundational models capable of handling diverse optimization tasks holds tremendous potential to accelerate drug discovery pipelines. These models offer unparalleled flexibility and scalability, enabling practitioners to adapt them to a wide range of therapeutic requirements without requiring resource-intensive training. By relying solely on an efficient inference process, such models democratize access to advanced optimization capabilities to a broader range of practitioners. This advancement could streamline the identification of novel drug candidates, significantly reducing the cost and time required to develop a new drug.

## 10 Ethics Statement

While MuMOInstruct has been carefully curated to include drug-like, commercially accessible molecules, we cannot guarantee that the dataset is entirely free from inaccuracies or harmful content. We also cannot eliminate the potential of our tuned GeLLM<sup>3</sup>Os to generate undesirable or harmful content (e.g., lethal drugs). We should emphasize that our models are specifically tuned to improve widely used molecule properties aligned with general drug discovery goals, and are not intended for generating toxic or lethal molecules.

The only property in MuMOInstruct that is related to toxicity is mutagenicity, which measures the risk of DNA mutations. Importantly, our models are tuned explicitly to reduce mutagenicity, and not to increase it. Furthermore, GeLLM<sup>3</sup>O models are tuned exclusively on drug-like molecules and optimization objectives aimed at reducing mutagenicity. As a result, they are unlikely to generate molecules with increased toxicity or molecules that can be lethal under a normal dosage.

However, if such molecules can be generated with adversarial prompts, this could potentially arise from the pretrained knowledge of the base models, which includes broader chemical information outside the scope of MuMOInstruct and our instruction-tuning. To mitigate such risks, safeguards such as usage monitoring, and integration with toxicity prediction pipelines should be implemented when deploying these models. Users of our dataset and models are expected to uphold the highest ethical standards and incorporate robust validation pipelines to prevent misuse.

## References

- Mistral AI. 2023. Mistral 7b. *arXiv preprint*.
- Anthropic. 2024. Claude 3.5. <https://www.anthropic.com/claude>.
- Dávid Bajusz, Anita Rácz, and Károly Héberger. 2015. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1).
- Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. 2020. Reinvent 2.0: an ai tool for de novo drug design. *Journal of chemical information and modeling*, 60(12):5918–5922.
- K Bouiahya, A Oulguidoum, A Laghzizil, M Shalabi, Jean-Michel Nunzi, and Sylvie Masse. 2020. Hydrophobic chemical surface functionalization of hydroxyapatite nanoparticles for naphthalene removal. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 595:124706.
- Navneet Bung, Sowmya Ramaswamy Krishnan, and Arijit Roy. 2022. An in silico explainable multi-parameter optimization approach for de novo drug design against proteins from the central nervous system. *Journal of Chemical Information and Modeling*, 62(11):2685–2695.
- Denise B. Catacutan, Jeremie Alexander, Autumn Arnold, and Jonathan M. Stokes. 2024. Machine learning in preclinical drug discovery. *Nature Chemical Biology*, 20(8):960–973.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. 15(3).
- Yanchi Chen, Zihan Meng, Yong Li, Shibo Liu, Pei Hu, and En Luo. 2024. Advanced glycation end products and reactive oxygen species: uncovering the potential role of ferroptosis in diabetic complications. *Molecular Medicine*, 30(1):141.
- Ziqi Chen, Martin Renqiang Min, Srinivasan Parthasarathy, and Xia Ning. 2021. A deep generative model for molecule optimization via one fragment modification. *Nature machine intelligence*, 3(12):1040–1049.
- Fleur MG Cornelissen, Greta Markert, Ghislaine Deutsch, Maria Antonara, Noa Faaij, Imke Bartelink, David Noske, W Peter Vandertop, Andreas Bender, and Bart A Westerman. 2023. Explaining blood–brain barrier permeability of small molecules by integrated analysis of different transport mechanisms. *Journal of Medicinal Chemistry*, 66(11):7253–7267.
- DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.
- Oluwakemi Ebenezer, Maryam Amra Jordaan, Gea Carena, Tommaso Bono, Michael Shapi, and Jack A Tuszynski. 2022. An overview of the biological evaluation of selected nitrogen-containing heterocycle medicinal chemistry compounds. *International Journal of Molecular Sciences*, 23(15):8117.
- Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1).
- Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Hua-jun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*.
- Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M Glass, and Jimeng Sun. 2021. Mimoso: Multi-constraint molecule sampling for molecule optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 125–133.
- Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. 2022. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35:21342–21357.
- Gilles H Goetz, Marina Shalaeva, Giulia Caron, Giuseppe Ermondi, and Laurence Philippe. 2017. Relationship between passive permeability and molecular polarity using block relevance analysis. *Molecular Pharmaceutics*, 14(2):386–393.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.
- Corwin Hansch. 1969. Quantitative approach to biochemical structure-activity relationships. *Accounts of Chemical Research*, 2(8):232–239.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- JP Hughes, S Rees, SB Kalindjian, and KL Philpott. 2011. Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239–1249.

- Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. 2022. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022.
- Jan H Jensen. 2019. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572.
- Diego García Jiménez, Maura Vallaro, Luigi Vitagliano, Lucía López López, Giulia Apprato, Giuseppe Ermondi, and Giulia Caron. 2024. Molecular properties, including chameleonism, as essential tools for designing the next generation of oral beyond rule of five drugs. *ADMET and DMPK*, 12(5):721–736.
- Nagaraju Kerru, Lalitha Gummidi, Suresh Maddila, Kranthi Kumar Gangu, and Sreekantha B Jonnalagadda. 2020. A review on recent advances in nitrogen-containing molecules and their biological applications. *Molecules*, 25(8):1909.
- Yusra Sajid Kiani and Ishrat Jabeen. 2019. Lipophilic metabolic efficiency (lipmete) and drug efficiency indices to explore the metabolic properties of the substrates of selected cytochrome p450 isoforms. *ACS omega*, 5(1):179–188.
- Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. 2024. Genetic-guided gflownets: Advancing in practical molecular optimization benchmark. *CoRR*, abs/2402.05961.
- Khiem Le and Nitesh V Chawla. 2024. Utilizing large language models in an iterative paradigm with domain feedback for molecule optimization. *arXiv preprint arXiv:2410.13147*.
- Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Saeed Gopal Paliwal, Arash Vahdat, and Weili Nie. 2024. Molecule generation with fragment retrieval augmentation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Elena Lenci, Lorenzo Calugi, and Andrea Trabocchi. 2021. Occurrence of morpholine in central nervous system drug discovery. *ACS Chemical Neuroscience*, 12(3):378–390.
- Jiatong Li, Junxian Li, Yunqing Liu, Dongzhan Zhou, and Qing Li. 2025. Tomg-bench: Evaluating llms on text-based open molecule generation. *Preprint*, arXiv:2412.14642.
- Shengchao Liu, Jiong Xiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2024. Conversational drug editing using retrieval and domain feedback. In *The Twelfth International Conference on Learning Representations*.
- Sohvi Luukkonen, Helle W. van den Maagdenberg, Michael T.M. Emmerich, and Gerard J.P. van Westen. 2023. Artificial intelligence in multi-objective drug design. *Current Opinion in Structural Biology*, 79:102537.
- Asher Mullard. 2018. Re-assessing the rule of 5, two decades on. *Nature reviews. Drug discovery*, 17(11):777.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- Kunal Nepali, Hsueh-Yun Lee, and Jing-Ping Liou. 2018. Nitro-group-containing drugs. *Journal of medicinal chemistry*, 62(6):2851–2893.
- Christos A. Nicolaou and Nathan Brown. 2013. Multi-objective optimization methods in drug design. *Drug Discovery Today: Technologies*, 10(3):e427–e435.
- Orazio Nicolotti, Ilenia Giangreco, Antonellina Introcaso, Francesco Leonetti, Angela Stefanachi, and Angelo Carotti. 2011. Strategies of multi-objective optimization in drug discovery and development. *Expert Opinion on Drug Discovery*, 6(9):871–884.
- Yifan Niu, Ziqi Gao, Tingyang Xu, Yatao Bian, Yu Rong, and Jia Li. 2024. Trading-off multiple properties for molecular optimization.
- Saúl Noriega, Jaime Cardoso-Ortiz, Argelia López-Luna, Ma Del Refugio Cuevas-Flores, and Juan Armando Flores De La Torre. 2022. The diverse biological activity of recently synthesized nitro compounds. *Pharmaceuticals*, 15(6):717.
- OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.
- N Plattard, A Dupuis, V Migeot, S Haddad, and N Venisse. 2021. An overview of the literature on emerging pollutants: chlorinated derivatives of bisphenol a (clxbpa). *Environment International*, 153:106547.
- Thomas A Pollak, Svetlana Drndarski, James M Stone, Anthony S David, Philip McGuire, and N Joan Abbott. 2018. The blood–brain barrier in psychosis. *The Lancet Psychiatry*, 5(1):79–92.
- Lucca Caiaffa Santos Rosa, Caio Oliveira Argolo, Cayque Monteiro Castro Nascimento, and Andre Silva Pimentel. 2024. Identifying substructures that facilitate compounds to penetrate the blood–brain barrier via passive transport using machine learning explainer models. *ACS Chemical Neuroscience*.
- Philip Seeman. 2006. Targeting the dopamine d2receptor in schizophrenia. *Expert Opinion on Therapeutic Targets*, 10(4):515–531.
- Aylin Sertkaya, Trinidad Beleche, Amber Jessup, and Benjamin D. Sommers. 2024. Costs of drug development and research and development intensity in the us, 2000–2018. *JAMA Network Open*, 7(6):e2415445–e2415445.



- Teague Sterling and John J. Irwin. 2015. [Zinc 15 – ligand discovery for everyone](#). *Journal of Chemical Information and Modeling*, 55(11):2324–2337. PMID: 26479676.
- Duxin Sun, Wei Gao, Hongxiang Hu, and Simon Zhou. 2022a. [Why 90% of clinical drug development fails and how to improve it?](#) *Acta Pharmaceutica Sinica B*, 12(7):3049–3062.
- Mengying Sun, Jing Xing, Han Meng, Huijun Wang, Bin Chen, and Jiayu Zhou. 2022b. [Molsearch: Search-based multi-objective molecular generation and property optimization](#). KDD '22, page 4724–4732, New York, NY, USA. Association for Computing Machinery.
- Kyle Swanson, Parker Walther, Jeremy Leitz, Souhrid Mukherjee, Joseph C Wu, Rabindra V Shivnaraine, and James Zou. 2024. [Admet-ai: a machine learning admet platform for evaluation of large-scale chemical libraries](#). *Bioinformatics*, 40(7):btae416.
- Morgan Thomas, Noel M. O’Boyle, Andreas Bender, and Chris De Graaf. 2024. [Molscore: a scoring, evaluation and benchmarking framework for generative models in de novo drug design](#). *Journal of Cheminformatics*, 16(1).
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#).
- Aleksandra Twarda-Clapa, Aleksandra Olczak, Aneta M Białkowska, and Maria Koziołkiewicz. 2022. Advanced glycation end-products (ages): Formation, chemistry, classification, receptors, and diseases related to ages. *Cells*, 11(8):1312.
- Hicham Wahnou, Fouzia Hmimid, Ahmed Errami, Imane Nait Irahah, Youness Limami, and Mounia Oudghiri. 2024. [Integrating admet, enrichment analysis, and molecular docking approach to elucidate the mechanism of artemisia herba alba for the treatment of inflammatory bowel disease-associated arthritis](#). *Journal of Toxicology and Environmental Health, Part A*, 87(20):836–854.
- Haorui Wang, Marta Skreta, Cher Tian Ser, Wenhao Gao, Ling kai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, Yuanqi Du, Alan Aspuru-Guzik, Kirill Neklyudov, and Chao Zhang. 2025. [Efficient evolutionary search over chemical space with large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Yao Wei, Luca Palazzolo, Omar Ben Mariem, Davide Bianchi, Tommaso Laurenzi, Uliano Guerrini, and Ivano Eberini. 2024. [Investigation of in silico studies for cytochrome p450 isoforms specificity](#). *Computational and Structural Biotechnology Journal*, 23:3090–3103.
- David Weininger. 1988. [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.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhenxing Wu, Odin Zhang, Xiaorui Wang, Li Fu, Huifeng Zhao, Jike Wang, Hongyan Du, Dejun Jiang, Yafeng Deng, Dongsheng Cao, et al. 2024. [Leveraging language model for advanced multiproperty molecular optimization via prompt engineering](#). *Nature Machine Intelligence*, pages 1–11.
- Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. 2021. [{MARS}: Markov molecular sampling for multi-objective drug discovery](#). In *International Conference on Learning Representations*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. [WizardLM: Empowering large pre-trained language models to follow complex instructions](#). In *The Twelfth International Conference on Learning Representations*.
- Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. 2021. [Hit and lead discovery with explorative RL and fragment-based molecule generation](#). In *Advances in Neural Information Processing Systems*.
- Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. 2025. [Drugassist: A large language model for molecule optimization](#). *Briefings in Bioinformatics*, 26(1):bbae693.
- Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. 2018. [Graph convolutional policy network for goal-directed molecular graph generation](#). *Advances in neural information processing systems*, 31.
- Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. [LlaSMol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset](#). In *First Conference on Language Modeling*.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. 2024. [Chem-llm: A chemical large language model](#). *Preprint*, arXiv:2402.06852.

