Large Language Models for Controllable Multi-property Multi-objective Molecule Optimization

Vishal Dey[1], Xiao Hu[1], Xia Ning[1,2,3,4]
[1] Department of Computer Science and Engineering, The Ohio State University, USA
[2] Translational Data Analytics Institute, The Ohio State University, USA
[3]Department of Biomedical Informatics, The Ohio State University, USA
[4] College of Pharmacy, The Ohio State University, USA
Correspondence: ning.104@osu.edu

Abstract

In real-world drug design, molecule optimization requires selectively improving multiple molecular properties up to pharmaceutically relevant levels, while maintaining others that already meet such criteria. However, existing computational approaches and instruction-tuned LLMs fail to capture such nuanced property-specific objectives, limiting their practical applicability. To address this, we introduce `C-MuMOInstruct`, the first instruction-tuning dataset focused on multi-property optimization with explicit, property-specific objectives. Leveraging `C-MuMOInstruct`, we develop `GeLLM`^{4}`O-Cs`, a series of instruction-tuned LLMs that can perform targeted property-specific optimization. Our experiments across 5 in-distribution and 5 out-of-distribution tasks show that `GeLLM`^{4}`O-Cs` consistently outperform strong baselines, achieving up to 126% higher success rate. Notably, `GeLLM`^{4}`O-Cs` exhibit impressive 0-shot generalization to novel optimization tasks and unseen instructions. This offers a step toward a foundational LLM to support realistic, diverse optimizations with property-specific objectives. `C-MuMOInstruct` and code are accessible through `https://github.com/ninglab/GeLLMO-C`.

1 Introduction

Developing a new drug is a time-consuming and expensive process, requiring over a decade and $2 billions (Sertkaya et al., 2024). A key stage in this process is lead optimization (Nicolaou and Brown, 2013), where "hit" molecules – exhibiting promising early-stage bioactivity against drug targets – are optimized for multiple molecular properties (Nicolotti et al., 2011) critical for pharmaceutical success. In practice, this stage often requires improving specific properties up to a pharmaceutically significant level, while maintaining already desirable ones within acceptable bounds. We refer to this setting as controllable multi-property, multi-objective optimization (C-MuMO), allowing for property-specific objectives, and thus greater control over the optimization.

Such controllable optimization requires navigating complex trade-offs among multiple properties that are often competing or even conflicting (Niu et al., 2024). For instance, optimizing an oral antipsychotic drug requires sufficiently high blood-brain barrier permeability (BBBP) (Pollak et al., 2018) and dopamine receptor D2 (DRD2) inhibition (Seeman, 2001) to access the central nervous system (CNS) and block dopamine receptors in the CNS (Seeman et al., 1976). Meanwhile, properties related to toxicity, such as Potassium (K^+) channel inhibition must be lowered, since excessive inhibition of K^+ channels in the brain (Shepard et al., 2007) can cause fatal cardiac arrythmias (Sanguinetti and Tristani-Firouzi, 2006). Additionally, properties supporting oral bioavailability, such as intestinal absorption, must be maintained if they already meet desirable levels. These trade-offs highlight the need for property-specific objectives to mimic realistic optimization tasks.

Most existing computational approaches (Gao et al., 2022; Jensen, 2019; You et al., 2018; Blaschke et al., 2020) cannot handle tasks with multiple objectives. Furthermore, existing approaches for multi-objective optimization (Sun et al., 2022; Kim et al., 2024; Wu et al., 2024) rely on manually curated reward functions and careful task-specific tuning – limiting their scalability and applicability to diverse tasks in practice. We refer readers to Appendix A for a detailed review of existing approaches. Recently, instruction-tuned LLMs (Dey et al., 2025), demonstrated strong performance on diverse multi-property optimization tasks. However, they only tackle tasks where all properties should be improved simultaneously. This setting fails to capture the nuanced property-specific objectives prevalent in realistic lead optimization.