## A Details on MuMOInstruct

### A.1 Details on Evaluation Tasks

In this section, we provide descriptions of 10 tasks in MuMOInstruct used for evaluation.

#### A.1.1 IND tasks

Below are the 5 IND tasks:

1. BDP: This task optimizes molecules to improve BBBP, DRD2 receptor inhibition, and lipophilicity (plogP). These properties are critical for central nervous system (CNS) drugs, where molecules must penetrate the blood-brain barrier, bind effectively to the DRD2 receptor (a common target for neurological disorders), and maintain sufficient lipophilicity for stability and membrane permeability.
2. BDQ: This task optimizes molecules to increase BBBP, DRD2 binding affinity, and improve QED. By balancing brain permeability, receptor activity, and drug-likeness, this task captures realistic trade-offs required in CNS drug development.
3. BPQ: This task aims to improve BBBP, plogP, and QED, prioritizing brain permeability and appropriate lipophilicity while ensuring the optimized molecules retain favorable drug-like properties.
4. DPQ: This task focuses on improving DRD2, plogP, and QED. It targets receptor binding potency while optimizing lipophilicity and maintaining overall drug-likeness, representing key requirements for receptor-specific drug design.
5. BDPQ: This task jointly optimizes BBBP, DRD2 activity, plogP, and QED, reflecting a challenging and comprehensive scenario for developing CNS drug candidates with high permeability, receptor activity, and drug-like characteristics.

#### A.1.2 OOD tasks

Below are the 5 tasks used for evaluating out-of-domain generalizability:

1. MPQ: This task focuses on reducing mutagenicity, improving plogP, and enhancing drug-likeness (QED). This task represents an early-stage lead optimization scenario to reduce genotoxic risks while ensuring adequate lipophilicity and drug-like properties.

2. BDMQ: This task optimizes BBBP, DRD2 inhibition, mutagenicity, and QED. It reflects CNS drug development by balancing dopamine receptor activity, brain permeability, and safety while ensuring overall drug-likeness.
3. BHMQ: This task focuses on increasing BBBP and HIA, reducing mutagenicity, and improving QED. It is particularly relevant for orally administered CNS drugs, where both brain and intestinal absorption are critical.
4. BMPQ: This task optimizes BBBP, mutagenicity, plogP, and QED. It reflects CNS drug design by balancing adequate lipophilicity, reduced toxicity, and favorable drug-like properties, simulating realistic requirements for CNS-active drugs.
5. HMPQ: This task enhances HIA, reduces mutagenicity, and improves plogP and QED. It represents optimization for orally administered drugs, focusing on absorption, genotoxic risk reduction, and overall drug-like quality.

### A.2 Additional Filtering in Test Set

Out of the initial pool of 250K molecules sampled from ZINC, we select a molecule into the test set of a task which has a property worse than the median  $MPS_{\text{train}}$ . Additionally, for properties with highly skewed distributions, we exclude molecules falling below the 10th percentile of properties in training hit molecules, thereby eliminating extreme cases (e.g., a molecule with a plogP of -30) that are rarely encountered as hits. After applying these steps to the initial pool of 250K molecules, we randomly select at most 500 molecules into the test set for each task, with possible overlap across tasks. Table A1 presents the property characteristics of training pairs and test molecules in all 10 evaluation tasks. Table A2 presents the property characteristics of all 63 tasks in MuMOInstruct.

### A.3 Quality Control

We implement multiple quality control measures to ensure dataset integrity. In MuMOInstruct, molecules are represented as Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988) strings that are canonicalized and deduplicated. For each molecule, empirical property scores are computed using well-established tools: ADMET-AI (Swanson et al., 2024) for BBBP, HIA, Mutag and QED, and the official implementation provided by You et al. (2018) for DRD2 and plogP.

Table A1: Overview of Properties in MuMOInstruct Tasks for Evaluation

Task ID	MPS <sub>train</sub> ( $\Delta_{train}$ )						APS <sub>test</sub>					
	BBBP <sup>†</sup>	DRD2 <sup>†</sup>	HIA <sup>†</sup>	Mutag <sup>‡</sup>	plogP <sup>†</sup>	QED <sup>†</sup>	BBBP <sup>†</sup>	DRD2 <sup>†</sup>	HIA <sup>†</sup>	Mutag <sup>‡</sup>	plogP <sup>†</sup>	QED <sup>†</sup>
BDP	0.51 (0.32)	0.04 (0.45)	-	-	-0.23 (1.98)	-	0.34	0.01	-	-	-2.33	-
BDQ	0.55 (0.32)	0.04 (0.44)	-	-	-	0.35 (0.24)	0.37	0.01	-	-	-	0.21
BPQ	0.52 (0.36)	-	-	-	-1.51 (2.23)	0.70 (0.17)	0.31	-	-	-	-2.87	0.41
DPQ	-	0.06 (0.48)	-	-	-0.84 (2.67)	0.48 (0.21)	-	0.01	-	-	-3.32	0.36
BDPQ	0.51 (0.35)	0.04 (0.51)	-	-	-1.15 (2.53)	0.37 (0.25)	0.26	0.02	-	-	-4.92	0.24
MPQ	-	-	-	0.50 (-0.25)	-0.44 (1.86)	0.72 (0.17)	-	-	-	0.71	-1.61	0.52
BDMQ	0.54 (0.31)	0.04 (0.42)	-	0.45 (-0.20)	-	0.35 (0.23)	0.34	0.01	-	0.58	-	0.19
BHMQ	0.43 (0.37)	-	0.74 (0.28)	0.46 (-0.19)	-	0.70 (0.20)	0.18	-	0.34	0.54	-	0.23
BMPQ	0.49 (0.33)	-	-	0.47 (-0.22)	-0.76 (2.05)	0.69 (0.19)	0.33	-	-	0.65	-2.27	0.38
HMPQ	-	-	0.71 (0.29)	0.50 (-0.23)	-2.04 (2.30)	0.62 (0.19)	-	-	0.39	0.65	-3.23	0.30

MPS<sub>train</sub> and APS<sub>test</sub> denote the median and average property scores of the hit molecule  $M_x$  in the training and test set, respectively.  $\Delta_{train}$  denotes the average property difference across all training pairs.

While these property scores are not experimentally validated, they provide reliable and computationally efficient estimates, making them well-suited for a large-scale dataset construction like ours.

We further emphasize that the computational property predictors used in this work (e.g., ADMET-AI) are among the top-performing models in the TDC benchmark (Catacutan et al., 2024) and have been widely adopted in recent literature (Wei et al., 2024; Thomas et al., 2024; Wahnou et al., 2024). These predictors demonstrate strong alignment with experimentally measured values and serve as reliable proxies in practice, enabling scalable and consistent evaluation across large benchmarks. Moreover, it is worth noting that experimentally validated data are extremely limited for many pharmacological properties. For example, BBBP measurements are only available for about 2k drug-like molecules in existing public datasets – far fewer than what is required to train deep learning models or LLMs at scale. Meanwhile, correlating predicted improvements in generated molecules with experimental assays would strengthen real-world validation, but doing so would require wet-lab synthesis and testing at scale, which remains cost-prohibitive and logistically infeasible. As a result, most AI research in molecular optimization leverages predicted properties, making our approach consistent with prevailing practices.

We also ensure instruction diversity to enhance the generalizability of instruction-tuned LLMs (Xu et al., 2024). We provide a manually written, clear and concise seed instruction into GPT-4 (OpenAI, 2024) to construct multiple distinctly phrased (i.e., diverse) instructions. We select into MuMOInstruct 5 diverse instructions synonymous with the seed instruction. To evaluate LLMs’ instruction understanding and generaliz-

ability to unseen instructions, we hold out one instruction for each task during training. Thus, each task in MuMOInstruct has 5 diverse instructions for instruction tuning, and 1 unseen instruction for testing. All instructions are presented in Appendix A.4.

#### A.4 Diverse Instructions

Figure A1 presents the prompt template used for instruction-tuning.

The ‘{instruction}’ will be replaced with one of 6 diverse instructions. 5 diverse instructions are used in training, and 1 is held out for testing in the unseen instruction setting. Below are the six diverse instructions, where the first one is manually written, and the rest are generated by GPT-4o. The last one is the hold-out instruction.

1. “Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.”
2. “Modify the given molecule to adjust the specified molecular properties by substituting functional groups while keeping changes to the core structure minimal. Output only the SMILES of the modified molecule, wrapped in <SMILES> </SMILES> tags.”
3. “Your goal is to fine-tune the specified molecular properties of the given compound with minimal structural changes. Make the necessary adjustments and return the modified molecule in a SMILES format enclosed in <SMILES> </SMILES> tags.”
4. “Adjust the structure of the given molecule to target the specified adjustments in molecular



Table A2: Overview of all 63 tasks in MuMOInstruct

Task ID	#Pairs	#Mols	APS <sub>train</sub> ( $\Delta_{train}$ )						
			BBBP <sup>†</sup>	DRD2 <sup>†</sup>	HIA <sup>†</sup>	Mutag <sup>†</sup>	plogP <sup>†</sup>	QED <sup>†</sup>	
B	42,296	54,398	0.51 (0.32)	-	-	-	-	-	-
D	116,755	60,651	-	0.14 (0.40)	-	-	-	-	-
H	13,929	21,388	-	-	0.58 (0.34)	-	-	-	-
M	110,586	116,604	-	-	-	0.47 (-0.21)	-	-	-
P	105,949	149,774	-	-	-	-	-0.96 (1.88)	-	-
Q	157,863	167,041	-	-	-	-	-	0.61 (0.18)	-
BD	11,703	9,804	0.50 (0.31)	0.09 (0.41)	-	-	-	-	-
BH	6,676	10,657	0.45 (0.39)	-	0.53 (0.41)	-	-	-	-
BM	4,126	5,637	0.50 (0.32)	-	-	0.47 (-0.19)	-	-	-
BP	12,197	19,360	0.49 (0.35)	-	-	-	-2.01 (2.21)	-	-
BQ	24,577	33,303	0.52 (0.32)	-	-	-	-	0.55 (0.22)	-
DH	1,924	2,151	-	0.09 (0.43)	0.68 (0.27)	-	-	-	-
DM	22,834	19,359	-	0.12 (0.41)	-	0.50 (-0.21)	-	-	-
DP	19,154	16,796	-	0.12 (0.45)	-	-	-1.43 (1.90)	-	-
DQ	26,860	19,109	-	0.15 (0.43)	-	-	-	0.46 (0.21)	-
HM	1,247	1,786	-	-	0.67 (0.25)	0.49 (-0.22)	-	-	-
HP	10,348	18,052	-	-	0.53 (0.37)	-	-2.83 (2.33)	-	-
HQ	5,194	7,208	-	-	0.58 (0.35)	-	-	0.64 (0.18)	-
MP	27,554	43,224	-	-	-	0.48 (-0.22)	-0.50 (1.82)	-	-
MQ	33,192	39,643	-	-	-	0.47 (-0.22)	-	0.64 (0.17)	-
PQ	15,874	22,621	-	-	-	-	-2.18 (2.09)	0.67 (0.16)	-
BDH	829	997	0.47 (0.36)	0.08 (0.45)	0.65 (0.31)	-	-	-	-
BDM	1,423	1,595	0.48 (0.30)	0.07 (0.39)	-	0.51 (-0.20)	-	-	-
BDP	2,294	2,449	0.49 (0.32)	0.08 (0.45)	-	-	-2.95 (1.98)	-	-
BDQ	4,969	4,614	0.52 (0.32)	0.09 (0.44)	-	-	-	0.37 (0.24)	-
BHM	380	617	0.41 (0.37)	-	0.64 (0.31)	0.46 (-0.19)	-	-	-
BHP	5,259	9,141	0.45 (0.39)	-	0.48 (0.45)	-	-3.05 (2.61)	-	-
BHQ	3,100	4,619	0.43 (0.41)	-	0.54 (0.40)	-	-	0.64 (0.18)	-
BMP	1,125	1,841	0.47 (0.32)	-	-	0.49 (-0.21)	-1.07 (2.06)	-	-
BMQ	2,414	3,388	0.52 (0.32)	-	-	0.47 (-0.20)	-	0.60 (0.19)	-
BPQ	4,498	6,953	0.49 (0.36)	-	-	-	-2.92 (2.23)	0.64 (0.17)	-
DHM	304	301	-	0.07 (0.42)	0.72 (0.22)	0.60 (-0.27)	-	-	-
DHP	967	1,246	-	0.07 (0.45)	0.65 (0.28)	-	-3.15 (2.17)	-	-
DHQ	708	790	-	0.08 (0.47)	0.69 (0.27)	-	-	0.44 (0.27)	-
DMP	4,805	5,427	-	0.10 (0.45)	-	0.50 (-0.23)	-0.54 (1.86)	-	-
DMQ	6,398	6,220	-	0.14 (0.43)	-	0.50 (-0.23)	-	0.47 (0.21)	-
DPQ	2,349	2,589	-	0.11 (0.48)	-	-	-5.53 (2.67)	0.49 (0.21)	-
HMP	730	1,211	-	-	0.62 (0.28)	0.47 (-0.22)	-2.44 (2.22)	-	-
HMQ	534	749	-	-	0.69 (0.25)	0.52 (-0.22)	-	0.59 (0.19)	-
HPQ	3,142	5,289	-	-	0.49 (0.42)	-	-3.38 (2.39)	0.65 (0.17)	-
MPQ	3,481	5,384	-	-	-	0.51 (-0.25)	-1.44 (1.86)	0.65 (0.17)	-
BDHM	83	116	0.38 (0.32)	0.09 (0.43)	0.67 (0.28)	0.56 (-0.23)	-	-	-
BDHP	415	543	0.46 (0.35)	0.06 (0.46)	0.64 (0.32)	-	-5.63 (2.45)	-	-
BDHQ	410	493	0.45 (0.38)	0.07 (0.50)	0.67 (0.29)	-	-	0.39 (0.30)	-
BDMP	321	438	0.43 (0.31)	0.05 (0.40)	-	0.59 (-0.23)	-0.40 (1.75)	-	-
BDMQ	668	791	0.50 (0.31)	0.10 (0.42)	-	0.47 (-0.20)	-	0.37 (0.23)	-
BDPQ	694	802	0.49 (0.35)	0.10 (0.51)	-	-	-8.34 (2.53)	0.41 (0.25)	-
BHMP	243	421	0.42 (0.35)	-	0.60 (0.34)	0.46 (-0.20)	-3.05 (2.83)	-	-
BHMQ	213	333	0.42 (0.37)	-	0.67 (0.28)	0.49 (-0.19)	-	0.60 (0.20)	-
BHPQ	2,102	3,560	0.43 (0.41)	-	0.45 (0.47)	-	-3.66 (2.58)	0.64 (0.17)	-
BMPQ	415	690	0.48 (0.33)	-	-	0.50 (-0.22)	-1.48 (2.05)	0.61 (0.19)	-
DHMP	162	170	-	0.05 (0.45)	0.71 (0.22)	0.64 (-0.30)	-0.89 (1.77)	-	-
DHMQ	130	136	-	0.07 (0.43)	0.70 (0.23)	0.65 (-0.31)	-	0.43 (0.24)	-
DHPQ	302	363	-	0.06 (0.48)	0.64 (0.30)	-	-7.22 (2.45)	0.47 (0.26)	-
DMPQ	577	797	-	0.10 (0.46)	-	0.51 (-0.26)	-1.97 (2.09)	0.49 (0.22)	-
HMPQ	260	417	-	-	0.63 (0.29)	0.52 (-0.23)	-2.87 (2.30)	0.60 (0.19)	-
BDHMP	42	59	0.33 (0.30)	0.06 (0.45)	0.70 (0.25)	0.63 (-0.26)	-1.89 (2.05)	-	-
BDHMQ	50	72	0.41 (0.30)	0.09 (0.47)	0.68 (0.27)	0.55 (-0.24)	-	0.39 (0.28)	-
BDHPQ	186	232	0.41 (0.40)	0.06 (0.50)	0.66 (0.30)	-	-10.79 (2.74)	0.42 (0.30)	-
BDMPQ	73	119	0.43 (0.31)	0.10 (0.47)	-	0.52 (-0.25)	-1.06 (1.61)	0.43 (0.25)	-
BHMPQ	120	199	0.43 (0.35)	-	0.63 (0.31)	0.51 (-0.21)	-3.48 (2.93)	0.61 (0.20)	-
DHMPQ	61	64	-	0.06 (0.46)	0.67 (0.26)	0.70 (-0.35)	-1.50 (1.71)	0.46 (0.24)	-
BDHMPQ	23	33	0.36 (0.29)	0.07 (0.49)	0.68 (0.27)	0.64 (-0.29)	-2.88 (1.93)	0.45 (0.26)	-

#Pairs and #Mols denote the unique number of training pairs and molecules in each task. APS<sub>train</sub> and  $\Delta_{train}$  denote the average property score of  $M_x$  and average property change across all training pairs in each task.

- properties. Retain the core structure as much as possible. Respond with only the SMILES of the modified molecule enclosed in <SMILES> </SMILES> tags."
5. "Alter the given molecule to meet the desired property changes with the least structural alteration possible. Output only the adjusted molecule in SMILES format, using <SMILES> </SMILES> tags."
  6. "Modify the given molecular structure to target specific property changes, aiming to keep structural adjustments minimal. Respond solely with the SMILES notation for the adjusted molecule, enclosed within <SMILES> </SMILES> tags."

```
[INST]
{instruction}

%% Input : <SMILES> {source-smiles} </SMILES>
%% Adjust: {change_i} {property_i}, ..., {change_k} {property_k}
[/INST]

%% Response: {target-smiles}
```

Figure A1: Prompt template used for instruction-tuning GeLLM<sup>3</sup>Os

**Property Names:** We used the following names for each property where the former is used during instruction-tuning and the latter is used for evaluation in the unseen instruction setting. For other evaluation settings, we used the same property name as used in tuning.

1. BBBP: "BBB permeability", "Blood-brain barrier permeability (BBBP)"
2. DRD2: "DRD2 inhibition", "inhibition probability of Dopamine receptor D2"
3. HIA: "Intestinal adsorption", "human intestinal adsorption ability"
4. Mutag: "Mutagenicity", "probability to induce genetic alterations (mutagenicity)"
5. plogP: "Penalized octanol-water partition coefficient (penalized logP)", "Penalized logP which is logP penalized by synthetic accessibility score and number of large rings"
6. QED: "QED", "drug-likeness quantified by QED score"

## B Details on Experimental Setup

### B.1 GeLLM<sup>3</sup>Os

We develop a series of generalist GeLLM<sup>3</sup>Os which are trained on the power sets of 3, 4, and 6 properties, denoted as GeLLM<sup>3</sup>O-P(3), GeLLM<sup>3</sup>O-P(4), and GeLLM<sup>3</sup>O-P(6), respectively. To train these models, we fine-tune 2 general-purpose LLMs: Mistral-7B-Instruct-v0.3 (AI, 2023) and Llama3.1-8B-Instruct (Grattafiori et al., 2024) using LoRA (Hu et al., 2022), leveraging the Huggingface Transformers library (Wolf et al., 2020). We fine-tune all models with a learning rate of 1e-4 and a batch size of 128, using a cosine learning rate scheduler with a 5% warm-up period. We fine-tune task-specific GeLLM<sup>3</sup>Os and generalist GeLLM<sup>3</sup>Os for 10 and 3

epochs, respectively, to balance efficiency and overfitting. We set LoRA parameters with  $\alpha = 16$ , dropout of 0.05, and a rank of 16, and apply LoRA adapters to all projection layers and the language modeling head. We perform 0-shot evaluations (i.e., without in-context examples) for all GeLLM<sup>3</sup>O models in all tasks. For each test molecule, we generate 20 molecules using beam search decoding, with the number of beams set to 20.

The number of trainable parameters varies from 42 million for Mistral-7B-Instruct-v0.3 to 44 million for Llama3.1-8B-Instruct. Task-specific GeLLM<sup>3</sup>Os need up to 1 hour on average on a NVIDIA H100 (Hopper) GPU for 10 epochs. Generalist GeLLM<sup>3</sup>Os take from 8 to 24 hours on average on the same GPU for 3 epochs, depending on the number of tasks (property combinations). In total, we spent about 120 GPU hours on an NVIDIA H100 GPU with 96 GB HBM2e memory.

### B.2 Baselines

In this section, we present the baselines considered and selected for our comparison. Table A3 details the licenses and sources for both the datasets and models (i.e., artifacts). We ensured that all artifacts used in this work were employed in a manner consistent with their intended use as specified by the original authors or licensors. For the models we developed, we identified ethical considerations which are discussed in Section 10.