To address these critical limitations, we in-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 20996–21023
November 4-9, 2025 ©2025 Association for Computational Linguistics

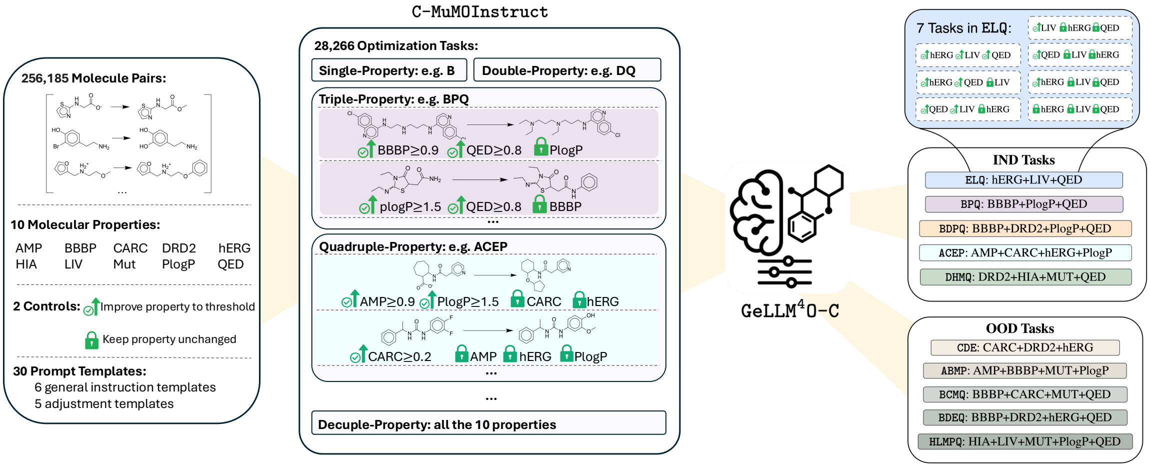


Figure 1: Overview of C-MuMOInstruct and GeLLM^{4}O-C. (Left) Illustration of property-specific optimization tasks. (Right) Training and evaluation workflow.

Table 1: Comparison among instruction-tuning datasets

Comparison	MolOpt-Instructions (Ye et al., 2025)	MuMOInstruct (Dey et al., 2025)	C-MuMOInstruct (ours)
Multi-objective	✗	✗	✓
Threshold-based	✓	✗	✓
Realistic	✗	✓	✓
#properties	5	6	10
#molecules	1,595,839	331,586	433,166
#pairs	1,029,949	255,174	256,185
#Total tasks	8	63	28,266
#Tasks $\geq$ 3 prop	0	42	27,401
#Eval $\geq$ 3 prop	0	10	119
#IND	8	5	51
#OOD	0	5	68

troduce C-MuMOInstruct, the first high-quality instruction-tuning dataset designed for C-MuMO tasks involving up to 10 molecular properties. Unlike prior datasets that require all properties to improve, C-MuMOInstruct explicitly incorporates controllable property-specific objectives – specifying which properties must be improved up to a user-defined property-specific threshold, and which must be maintained within acceptable bounds. This design better reflects real-world lead optimization, where some properties reach pharmaceutically significant levels in early stages, while others require multiple iterations for further improvement.

Built on C-MuMOInstruct, we introduce a family of Generalizable Large Language Models for Multi-property, Multi-Objective Controllable optimization, GeLLM^{4}O-C, by instruction-tuning general-purpose LLMs. GeLLM^{4}O-C is trained to handle tasks requiring selective improvement of specific properties while maintaining already desirable ones. We develop both specialist and generalist variants. Each specialist GeLLM^{4}O-C is trained on a single property combination with multiple controllable multi-objective tasks. Generalist GeLLM^{4}O-C is trained across diverse multi-property combinations and multiple controllable objectives within each combination, enabling cross-task knowledge transfer. This enables a single foundational model to handle novel and diverse C-MuMO tasks without task-specific fine-tuning.

We evaluate our GeLLM^{4}O-C models with strong general-purpose LLMs and foundational LLMs for chemistry across 5 in-distribution (IND) and 5 out-of-distribution (OOD) tasks. Our results reveal several key findings: **(1)** All GeLLM^{4}O-Cs substantially outperform state-of-the-art baselines on all IND and OOD tasks, with gains of up to 126% over the best baselines. **(2)** Generalist GeLLM^{4}O-Cs outperform specialist ones on 4 out of 5 IND tasks, with impressive gains of up to 26% on challenging tasks. **(3)** Generalist GeLLM^{4}O-Cs demonstrate remarkable 0-shot generalization to OOD tasks, outperforming strong baselines by 27% on average.

To the best of our knowledge, C-MuMOInstruct is the first large scale, high-quality instruction-tuning dataset specifically focused on controllable, multi-objective optimization with up to 10 properties. Generalist GeLLM^{4}O-Cs tuned on C-MuMOInstruct demonstrate strong generalization abilities, which highlights their strong potential to tackle unseen, diverse C-MuMO tasks prevalent in realistic drug design scenarios. Figure 1 presents the overall framework of GeLLM^{4}O-C. Dataset and code are accessible through https://github.com/ninglab/GeLLMO-C.