**General-purpose LLMs:** We evaluate 4 general-purpose LLMs: 2 open-weights LLMs, namely, Mistral-7B Instruct-v0.3 (AI, 2023), Llama-3.1 8B-Instruct (Touvron et al., 2023), and 2 closed-weights LLMs, namely, Claude-3.5 and GPT-4o to assess the performance of such LLMs in molecule optimization. For Mistral-7B Instruct-v0.3 and Llama-3.1 8B-Instruct, we use the officially released checkpoints provided in Huggingface. For Claude-3.5, we access the Sonnet checkpoints using the official API. We conduct both 0-shot and

5-shot inference (i.e., with 0 and 5 in-context examples, respectively) on all general-purpose LLMs using the same prompt templates (Appendix C.1). Due to the high cost of GPT-4o APIs, we adopt 1-shot prompting as a practical trade-off, as increasing to 5-shot yields minimal performance gains while significantly increasing inference cost.

**Foundational LLMs for Chemistry:** We use LLaSMo1<sub>Mistral</sub> (i.e., LLaSMo1 tuned over the base model Mistral-7B) and ChemLLM as two foundational LLMs for chemistry since they demonstrated state-of-the-art performance over others, such as MolInst (Fang et al., 2024), on a wide range of molecular tasks. We conduct only 0-shot inference since we did not observe any improvement with in-context examples in our preliminary experiments. We use a simpler prompt template (Appendix C.2) for inference since LLaSMo1 struggles to follow instructions in more detailed prompts.

**Task-specific non-LLM:** We use Prompt-Mo1Opt as our task-specific non-LLM baseline since it demonstrated better performance over other methods such as Modof on single- and double-property optimization tasks. During inference, we leverage its embedding generator and transformer modules which are separately trained for each task. We discuss the training details and limitations of Prompt-Mo1Opt in Appendix E.1 and E.2, respectively.

### B.3 Evaluation Metrics

We use the following evaluation metrics for a holistic comparison.

1. **Success rate (SR):** Success rate is the proportion of test molecules for which at least one of 20 generated molecules has improvements in all desired properties. If multiple generated molecules have improved properties, the one achieving the highest improvement across all properties is selected for evaluation. Higher SR demonstrates the models’ ability to successfully optimize most hit molecules.
2. **Validity (Val):** Validity is the proportion of test hit molecules for which at least one of 20 generated molecules is chemically valid. A molecule is considered valid if it can be successfully parsed by RDKit. Higher validity indicates more test cases have valid generations.
3. **Similarity (Sim):** Sim denotes the average Tanimoto similarity between successfully optimized molecules and the corresponding test molecules. The Tanimoto similarity is computed using binary Morgan fingerprints with a dimension of 2,048 and a radius of 2. Higher Sim indicates minimal structural modifications, which is desirable for retaining the core scaffold – a key requirement in lead optimization.
4. **Novelty (Nov):** Novelty is defined as the percentage of optimized molecules that are unseen during training. Higher Nov indicates the models’ ability to generate novel molecules, which is important for ensuring chemical diversity and finding new lead molecules.
5. **Synthetic Accessibility Score (SAS):** SAS estimates how easily a molecule can be synthesized based on its structural complexity and the presence of uncommon fragments. SAS generally ranges from 1 (easy to synthesize) to 10 (challenging to synthesize (Ertl and Schuffenhauer, 2009)).
6. **Relative Improvement (RI):** RI measures the average improvement in each property relative to its initial score in the input hit molecule. Higher RI implies significant improvements in desired properties on average.
7. **Average Property Score (APS):** APS represents the average property score of the generated optimized molecules in each property. Higher APS indicates that the model consistently generates molecules with better properties.

## C Prompt Templates

The following prompts are used to evaluate general-purpose LLMs and LLaSMo1.

### C.1 Prompt Template for General-purpose LLMs

We use a detailed prompt template which is clearly structured with a system prompt, task instruction, the input hit molecule, and task-specific optimization goals. Figure A2 shows the prompt template with an example task.

### C.2 Prompt Template for LLaSMo1

Unlike general-purpose LLMs, LLaSMo1 was instruction-tuned on multiple chemistry tasks with a specific prompt template. We observed that

Table A3: Licenses and Sources of Artifacts

Artifact	Source	License Type	Accessibility
Modof	<a href="https://github.com/ziqi92/Modof">https://github.com/ziqi92/Modof</a>	PolyForm Noncommercial License 1.0.0	Open Source
Prompt-MolOpt	<a href="https://github.com/wzxxxx/Prompt-MolOpt">https://github.com/wzxxxx/Prompt-MolOpt</a>	MIT License	Open Source
ChemLLM	<a href="https://huggingface.co/AI4Chem/ChemLLM-7B-Chat">https://huggingface.co/AI4Chem/ChemLLM-7B-Chat</a>	Apache license 2.0	Checkpoint
LlaSMol <sub>Mistral</sub>	<a href="https://huggingface.co/datasets/osunlp/SMolInstruct">https://huggingface.co/datasets/osunlp/SMolInstruct</a>	Creative Commons Attribution 4.0	Checkpoint
Claude 3.5 (Sonnet)	<a href="https://docs.anthropic.com/claude/reference/getting-started-with-the-api">https://docs.anthropic.com/claude/reference/getting-started-with-the-api</a>	Proprietary	API
GPT-4o	<a href="https://openai.com/api/">https://openai.com/api/</a>	Proprietary	API
Llama-3.1 8B-Instruct	<a href="https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct">https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct</a>	Llama 3.1	Checkpoint
Mistral-7B-Instruct-v0.3	<a href="https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3">https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3</a>	Apache license 2.0	Checkpoint

```

<<SYS>>
You are an expert medicinal chemist specializing in molecular optimization. You understand how structural modifications affect key ADMET properties and inhibitions of common receptor targets like DRD2.
<</SYS>>

[INST]
Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Use the examples (if provided) as a guide. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.

Examples:
%% Input : <SMILES> COCC(=O)Nc1ccc(C(N)=O)cc1 </SMILES>
%% Adjust: decrease Mutagenicity, increase Penalized octanol-water partition coefficient (penalized logP) and increase QED
%% Response: <SMILES> COCC(=O)Nc1ccc(Br)cc1 </SMILES>

Task:
%% Input : <SMILES> COC1COCCN(C(=O)c2ccno2)C1 </SMILES>
%% Adjust: decrease Mutagenicity, increase Penalized octanol-water partition coefficient (penalized logP) and increase QED
[/INST]

%% Response:

```

Figure A2: An example of a prompt used for general-purpose LLMs

LlaSMol struggled with following instructions using the prompt template for general-purpose LLMs, resulting in poor performance. Hence, we used a simpler prompt template without a system prompt and without a clear separation of task instruction, input, and response. Moreover, we only conduct 0-shot inference for LlaSMol. Figure A3 shows the prompt template with the same task used above but without in-context examples:

## D Complete Experimental Results

### D.1 IND Evaluation

Tables A4, A5, A6, A7 and A8 present the performance comparison of GeLLM<sup>3</sup>Os with baselines on

all 5 IND tasks.

### D.2 OOD Evaluation

Tables A9, A10, A11, A12 and A13 present the performance comparison of GeLLM<sup>3</sup>Os with baselines on all 5 OOD tasks.

Since OOD tasks represent novel property combinations excluded from the training data, task-specific models are not applicable in this setting. Additionally, several properties in these tasks are not used in training generalist models GeLLM<sup>3</sup>O-P(3) and GeLLM<sup>3</sup>O-P(4), making comparison with these models infeasible.



Modify the molecule <SMILES> COC1COCCN(C(=O)c2ccno2)C1 </SMILES> to decrease its Mutagenicity, increase its Penalized octanol-water partition coefficient (penalized logP) value, and increase its QED value. Keep the modifications to the molecule structure as minimal as possible.  
 %% Response:

Figure A3: An example of a prompt used for LlaSMol

Table A4: Overall Performance on BDP

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
							BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	plogP <sup>↑</sup>
<b>General-purpose LLMs</b>									
Mistral (0-shot)	6.60	75.00	<b>0.81</b>	<b>100.00</b>	3.72	0.68	0.47	0.03	-0.91
Llama (0-shot)	22.00	85.60	0.73	<b>100.00</b>	3.95	0.74	0.58	0.03	-1.94
Claude-3.5 (0-shot)	19.60	94.40	0.66	<b>100.00</b>	3.53	1.05	0.65	0.05	-1.49
GPT-4o (0-shot)	7.80	87.20	0.69	<b>100.00</b>	3.36	0.90	0.61	0.04	-1.10
Mistral (5-shot)	35.20	95.20	0.64	96.59	3.42	2.10	0.66	0.11	-0.87
Llama (5-shot)	35.40	96.80	0.57	79.10	3.50	2.71	0.64	0.17	-0.83
Claude-3.5 (5-shot)	35.40	95.40	0.50	<b>100.00</b>	3.18	2.43	0.77	0.10	-0.45
GPT-4o (1-shot)	9.40	88.60	0.69	<b>100.00</b>	3.59	0.79	0.62	0.04	-0.91
<b>Foundational LLMs for Chemistry</b>									
ChemLLM	0.20	12.80	0.17	<b>100.00</b>	<b>2.48</b>	1.20	<b>0.97</b>	0.03	0.81
LlaSMol <sub>Mistral</sub>	<b>43.60</b>	<b>98.40</b>	0.62	<b>99.54</b>	<b>3.38</b>	1.09	<b>0.59</b>	<b>0.05</b>	<b>-1.09</b>
<b>Task-specific non-LLMs</b>									
Prompt-MolOpt	12.20	20.80	0.12	96.72	2.66	<b>7.46</b>	0.96	<b>0.45</b>	<b>1.59</b>
<b>Task-specific LLMs</b>									
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	84.80	96.80	0.47	<b>100.00</b>	3.06	4.30	0.77	0.23	0.46
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	<b>86.80</b>	99.00	0.48	<b>99.31</b>	<b>3.01</b>	<b>4.38</b>	<b>0.76</b>	<b>0.25</b>	<b>0.42</b>
GeLLM <sup>3</sup> O-4 <sub>Mistral</sub>	71.60	97.40	0.49	95.25	2.92	3.27	0.74	0.18	0.17
GeLLM <sup>3</sup> O-4 <sub>Llama</sub>	53.60	98.80	0.63	99.25	3.19	1.94	0.68	0.09	-0.57
Impv-Task	99.1	0.6	-22.6	-0.2	10.9	301.8	28.8	400.0	138.5
<b>Generalist LLMs</b>									
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	75.60	98.20	0.56	<b>100.00</b>	3.18	3.31	0.68	0.16	0.07
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	77.40	99.00	0.51	99.74	3.10	3.16	0.74	0.16	0.04
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	81.40	98.80	0.55	99.75	3.07	3.95	0.73	0.19	0.12
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	80.40	<b>99.40</b>	0.54	99.75	3.01	3.60	0.73	0.18	0.02
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>83.00</b>	98.80	0.57	<b>99.76</b>	<b>3.17</b>	<b>3.60</b>	<b>0.72</b>	<b>0.17</b>	<b>0.12</b>
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	77.00	<b>99.40</b>	0.53	99.48	3.00	3.73	0.74	0.18	0.25
Impv-Gen	90.4	0.4	-8.1	0.2	6.2	230.3	22.0	240.0	111.0

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task. ‘-’ indicates cases where models are trained on 3 properties but the task has additional properties not included in the model training.

### D.3 Generalizability to Unseen Instructions

Tables A14, A15, A16, A17 and A18 present the performance comparison of GeLLM<sup>3</sup>O<sub>s</sub> with baselines on all 5 IND tasks when prompted with unseen instructions and unseen property names, than those used during instruction-tuning. This evaluation is meaningful as it mimics real-world scenarios where users may describe optimization tasks using varying terminologies, requiring models to understand the underlying semantics of the task rather than relying on exact token matching.

### D.4 Additional Case Studies

In this section, we provide two additional cases from IND task BDQ. As shown in Figure A4a, GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> improves molecular properties by removing a nitro group (-NO<sub>2</sub>) from the aromatic ring and replacing it with a chlorine atom (-Cl), while LlaSMol<sub>Mistral</sub> replaces the nitro group with two morpholine rings (highlighted structures). Removing the nitro group reduces polarity and eliminates a structural alert associated with toxicity and poor pharmacokinetics (Nepali et al., 2018), and the chlorine substitution enhances lipophilicity and promotes passive diffusion across the BBB (Plattard et al., 2021; Rosa et al., 2024).

Table A5: Overall Performance on BDQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
							BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>									
Mistral (0-shot)	3.00	86.80	<b>0.76</b>	<b>100.00</b>	3.91	0.53	0.56	0.02	0.23
Llama (0-shot)	2.20	92.00	0.64	<b>100.00</b>	4.23	0.53	0.49	0.02	0.22
Claude-3.5 (0-shot)	13.00	98.00	0.62	<b>100.00</b>	2.96	1.14	0.61	0.04	0.35
GPT-4o (0-shot)	2.00	91.20	0.69	<b>100.00</b>	2.81	0.62	0.63	0.03	0.33
Mistral (5-shot)	17.00	97.60	0.60	94.12	3.26	2.32	0.62	0.14	0.36
Llama (5-shot)	16.60	97.80	0.43	63.86	3.47	<b>5.70</b>	0.70	<b>0.26</b>	0.43
Claude-3.5 (5-shot)	29.40	94.20	0.43	<b>100.00</b>	<b>2.70</b>	3.80	0.79	0.17	0.51
GPT-4o (1-shot)	7.60	92.40	0.66	<b>100.00</b>	2.83	0.61	0.64	0.04	0.31
<b>Foundational LLMs for Chemistry</b>									
ChemLLM	1.00	21.20	0.55	<b>100.00</b>	4.12	0.82	0.56	0.02	0.31
LlaSMol <sub>Mistral</sub>	<b>31.40</b>	<b>98.80</b>	0.66	<b>100.00</b>	2.97	0.93	0.58	0.04	0.31
<b>Task-specific non-LLMs</b>									
Prompt-MolOpt	23.20	36.40	0.10	<b>100.00</b>	3.51	5.40	<b>0.95</b>	0.22	<b>0.73</b>
<b>Task-specific LLMs</b>									
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	87.00	98.80	0.47	<b>100.00</b>	2.90	5.61	0.79	0.22	0.49
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	<b>90.00</b>	99.40	0.46	99.78	2.93	5.66	0.81	0.23	0.51
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	57.40	99.20	0.55	99.65	3.05	2.56	0.70	0.11	0.43
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	48.60	99.60	0.59	<b>100.00</b>	3.36	1.29	0.64	0.06	0.37
Impv-Task	186.6	0.6	-30.3	-0.2	1.3	508.6	39.7	475.0	64.5
<b>Generalist LLMs</b>									
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	79.40	99.40	0.53	99.75	2.92	4.52	0.76	0.18	0.44
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	76.40	99.60	0.57	99.74	2.90	4.41	0.74	0.17	0.41
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	82.60	<b>99.80</b>	0.56	99.76	2.90	5.24	0.76	0.22	0.42
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	81.40	99.60	0.56	<b>100.00</b>	2.87	4.81	0.76	0.19	0.42
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>85.80</b>	99.60	0.59	<b>100.00</b>	2.94	4.78	0.75	0.19	0.40
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	79.60	<b>99.80</b>	0.56	<b>100.00</b>	2.87	5.05	0.76	0.20	0.42
Impv-Gen	173.2	0.8	-10.6	0.0	1.0	414.0	29.3	375.0	29.0

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task.

As a result, GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> achieves a notable increase in BBBP (+0.31), despite the hit molecule already having a relatively good BBBP value of 0.48, making further optimization more challenging. Additionally, the nitro group removal improves drug-likeness (Nepali et al., 2018; Noriega et al., 2022), and the minimal fragment substitution preserves the molecular scaffold, resulting in higher structural similarity (0.72 vs. 0.62). In contrast, LlaSMol<sub>Mistral</sub>'s morpholine rings increase polarity and hydrogen bonding capacity (Lenci et al., 2021), which shows limited BBBP gains (+0.08) (Figure A4b).

In another case shown in Figure A5a, GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> improves the properties by replacing the hydrophobic naphthalene group in the hit molecule with a nitrogen-containing heteroaromatic ring resembling pyrazine (highlighted structures). The naphthalene group increases hydrophobicity (Bouiahya et al., 2020) but may disrupt the balance between lipophilicity and

polarity required for efficient BBB penetration (Jiménez et al., 2024; Cornelissen et al., 2023), while also posing potential metabolic stability concerns (Kiani and Jabeen, 2019). Replacing it with a nitrogen-containing heterocyclic ring fine-tunes this lipophilicity-polarity balance, a well-established medicinal chemistry strategy to improve permeability and metabolic stability (Kerru et al., 2020). This modification enables GeLLM<sup>3</sup>O-P(6)<sub>Mistral</sub> to achieve notable improvements in BBBP (+0.41), DRD2 (+0.41), and QED (+0.11). Notably, the hit molecule's BBBP is 0.51, a decent but not maximal value, making further optimization a nontrivial task. In contrast, LlaSMol<sub>Mistral</sub>'s adjustment is a minor local change to a single bond near the core structure (Figure A5b), which neither meaningfully shifts polarity and lipophilicity nor reduces structural complexity, resulting in only negligible property improvements.

Table A6: Overall Performance on BPQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
							BBBP <sup>↑</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>									
Mistral (0-shot)	15.80	75.20	0.73	<b>100.00</b>	3.75	0.51	0.47	-1.75	0.47
Llama (0-shot)	28.40	89.40	0.64	<b>100.00</b>	3.91	0.72	0.59	-1.99	0.51
Claude-3.5 (0-shot)	56.00	93.00	0.62	<b>100.00</b>	3.68	0.86	0.58	-1.35	0.56
GPT-4o (0-shot)	36.40	89.00	0.73	<b>100.00</b>	3.62	0.42	0.50	-1.90	0.55
Mistral (5-shot)	68.60	89.80	0.63	98.54	3.44	0.79	0.66	-1.35	0.59
Llama (5-shot)	34.60	94.00	0.70	98.84	3.66	0.64	0.58	-1.64	0.53
Claude-3.5 (5-shot)	76.80	95.40	0.53	99.74	3.30	1.24	0.69	-0.45	0.64
GPT-4o (1-shot)	40.00	90.00	<b>0.75</b>	<b>100.00</b>	3.79	0.41	0.47	-1.95	0.50
<b>Foundational LLMs for Chemistry</b>									
ChemLLM	4.80	19.80	0.29	<b>100.00</b>	3.12	0.96	0.84	-1.04	0.62
LlaSMo1 <sub>Mistral</sub>	<b>86.00</b>	<b>96.00</b>	0.58	98.37	3.37	0.84	0.62	-1.17	0.62
<b>Task-specific non-LLMs</b>									
Prompt-MolOpt	15.80	16.60	0.10	98.73	<b>2.64</b>	<b>1.50</b>	<b>0.96</b>	<b>1.26</b>	<b>0.76</b>
<b>Task-specific LLMs</b>									
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	93.00	97.40	0.46	99.57	3.14	1.49	0.77	-0.16	0.69
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	94.00	98.20	0.50	98.94	3.13	1.38	0.77	-0.22	0.67
GeLLM <sup>3</sup> O-4 <sub>Mistral</sub>	90.20	96.40	0.46	97.78	2.97	1.41	0.77	0.01	0.69
GeLLM <sup>3</sup> O-4 <sub>Llama</sub>	93.40	97.40	0.59	99.36	3.28	1.12	0.71	-0.69	0.63
Impv-Task	9.3	2.3	-13.8	0.6	7.1	64.3	24.2	81.2	8.1
<b>Generalist LLMs</b>									
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	93.20	97.40	0.55	99.79	3.30	1.23	0.72	-0.51	0.65
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	95.40	<b>99.60</b>	0.50	99.58	3.14	1.46	0.77	-0.17	0.68
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	96.20	98.60	0.52	99.58	3.11	1.52	0.78	-0.18	0.69
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	93.80	99.20	0.47	98.72	3.03	1.64	0.82	-0.04	0.70
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>96.80</b>	<b>99.20</b>	0.53	99.38	3.23	<b>1.48</b>	0.78	-0.34	0.68
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	95.00	98.60	0.47	99.58	2.98	<b>1.66</b>	0.81	0.16	0.71
Impv-Gen	12.6	3.3	-8.6	1.0	4.2	76.2	25.8	70.9	9.7

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task.

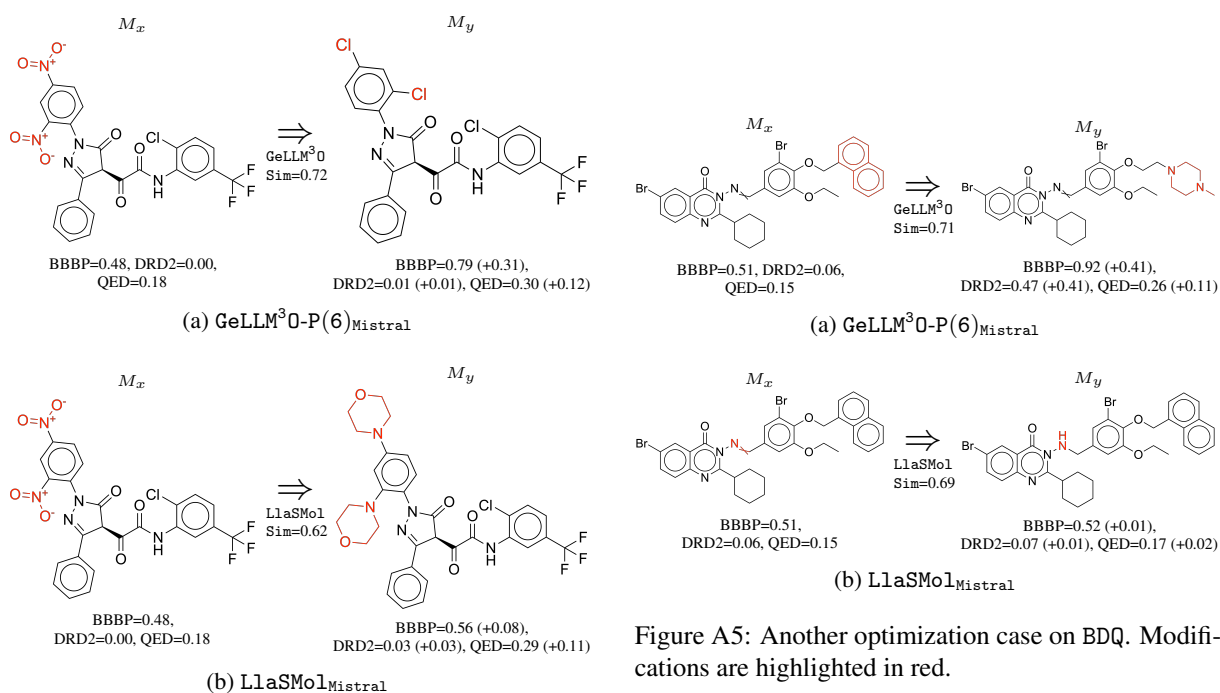


Figure A5: Another optimization case on BDQ. Modifications are highlighted in red.