2 C-MuMOInstruct

In this paper, we introduce C-MuMOInstruct, which provides control over each property objective in multi-property optimization tasks, unlike existing datasets such as MuMOInstruct. This enables models tuned on C-MuMOInstruct to improve specific properties up to a user-defined level, while maintaining others at already desirable levels – a crucial capability that distinguishes `C-MuMOInstruct` from existing datasets. These key differences are highlighted in Table 1.

Problem Definition: A C-MuMO task is to modify a hit molecule M_x into an improved lead molecule M_y, via structural modifications on M_x, guided by property-specific objectives – controlling which properties to be improved and the extent of such improvement. Given $\mathcal{P}$ molecular properties, we define a pharmaceutically relevant level, Θ_p, for each property $p \in \mathcal{P}$, Accordingly, p is considered near-optimal if its score in M_x – denoted as $p(M_x)$ – is more desirable than Θ_p (represented as $p(M_x) \prec \Theta_p$), and sub-optimal, otherwise (represented as $p(M_x) \succeq \Theta_p$). The desirability of each property is determined by the intended pharmaceutical goal, where either higher or lower property scores increase the molecule's likelihood to be a successful drug candidate. For example, a higher BBBP is desired for drugs targeting the CNS to ensure their access to the brain, whereas a lower BBBP is desired for peripheral targets to prevent damage to the CNS.

Formally, a C-MuMO task optimizing M_x to M_y aims to improve all sub-optimal properties $\mathcal{P}_{\mathtt{i}} = \{p \in \mathcal{P} | p(M_x) \prec \Theta_p\}$ while maintaining all near-optimal properties $\mathcal{P}_{\mathtt{s}} = \{p \in \mathcal{P} \mid p(M_x) \succeq \Theta_p\}$ such that: **(1)** M_y remains structurally similar to M_x (similarity constraint); **(2)** M_y improves upon M_x in each sub-optimal property $p \in \mathcal{P}_{\mathtt{i}}$ by at least a property-specific threshold, Δ_p, represented as $(M_x \prec_{\Delta_p} M_y)_{\forall p \in \mathcal{P}_{\mathtt{i}}}$ (property improvement constraint); and **(3)** the absolute change from M_x to M_y in each near-optimal property $p \in \mathcal{P}_{\mathtt{s}}$ remains within Δ_p to ensure such properties with already desirable scores are maintained, represented as $(M_x \cong_{\Delta_p} M_y)_{\forall p \in \mathcal{P}_{\mathtt{s}}}$ (property stability constraint).

2.1 Design Principles

Following the above definition, we construct `C-MuMOInstruct`, the first high-quality instruction tuning dataset for C-MuMO tasks with property-specific objectives. Our design of `C-MuMOInstruct` is based on 5 key principles:

(1) Real-world relevance: C-MuMO tasks are widely prevalent in real-world lead optimization, where some properties may already meet desirable levels while others require further improvement. Each optimization task in `C-MuMOInstruct` is carefully curated to reflect nuanced multi-property objectives encountered in real-world drug design. By combining ADMET properties (e.g., intestinal absorption, mutagenicity) with properties related to specific therapeutic endpoints (e.g., dopamine receptor and potassium channel inhibition), `C-MuMOInstruct` captures complex and realistic multi-property trade-offs.

(2) Controllable multi-property threshold-based optimization: Unlike prior datasets such as `MuMOInstruct`, which enforces the same objective for all properties (i.e., 'improve all' simultaneously), `C-MuMOInstruct` introduces property-specific objectives – specifying sub-optimal properties to improve and near-optimal ones to maintain – in addition to 'improve all' objectives. Such property-specific objectives enables modeling diverse multi-property trade-offs, thereby capturing more realistic optimization scenarios. Furthermore, `C-MuMOInstruct` introduces property-specific thresholds, requiring each sub-optimal property to be improved up to a level considered sufficient for pharmaceutical success. This enables models tuned on `C-MuMOInstruct` to learn more targeted optimization strategies and navigate nuanced multi-property trade-offs more effectively than models tuned on datasets lacking finer control. Meanwhile, learning such nuanced and controllable optimization introduces additional modeling challenges, making `C-MuMOInstruct` a more practical and difficult dataset than existing ones.