Figure A4: An optimization case on BDQ. Modifications are highlighted in red.

Table A7: Overall Performance on DPQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
							DRD2 <sup>↑</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>									
Mistral (0-shot)	2.20	75.20	<b>0.65</b>	<b>100.00</b>	4.15	0.41	0.03	-2.79	0.44
Llama (0-shot)	2.60	87.60	0.62	<b>100.00</b>	4.01	0.32	0.03	-1.79	0.49
Claude-3.5 (0-shot)	11.00	95.80	0.54	<b>100.00</b>	3.70	0.51	0.04	-1.30	0.54
GPT-4o (0-shot)	2.80	86.00	0.57	<b>100.00</b>	3.55	0.50	0.03	-1.43	0.50
Mistral (5-shot)	10.40	92.60	0.54	92.31	3.49	1.10	0.13	-1.33	0.54
Llama (5-shot)	8.20	96.00	0.44	60.98	3.51	3.02	0.24	-0.83	0.59
Claude-3.5 (5-shot)	<u>29.20</u>	92.60	0.37	98.63	<b>3.02</b>	<b>2.87</b>	0.16	0.08	0.64
GPT-4o (1-shot)	7.00	86.60	0.62	<b>100.00</b>	3.81	0.44	0.03	-1.58	0.47
<b>Foundational LLMs for Chemistry</b>									
ChemLLM	0.60	18.20	0.28	<b>100.00</b>	3.31	0.42	0.03	-2.01	0.53
LlaSMol <sub>Mistral</sub>	24.00	<u>97.60</u>	0.57	<b>100.00</b>	3.53	0.61	0.04	-1.64	0.51
<b>Task-specific non-LLMs</b>									
Prompt-MolOpt	23.60	31.60	0.10	96.61	<b>2.58</b>	<b>5.46</b>	<b>0.46</b>	<b>1.08</b>	<b>0.80</b>
<b>Task-specific LLMs</b>									
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	62.80	98.20	0.37	97.45	3.04	3.87	0.24	-0.08	0.64
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	60.60	99.00	0.44	97.03	3.08	3.76	0.24	-0.12	0.59
GeLLM <sup>3</sup> O-4 <sub>Mistral</sub>	54.00	96.60	0.44	94.07	3.01	3.02	0.21	-0.54	0.58
GeLLM <sup>3</sup> O-4 <sub>Llama</sub>	39.60	98.40	0.57	98.99	3.36	1.32	0.08	-1.20	0.54
Impv-Task	115.1	6.0	0.0	-1.2	-0.7	34.8	50.0	-200.0	0.0
<b>Generalist LLMs</b>									
GeLLM <sup>3</sup> O-P(3) <sub>Mistral</sub>	57.20	98.20	0.50	99.65	3.26	2.22	0.13	-0.57	0.58
GeLLM <sup>3</sup> O-P(3) <sub>Llama</sub>	63.40	<b>99.80</b>	0.49	<b>100.00</b>	3.17	2.46	0.14	-0.45	0.59
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	<b>66.60</b>	99.20	0.53	99.40	3.26	<b>2.41</b>	0.13	-0.69	0.55
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	61.40	99.00	0.50	<b>100.00</b>	3.16	2.02	0.12	-0.40	0.58
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	60.80	99.40	0.54	99.67	3.31	2.16	0.12	-0.57	0.57
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	57.00	99.00	0.49	99.65	3.14	2.50	0.14	-0.36	0.58
Impv-Gen	128.1	7.1	43.2	0.8	-7.9	-16.0	-18.8	-962.5	-14.1

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task.

## E Discussion on Prompt-MolOpt

### E.1 Training Details

We use the official implementation of Prompt-MolOpt, and follow their two-stage training process. First, we train the multitask training process. First, we train the multitask graph attention network (MGA) for property predictions within each task (where a task refers to multi-property optimization). This MGA is later used to generate task-specific atom embeddings for optimization guidance. Specifically, we extract the unique molecules from each IND task’s training and validation sets for the MGA training. Secondly, we use task-specific molecule pairs, with atom embeddings provided generated by the trained MGA, to train the sequence-to-sequence transformer. The task-specific atom embeddings are aggregated with token embeddings through summation for transformer input. In total, Prompt-MolOpt has approximately 25 million parameters. Training for each task with early stopping typically takes 1.5 hours on a single NVIDIA V100 GPU with

16GB memory, hence totalling 7.5 GPU hours for 5 IND tasks. Training data statistics and best hyper-parameters are presented in Table A19.

### E.2 Limitations

First, in Prompt-MolOpt’s paper, the atom embedding produced by the MGA is claimed to be “property-specific”, implying that the embedding is trained independently for each property prediction (that is, only contains the specific property’s information). However, we find that during MGA training, this atom embedding is actually shared across all properties and is not differentiated for individual property prediction. This design introduces a coupling effect, where the atom embedding actually encodes information across multiple properties simultaneously. As a result, when these embeddings are later used to guide molecular optimization towards improving a particular property, the encoded information from other properties will bias the property-specific optimization. Therefore, we argue that these embeddings are more accu-



Table A8: Overall Performance on BDPQ

Model	SR <sup>†</sup>	Val <sup>†</sup>	Sim <sup>†</sup>	Nov <sup>†</sup>	SAS <sup>↓</sup>	RI <sup>†</sup>	APS			
							BBBP <sup>†</sup>	DRD2 <sup>†</sup>	plogP <sup>†</sup>	QED <sup>†</sup>
<b>General-purpose LLMs</b>										
Mistral (0-shot)	3.20	67.00	0.77	<b>100.00</b>	4.26	0.87	0.53	0.03	-1.85	0.32
Llama (0-shot)	5.20	83.40	<b>0.80</b>	<b>100.00</b>	4.52	0.62	0.46	0.02	-2.73	0.23
Claude-3.5 (0-shot)	8.00	<u>94.80</u>	0.60	<b>100.00</b>	3.77	1.34	0.49	0.06	-3.29	0.40
GPT-4o (0-shot)	1.80	87.00	0.71	<b>100.00</b>	4.08	0.39	0.41	0.02	-1.64	0.33
Mistral (5-shot)	11.00	79.00	0.69	98.18	3.71	0.96	0.57	0.06	-3.25	0.41
Llama (5-shot)	9.60	89.20	0.54	72.92	3.75	3.45	0.57	0.15	-2.04	0.40
Claude-3.5 (5-shot)	<u>20.80</u>	93.00	0.35	98.08	3.04	<b>3.53</b>	0.77	0.15	-0.58	0.61
GPT-4o (1-shot)	3.40	88.20	0.70	<b>100.00</b>	3.66	0.61	0.46	0.03	-0.84	0.39
<b>Foundational LLMs for Chemistry</b>										
ChemLLM	0.00	10.60	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
LlaSMol <sub>Mistral</sub>	14.00	90.20	0.62	98.57	3.48	1.03	0.50	0.06	-1.97	0.44
<b>Task-specific non-LLMs</b>										
Prompt-MolOpt	6.60	21.80	0.11	<b>100.00</b>	<b>2.70</b>	<b>5.36</b>	<b>0.92</b>	<b>0.39</b>	<b>0.51</b>	<b>0.77</b>
<b>Task-specific LLMs</b>										
GeLLM <sup>3</sup> O-4 <sub>Mistral</sub>	30.00	93.00	0.48	95.33	3.02	3.44	0.65	0.17	-1.55	0.53
GeLLM <sup>3</sup> O-4 <sub>Llama</sub>	28.00	94.00	0.66	98.57	3.57	1.02	0.56	0.05	-2.68	0.42
Impv-Task	44.2	0.0	37.1	-2.8	0.7	-2.5	-15.6	13.3	-167.2	-13.1
<b>Generalist LLMs</b>										
GeLLM <sup>3</sup> O-P(4) <sub>Mistral</sub>	<b>57.40</b>	<b>97.60</b>	0.52	99.65	3.29	3.04	0.65	0.15	-0.88	0.49
GeLLM <sup>3</sup> O-P(4) <sub>Llama</sub>	49.80	97.40	0.48	<b>100.00</b>	3.18	3.26	0.68	0.16	-0.69	0.52
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	54.00	97.40	0.54	99.26	3.34	3.09	0.65	0.16	-0.93	0.48
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	52.20	97.20	0.49	99.23	3.17	3.48	0.69	0.16	-0.65	0.53
Impv-Gen	176.0	4.9	48.6	1.6	-8.2	-13.9	-15.6	0.0	-51.7	-19.7

<sup>†</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Task and Impv-Gen denote the percentage improvement from the best task-specific LLM and best generalist LLM over the best baseline, respectively, where the best models are selected based on SR for each task. 'n/a' indicates undefined metrics due to no optimized cases (i.e., 0% SR).

Table A9: Overall Performance on MPQ

Model	SR <sup>†</sup>	Val <sup>†</sup>	Sim <sup>†</sup>	Nov <sup>†</sup>	SAS <sup>↓</sup>	RI <sup>†</sup>	APS		
							Mutag <sup>↓</sup>	plogP <sup>†</sup>	QED <sup>†</sup>
<b>General-purpose LLMs</b>									
Mistral (0-shot)	11.20	79.40	0.57	<b>100.00</b>	2.84	0.48	0.49	-0.33	0.61
Llama (0-shot)	25.80	89.20	0.44	99.22	2.89	0.61	<u>0.37</u>	-0.41	<u>0.68</u>
Claude-3.5 (0-shot)	17.40	95.00	0.49	<b>100.00</b>	3.22	0.52	0.47	-0.42	0.66
GPT-4o (0-shot)	19.40	85.40	<b>0.61</b>	<b>100.00</b>	3.25	0.35	0.53	-0.83	0.61
Mistral (5-shot)	59.60	98.40	0.54	98.66	3.07	0.57	0.39	-0.38	0.66
Llama (5-shot)	34.80	95.20	0.57	97.13	3.18	0.53	0.48	-0.41	0.65
Claude-3.5 (5-shot)	50.60	93.60	0.49	99.21	3.01	<u>0.71</u>	0.41	<u>0.13</u>	<u>0.68</u>
GPT-4o (1-shot)	21.40	86.20	0.60	<b>100.00</b>	3.17	0.48	0.47	-0.52	0.65
<b>Foundational LLMs for Chemistry</b>									
ChemLLM	6.20	31.60	0.31	<b>100.00</b>	<u>2.82</u>	0.61	0.40	-0.28	0.63
LlaSMol <sub>Mistral</sub>	<u>76.40</u>	<b>100.00</b>	0.55	99.74	3.07	0.53	0.42	-0.48	0.67
<b>Generalist LLMs</b>									
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>95.20</b>	99.80	0.53	99.79	2.97	0.85	0.37	0.46	<b>0.70</b>
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	93.60	<b>100.00</b>	0.48	99.79	<b>2.80</b>	<b>0.91</b>	<b>0.35</b>	<b>0.68</b>	<b>0.70</b>
Impv-Gen	24.6	-0.2	-3.6	0.1	3.3	60.4	-11.9	195.8	4.5

<sup>†</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Gen denotes the percentage improvement from the best generalist LLM over the best baseline, where the best models are selected based on SR for each task.

rately described as "task-specific", meaning they are only appropriate for guiding optimization under the same property combination (i.e., task) used during the transformer training.