(3) Comprehensive coverage: Spanning across 10 pharmacologically relevant molecular properties, `C-MuMOInstruct` covers a wide range of multi-property combinations, and multi-objective tasks with property-specific objectives for each property combination. This leads to a comprehensive set of optimization tasks, better capturing the complexity of real-world drug design.

(4) Pairwise optimization: Following `MuMOInstruct`, `C-MuMOInstruct` is constructed from molecule pairs that satisfy similarity, property improvement, and stability constraints. This enables models to effectively associate targeted structural modifications with property changes.

(5) Diverse instructions: `C-MuMOInstruct` provides diverse natural language instructions for each task with varied phrasings. This prevents instruction-tuned LLMs from overfitting to a specific phrasing, and enables them to generalize to unseen instructions – a crucial capability in practice, where task descriptions can widely vary.

2.2 Overview of `C-MuMOInstruct` Tasks

`C-MuMOInstruct` comprises a total of 28,266 tasks, with 27,401 tasks optimizing a combination of at least 3 properties. All tasks in `C-MuMOInstruct` are systematically curated by combining subsets of 10 pharmacologically relevant molecular properties: **(1) Penalized LogP (PlogP):** representing solubility, lipophilicity, synthetic accessibility, and ring complexity – higher PlogP is typically preferred in drug candidates; **(2) Quantitative Estimate of Drug-Likeness (QED):** assessing overall drug-likeness by incorporating molecular weight, lipophilicity, and hydrogen bonding ability – higher QED is desired for better drug-likeness; **(3) Parallel Artificial Membrane Permeability Assay (AMP):** evaluating drug permeability across the cellular membrane – higher AMP indicates improved drug absorption; **(4) Blood-Brain Barrier Permeability (BBBP):** representing the ability of a drug to permeate the blood-brain barrier – higher BBBP is essential for CNS drugs; **(5) human Intestinal Absorption (HIA):** indicating the ability of a drug to be absorbed through the gastrointestinal tract – higher HIA supports effective absorption of orally administered drugs; **(6) human Ether-à-go-go Related Gene inhibition (hERG):** referring to the drug's ability to inhibit the human ether-à-go-go related gene, which in turn blocks the potassium channel, causing severe cardiac issues – lower hERG is necessary to reduce cardiac risks; **(7) Carcinogenicity (CARC):** indicating the potential of a drug to induce cancer by damaging the genome or disrupting cellular processes – lower CARC is desired for safety; **(8) Mutagenicity (MUT):** referring to the likelihood of a drug causing genetic mutations – lower MUT scores are preferred to reduce genotoxicity; **(9) Drug-induced Liver Injury (LIV):** representing a drug's potential to induce liver damage (hepatotoxicity) – lower DILI is crucial to reduce toxicity; **(10) Dopamine Receptor D2 Inhibition (DRD2):** indicating binding affinity to dopaminergic pathways – higher DRD2 scores are desired for antipsychotic drugs targeting the DRD2 receptor.

We focus on these 10 properties due to their key role in determining a drug's pharmacokinetic behavior, toxicity risk, and overall drug-likeness – essential factors in real-world lead optimization. Moreover, these properties are well-studied and typically considered in existing optimization benchmarks (Gao et al., 2022; Dey et al., 2025). For evaluation, 10 representative property combinations (Section B) with 119 multi-objective tasks are selected and grouped into 51 IND and 68 OOD tasks. (Section 2.6). These tasks can be divided into 2 categories: **(1) General Drug-Likeness and Toxicity (`GT`):** tasks focused on broadly applicable molecular properties relevant for any successful drug candidate, irrespective of the specific therapeutic endpoint. **(2) Context-Specific Objectives (`CS`):** tasks involving properties that are specific to the therapeutic end-point, such as DRD2 inhibition or tissue-specific permeability (e.g., BBBP).