Second, Prompt-MolOpt aggregates the task-specific atom embeddings and token embeddings through direct summation. However, this approach is problematic because the atom embeddings gener-

Table A10: Overall Performance on BDMQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS			
							BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	Mutag <sup>↓</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>										
Mistral (0-shot)	1.20	82.00	0.68	<u>100.00</u>	3.13	0.37	0.50	0.02	0.42	0.24
Llama (0-shot)	1.20	87.20	<b>0.76</b>	<u>100.00</u>	3.89	0.30	0.46	0.02	0.45	0.18
Claude-3.5 (0-shot)	15.00	97.00	0.57	<u>100.00</u>	2.84	0.87	0.58	0.06	0.32	0.34
GPT-4o (0-shot)	1.60	89.40	0.67	<u>100.00</u>	3.08	0.18	0.45	0.02	0.36	0.32
Mistral (5-shot)	20.40	94.00	0.59	94.12	2.98	1.65	0.61	0.11	0.30	0.37
Llama (5-shot)	16.80	95.80	0.39	54.76	3.26	<u>3.22</u>	<u>0.71</u>	<b>0.29</b>	<b>0.24</b>	0.43
Claude-3.5 (5-shot)	<u>30.40</u>	95.60	0.49	<u>100.00</u>	<b>2.71</b>	2.32	0.68	0.12	0.31	<b>0.45</b>
GPT-4o (1-shot)	6.20	90.80	0.70	<u>100.00</u>	2.86	0.75	0.51	0.04	0.34	0.29
<b>Foundational LLMs for Chemistry</b>										
ChemLLM	0.00	13.20	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
LlaSMol <sub>Mistral</sub>	28.20	<u>98.20</u>	0.66	<u>100.00</u>	2.89	0.52	0.51	0.03	0.37	0.31
<b>Generalist LLMs</b>										
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	<b>79.00</b>	99.00	0.56	<u>100.00</u>	2.84	3.10	<b>0.73</b>	0.16	0.33	0.42
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	74.20	<b>99.60</b>	0.55	<u>100.00</u>	2.74	<b>3.25</b>	<b>0.73</b>	0.16	0.33	<b>0.45</b>
Impv-Gen	159.9	3.6	14.3	0.0	-4.8	33.6	7.4	33.3	6.5	-6.7

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Gen denotes the percentage improvement from the best generalist LLM over the best baseline, where the best models are selected based on SR for each task. ‘n/a’ indicates undefined metrics due to no optimized cases (i.e., 0% SR).

Table A11: Overall Performance on BHMQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS			
							BBBP <sup>↑</sup>	HIA <sup>↑</sup>	Mutag <sup>↓</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>										
Mistral (0-shot)	12.71	76.27	0.73	<u>100.00</u>	3.56	1.90	0.32	0.64	0.37	0.24
Llama (0-shot)	11.02	92.37	0.74	<u>100.00</u>	4.39	0.68	0.28	0.63	0.45	0.20
Claude-3.5 (0-shot)	38.98	94.92	0.51	<u>100.00</u>	2.93	2.35	0.49	0.85	0.36	0.52
GPT-4o (0-shot)	17.80	94.92	0.66	<u>100.00</u>	3.70	0.65	0.32	0.67	0.37	0.43
Mistral (5-shot)	34.75	86.44	0.70	<u>100.00</u>	3.36	1.31	0.42	0.70	0.39	0.40
Llama (5-shot)	36.44	92.37	0.67	97.67	3.78	1.13	0.37	0.64	0.39	0.34
Claude-3.5 (5-shot)	52.54	95.76	0.48	<u>100.00</u>	<b>2.78</b>	<u>2.52</u>	0.50	<b>0.92</b>	0.37	<b>0.58</b>
GPT-4o (1-shot)	14.41	92.37	<b>0.75</b>	<u>100.00</u>	3.65	0.67	0.28	0.60	0.39	0.34
<b>Foundational LLMs for Chemistry</b>										
ChemLLM	1.69	12.71	0.53	<u>100.00</u>	3.10	0.74	<u>0.54</u>	0.54	<b>0.22</b>	0.32
LlaSMol <sub>Mistral</sub>	<u>53.39</u>	<u>96.61</u>	0.62	<u>100.00</u>	3.16	1.14	0.37	0.69	0.41	0.45
<b>Generalist LLMs</b>										
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	86.44	98.31	0.54	<u>100.00</u>	3.19	2.58	0.60	0.84	0.37	0.51
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	<b>93.22</b>	<b>100.00</b>	0.49	99.09	3.02	<b>3.57</b>	<b>0.64</b>	<b>0.92</b>	0.34	<b>0.58</b>
Impv-Gen	74.6	3.5	-21.0	-0.9	4.4	213.2	73.0	33.3	-17.1	28.9

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Gen denotes the percentage improvement from the best generalist LLM over the best baseline, where the best models are selected based on SR for each task.

ated by MGA and the token embeddings reside in different latent spaces. Combining representations from different spaces directly through summation is an ill-considered fusion strategy. A more principled approach would involve introducing a projection layer to align both embeddings into a common space before aggregation.

## F Discussions on DeepSeek-R1

DeepSeek-R1 (DeepSeek-AI, 2025) is a recently open-sourced, reasoning-focused LLM trained via

large-scale reinforcement learning without relying on large amounts of supervised fine-tuning data. Experiments demonstrated that DeepSeek-R1 has strong reasoning capabilities comparable to OpenAI-o1-1217 on tasks such as logical inference, mathematics, and coding. Their experiments also highlight the effectiveness of distillation, where smaller distilled models, such as Qwen2.5-14B, outperform the larger base model QwQ-32B-Preview by a significant margin. Given the relevance of reasoning capabilities in multi-property

Table A12: Overall Performance on BMPQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS			
							BBBP <sup>↑</sup>	Mutag <sup>↓</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>										
Mistral (0-shot)	12.57	79.06	0.61	<b>100.00</b>	3.18	0.54	0.60	0.47	-1.07	0.45
Llama (0-shot)	16.75	93.72	0.51	<b>100.00</b>	3.16	0.57	0.48	0.40	-1.32	0.46
Claude-3.5 (0-shot)	44.50	94.76	0.55	<b>100.00</b>	3.34	0.85	0.59	0.44	-0.55	0.51
GPT-4o (0-shot)	25.13	93.19	0.67	<b>100.00</b>	3.52	0.35	0.48	0.54	-1.66	0.50
Mistral (5-shot)	49.21	95.81	0.62	96.81	3.30	0.73	0.63	0.46	-0.93	0.55
Llama (5-shot)	31.94	96.34	0.66	96.72	3.40	0.60	0.60	0.48	-1.02	0.49
Claude-3.5 (5-shot)	52.36	92.15	0.46	<b>100.00</b>	2.97	1.08	0.69	0.37	<b>0.43</b>	0.61
GPT-4o (1-shot)	24.08	90.58	<b>0.68</b>	<b>100.00</b>	3.50	0.45	0.49	0.52	-1.41	0.47
<b>Foundational LLMs for Chemistry</b>										
ChemLLM	5.24	19.37	0.24	<b>100.00</b>	2.93	0.72	<b>0.80</b>	<b>0.32</b>	-0.83	0.57
LlaSMol <sub>Mistral</sub>	<u>64.92</u>	<u>98.95</u>	0.58	99.19	3.14	0.57	0.56	0.49	-0.91	0.57
<b>Generalist LLMs</b>										
GeLLM <sup>3</sup> 0-P(6) <sub>Mistral</sub>	91.10	<b>100.00</b>	0.53	99.43	3.04	1.06	0.74	0.40	-0.09	0.62
GeLLM <sup>3</sup> 0-P(6) <sub>Llama</sub>	<b>95.29</b>	98.95	0.49	99.45	<b>2.87</b>	<b>1.20</b>	0.76	0.37	0.29	<b>0.65</b>
Impv-Gen	46.8	0.0	-15.5	0.3	8.6	110.5	35.7	-24.5	131.9	14.0

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Gen denotes the percentage improvement from the best generalist LLM over the best baseline, where the best models are selected based on SR for each task.

Table A13: Overall Performance on HMPQ

Model	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS			
							HIA <sup>↑</sup>	Mutag <sup>↓</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
<b>General-purpose LLMs</b>										
Mistral (0-shot)	21.88	84.38	<b>0.72</b>	<b>100.00</b>	3.62	0.72	0.66	0.50	-1.41	0.36
Llama (0-shot)	15.62	91.67	0.47	<b>100.00</b>	3.14	0.60	0.78	<b>0.33</b>	-1.25	0.50
Claude-3.5 (0-shot)	38.54	96.88	0.54	<b>100.00</b>	3.42	1.01	0.75	0.42	-0.91	0.45
GPT-4o (0-shot)	20.83	92.71	0.67	<b>100.00</b>	3.64	0.37	0.70	0.50	-2.29	0.42
Mistral (5-shot)	46.88	89.58	0.66	97.78	3.68	0.91	0.73	0.49	-1.72	0.42
Llama (5-shot)	33.33	93.75	0.68	<b>100.00</b>	3.66	0.61	0.71	0.46	-1.73	0.39
Claude-3.5 (5-shot)	<u>65.62</u>	96.88	0.48	<b>100.00</b>	3.12	<u>1.32</u>	0.87	0.41	-0.43	0.56
GPT-4o (1-shot)	25.00	92.71	0.70	<b>100.00</b>	3.74	0.54	0.62	0.46	-2.00	0.40
<b>Foundational LLMs for Chemistry</b>										
ChemLLM	3.12	22.92	0.58	<b>100.00</b>	<b>3.02</b>	0.51	0.54	0.36	-1.90	0.54
LlaSMol <sub>Mistral</sub>	53.12	<u>98.96</u>	0.62	<b>100.00</b>	3.37	0.70	0.74	0.50	-1.89	0.48
<b>Generalist LLMs</b>										
GeLLM <sup>3</sup> 0-P(6) <sub>Mistral</sub>	91.67	<b>100.00</b>	0.55	<b>100.00</b>	3.34	1.42	0.91	0.41	-0.70	0.56
GeLLM <sup>3</sup> 0-P(6) <sub>Llama</sub>	<b>97.92</b>	<b>100.00</b>	0.46	98.94	3.06	<b>1.76</b>	<b>0.94</b>	0.39	<b>-0.30</b>	<b>0.64</b>
Impv-Gen	49.2	3.2	-4.2	-1.1	1.9	33.3	8.0	-4.9	30.2	14.3

<sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. For each task, the best baseline performance is underlined and the best overall performance is in **bold** for each metric. Impv-Gen denotes the percentage improvement from the best generalist LLM over the best baseline, where the best models are selected based on SR for each task.

molecule optimization, we chose to evaluate the distilled version of Llama-3.1-8B, DeepSeek-Distill-R1-Llama-8B, as it is the only version directly comparable to the Llama-based models in our experiments.

Following the recommendation of DeepSeek’s authors in their paper, we avoided using system prompts and few-shot prompting, as such settings have been shown to degrade the model’s performance. Additionally, using few-shot prompts would significantly increase the cost and resource requirements, as it resulted in considerably longer

response generation times in our preliminary demonstrations. Thus, we employed a zero-shot setting, balancing efficiency and adherence to best practices.

Initially, we experimented with the same instruction template used in our general-purpose LLM evaluations. However, DeepSeek-R1-Distill-Llama-8B consistently failed to optimize or generate modified molecules. Instead, it simply echoed the input molecule as its response, even after increasing the token limit. This behavior is likely because the prompt asked the model to only gener-

Table A14: Performance on Unseen Instructions for BDQ

Model	Instr	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
								BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	plogP <sup>↑</sup>
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	seen	84.80	96.80	0.47	100.00	3.06	4.30	0.77	0.23	0.46
	unseen	<b>89.60</b>	97.60	0.45	99.55	3.05	<b>5.11</b>	0.79	<b>0.28</b>	0.47
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	seen	86.80	99.00	<b>0.48</b>	99.31	3.01	4.38	0.76	0.25	0.42
	unseen	85.40	98.80	0.44	99.30	2.90	<b>4.69</b>	0.78	<b>0.28</b>	<b>0.64</b>
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	seen	<b>83.00</b>	98.80	0.57	99.76	3.17	<b>3.60</b>	0.72	<b>0.17</b>	<b>0.12</b>
	unseen	75.80	98.60	0.59	99.74	3.24	3.15	0.70	0.14	-0.12
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	seen	<b>77.00</b>	99.40	0.53	99.48	3.00	<b>3.73</b>	0.74	<b>0.18</b>	<b>0.25</b>
	unseen	64.60	99.00	0.53	99.69	2.99	3.06	0.74	0.14	0.06

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O<sub>s</sub> are evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

Table A15: Performance on Unseen Instructions for BDQ

Model	Instr	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
								BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	QED <sup>↑</sup>
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	seen	87.00	98.80	<b>0.47</b>	100.00	2.90	5.61	0.79	0.22	0.49
	unseen	87.40	99.00	0.44	100.00	2.83	<b>6.29</b>	0.81	<b>0.25</b>	0.50
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	seen	90.00	99.40	0.46	99.78	2.93	5.66	0.81	0.23	0.51
	unseen	90.40	99.80	0.46	99.56	2.83	5.68	0.81	0.23	0.51
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	seen	<b>85.80</b>	99.60	0.59	100.00	2.94	<b>4.78</b>	0.75	<b>0.19</b>	0.40
	unseen	80.40	99.40	0.59	99.75	2.93	4.54	0.74	0.17	0.39
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	seen	<b>79.60</b>	99.80	0.56	100.00	2.87	<b>5.05</b>	0.76	<b>0.20</b>	0.42
	unseen	73.40	99.80	0.57	100.00	2.85	4.56	0.75	0.19	0.41

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O<sub>s</sub> are evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

Table A16: Performance on Unseen Instructions for BPQ

Model	Instr	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
								BBBP <sup>↑</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	seen	93.00	97.40	0.46	99.57	3.14	1.49	0.77	<b>-0.16</b>	0.69
	unseen	93.00	97.80	0.45	98.71	3.13	1.48	0.78	-0.13	0.69
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	seen	94.00	98.20	0.50	98.94	3.13	1.38	0.77	<b>-0.22</b>	0.67
	unseen	93.80	98.60	0.49	98.72	3.07	1.42	0.77	-0.11	0.68
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	seen	96.80	99.20	0.53	99.38	3.23	1.48	0.78	-0.34	0.68
	unseen	96.20	98.80	0.54	98.96	3.22	1.42	0.77	<b>-0.46</b>	0.67
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	seen	95.00	98.60	0.47	99.58	2.98	1.66	0.81	<b>0.16</b>	0.71
	unseen	95.60	98.40	0.47	99.58	2.98	1.66	0.81	0.10	0.71

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O<sub>s</sub> are evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

ate SMILES strings without explicitly mentioning step-by-step reasoning (i.e., chain-of-thought). Figure A6 provides an example of such a failure case.

We then modified the prompt structure, removing the clear separation between task instructions, input, and output, and explicitly instructing the model to “think step by step”. As shown in Figure A7, this adjustment enabled DeepSeek-R1-

Distill-Llama-8B to generate longer reasoning processes. Interestingly, the model demonstrated self-reflection and explored multiple reasoning trajectories, akin to an “aha moment” expressed in Table 3 in the original paper (DeepSeek-AI, 2025). Despite this, the final molecule was still not optimized for the desired properties, and the generation required significantly more tokens – up to 20K tokens more



Table A17: Performance on Unseen Instructions for DPQ

Model	Instr	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS		
								DRD2 <sup>↑</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	seen	62.80	98.20	<b>0.37</b>	97.45	3.04	3.87	0.24	<b>-0.08</b>	0.64
	unseen	64.20	98.40	0.35	98.44	2.90	3.95	0.25	0.39	0.65
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	seen	60.60	99.00	<b>0.44</b>	97.03	3.08	3.76	0.24	<b>-0.12</b>	0.59
	unseen	63.60	98.60	0.39	95.91	2.94	<b>4.36</b>	<b>0.28</b>	0.10	<b>0.62</b>
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	seen	<b>60.80</b>	99.40	0.54	99.67	3.31	<b>2.16</b>	<b>0.12</b>	-0.57	0.57
	unseen	54.60	98.80	0.55	99.63	3.32	1.99	0.11	<b>-0.82</b>	0.55
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	seen	<b>57.00</b>	99.00	0.49	99.65	3.14	<b>2.50</b>	<b>0.14</b>	-0.36	0.58
	unseen	53.60	99.60	0.50	100.00	3.15	2.15	0.12	<b>-0.39</b>	0.58

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O<sub>s</sub> are evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

Table A18: Performance on Unseen Instructions for BDPQ

Model	Instr	SR <sup>↑</sup>	Val <sup>↑</sup>	Sim <sup>↑</sup>	Nov <sup>↑</sup>	SAS <sup>↓</sup>	RI <sup>↑</sup>	APS			
								BBBP <sup>↑</sup>	DRD2 <sup>↑</sup>	plogP <sup>↑</sup>	QED <sup>↑</sup>
GeLLM <sup>3</sup> O-3 <sub>Mistral</sub>	seen	30.00	93.00	<b>0.48</b>	95.33	3.02	3.44	0.65	0.17	<b>-1.55</b>	0.53
	unseen	<b>32.80</b>	90.60	0.45	93.29	2.98	<b>3.62</b>	0.63	<b>0.19</b>	-1.24	0.52
GeLLM <sup>3</sup> O-3 <sub>Llama</sub>	seen	<b>28.00</b>	94.00	0.66	98.57	3.57	1.02	0.56	0.05	-2.68	0.42
	unseen	24.20	93.40	0.64	97.52	3.47	<b>1.29</b>	0.58	<b>0.06</b>	-2.67	0.44
GeLLM <sup>3</sup> O-P(6) <sub>Mistral</sub>	seen	<b>54.00</b>	97.40	0.54	99.26	3.34	<b>3.09</b>	0.65	<b>0.16</b>	-0.93	0.48
	unseen	49.80	97.20	<b>0.57</b>	99.20	3.37	2.81	0.63	0.14	-0.96	0.48
GeLLM <sup>3</sup> O-P(6) <sub>Llama</sub>	seen	<b>52.20</b>	97.20	0.49	99.23	3.17	3.48	0.69	0.16	-0.65	0.53
	unseen	46.40	97.20	0.48	99.14	3.09	3.52	0.68	0.16	-0.68	0.55

“seen” and “unseen” indicate whether the GeLLM<sup>3</sup>O<sub>s</sub> are evaluated with seen and unseen instructions, respectively. <sup>↑</sup> and <sup>↓</sup> denote whether a higher or lower value of the metric is desirable, respectively. The best-performing GeLLM<sup>3</sup>O in each row block is in **bold** if the performance difference between the models evaluated with seen and unseen instructions exceeds 5%.

Table A19: Training Details of Prompt-MolOpt

Task	MGA Training		Transformer Training	
	Data (# of uni mols)	Optimal parameters	Data (# of mol pairs)	Optimal parameters
BDQ	3,691	Epoch: 64	4,472	Epoch: 73
BPQ	5,562	Epoch: 81	4,048	Epoch: 113
BDP	1,959	Epoch: 115	2,064	Epoch: 116
DPQ	2,071	Epoch: 76	2,114	Epoch: 99
BDPQ	641	Epoch: 166	624	Epoch: 249

than those used in our instruction-tuned LLMs.

This token inefficiency makes large-scale evaluation of DeepSeek-R1 using such prompts infeasible. Due to these limitations, we decided not to pursue further experiments, such as instruction-tuning on top of DeepSeek-R1-Distill-Llama-8B or other distilled models. While the reasoning capabilities of DeepSeek-R1 are impressive, its current limitations in token efficiency, prompt sensitivity, and molecular optimization restrict its applicability in this domain. We plan to explore this further in future work.

```

[INST]
Your task is to modify the given molecule to adjust specific molecular properties
while keeping structural changes as minimal as possible. Use the examples (if
provided) as a guide. Your response should only contain a valid SMILES
representation of the modified molecule enclosed with <SMILES> </SMILES> tag.

Task:
%% Input : <SMILES> CC(=O)OCC1=C(C(=O)O)N2C(=O)[C@@H](NC(=O)CCC[C@H](N)C(=O)O)[
C@H]2SC1 </SMILES>
%% Adjust: increase BBB permeability, increase DRD2 inhibition, increase
Penalized octanol-water partition coefficient (penalized logP) and increase QED

[/INST]
%% Response:
<SMILES>CC(=O)OCC1=C(C(=O)O)N2C(=O)[C@@H](NC(=O)CCC[C@H](N)C(=O)O)[C@H]2SC1</
SMILES>
</br>
</br>
</br>
...

```

Figure A6: A failure case of DeepSeek-R1-Distill-Llama-8B on task BDPQ

```

Modify the molecule <SMILES> CC(=O)OCC1=C(C(=O)O)N2C(=O)[C@@H](NC(=O)CCC[C@H](N)C
(=O)O)[C@H]2SC1 </SMILES> to increase its BBB permeability value, and increase its
QED value. Keep the modifications to the molecule structure as minimal as
possible. Think step by step and then output the modified molecule.

%% Response:
Okay, so I need to modify the given molecule to increase both its BBB permeability
and QED value, while keeping the changes minimal. Let me start by understanding
what BBB and QED mean.
...
I notice it's a cyclic structure with some substituents. Let me try to visualize
it. It seems to have a central ring, maybe a thiazole or similar, with various
substituents including ester groups, amides, and a sulfur atom.
...
Another idea is to reduce the number of amide
bonds. Amides are hydrophilic, but they can make the molecule bulkier and more
polar, which might not always be good for BBB permeability.

Wait, the chain is NC(=O)CCC[C@H](N)C(=O)O. That's an amide group connected...
...
One approach could be to remove some of the ester or amide groups
...
Alternatively, maybe removing one of the carbonyl groups.
...
Wait, but replacing a carbonyl with a CH2 would change the structure significantly. I need to make
sure that the molecule still retains its functional groups necessary for its activity.
...
Another thought: the molecule has multiple ester and amide groups,
...
So, the final modified molecule would have the NC(=O) chain shortened from four
carbons to three carbons.
</think>

The molecule has been modified by shortening the NC(=O)CCC[C@H](N)C(=O)O chain to
NC(=O)CC[C@H](N)C(=O)O, reducing the carbon count from four to three. This change
decreases the molecular weight and increases hydrophilicity, enhancing both BBB
permeability and QED.

SMILES: CC(=O)OCC1=C(C(=O)O)N2C(=O)[C@@H](NC(=O)CC[C@H](N)C(=O)O)[C@H]2SC1

```

Figure A7: "aha moment" even in the distilled models, DeepSeek-R1-Distill-Llama-8B, where it does **self-reflection** on its own and even **explores multiple reasoning** trajectories or chain-of-thoughts. Still fails to optimize on the same case as in Figure A6.