2.3 Constructing Task-Specific Training Pairs

Following Algorithm A1, we construct task-specific training pairs (M_x, M_y) from the dataset curated by (Chen et al., 2021), which contains 256K molecule pairs satisfying the similarity constraint (i.e., Tanimoto similarity > 0.6). Out of these pairs, we select those that satisfy all $\mathcal{P}_\mathtt{i}$ property improvement constraints (i.e., $(M_x \prec_{\Delta_p} M_y)_{\forall p \in \mathcal{P}_\mathtt{i}}$) and all $\mathcal{P}_\mathtt{s}$ property stability constraints (i.e., $(M_x \cong_{\Delta_p} M_y)_{\forall p \in \mathcal{P}_\mathtt{s}}$) for each task optimizing sub-optimal $\mathcal{P}_\mathtt{i}$ properties and near-optimal $\mathcal{P}_\mathtt{s}$ properties (Appendix B.1). For a given task with $\mathcal{P}$ properties, each property $p \in \mathcal{P}$ is considered sub-optimal or near-optimal based on Θ_p (shown in Table 2) as described earlier in Section 2. These thresholds are set to the 60th percentile of all training molecules among 256K pairs, reflecting desirable scores for an optimized lead molecule.

2.4 Constructing Task-Specific Test Set

We construct a test set by randomly sampling 250K molecules from ZINC (Sterling and Irwin, 2015), a widely used subset of commercially available molecules. All sampled molecules satisfy Lipsinki's rule of 5 (Lipinski et al., 2001), and do not overlap with the training set to ensure no data leakage. This creates an initial pool of drug-like molecules having some near-optimal properties with desirable scores, and some sub-optimal ones requiring further improvement. From this pool, we select a molecule M_x into the test set of a task improving $\mathcal{P}_\mathtt{i}$ and maintaining $\mathcal{P}_\mathtt{s}$ properties, if M_x has every property $p \in \mathcal{P}_\mathtt{i}$ worse than Θ_p, and every property $p \in \mathcal{P}_\mathtt{s}$ exceeding Θ_p. This selection ensures a representative test set for evaluation on diverse multi-objective tasks, given a specific property combination. Following this selection process,

Table 2: Summary of C-MuMOInstruct Tasks for Evaluation

		Properties										#Pairs	#Mols	#Test	#Tasks	Cat
Type	$\mathcal{P}$-Comb	AMP$^\uparrow$	BBBP$^\uparrow$	CARC$^\downarrow$	DRD2$^\uparrow$	hERG$^\downarrow$	HIA$^\uparrow$	LIV$^\downarrow$	MUT$^\downarrow$	PlogP$^\uparrow$	QED$^\uparrow$					
	$(\Delta_p =)$	0.1	0.1	0.2	0.1	0.2	0.1	0.1	0.1	1.0	0.1					
	$(\Theta_p =)$	0.8	0.8	0.2	0.4	0.3	0.4	0.9	0.2	1.5	0.9					
IND	BPQ	–	✓	–	–	–	–	–	–	✓	✓	700	1,371	500	7	CS
	ELQ	–	–	–	–	✓	–	✓	–	–	✓	700	1,376	500	7	GT
	ACEP	✓	–	✓	–	✓	–	–	–	✓	–	1,242	2,347	500	15	GT
	BDPQ	–	✓	–	✓	–	–	–	–	✓	✓	895	1,561	500	13	CS
	DHMQ	–	–	–	✓	–	✓	–	✓	–	✓	787	1,402	500	9	CS
OOD	CDE	–	–	✓	✓	✓	–	–	–	–	–	516	832	500	6	CS
	ABMP	✓	✓	–	–	–	–	–	✓	✓	–	1,500	2,809	500	15	CS
	BCMQ	–	✓	✓	–	–	–	–	✓	–	✓	1,398	2,696	500	15	CS
	BDEQ	–	✓	–	✓	✓	–	–	–	–	✓	603	840	500	11	CS
	HLMPQ	–	–	–	–	–	✓	✓	✓	✓	✓	1,800	3,329	500	21	GT

"$\mathcal{P}$-Comb" denotes the combination of $\mathcal{P}$ properties with multiple objectives. "#Pairs" and "#Mols", denote the number of molecule pairs and unique molecules in training, respectively. "#Test" and "#Tasks" denote the number of test samples and multi-property objectives for a specific property combination, respectively. "Cat" indicates task category. ✓indicates properties included in the task; – indicates properties not involved. $^\uparrow$ and $^\downarrow$ indicate whether higher or lower scores of a given property are desirable.

we randomly sample 500 molecules for each of 10 representative property combinations in evaluation.

2.5 Quality Control

We implement several quality control measures, detailed in Appendix B.2, to ensure the integrity and rigor of C-MuMOInstruct. We eliminate duplicate molecules by comparing their canonicalized SMILES representations. We compute all molecular property scores empirically using established and widely-used tools such as ADMET-AI (Swanson et al., 2024). To promote robustness in instruction following, we curate 30 distinctly phrased instructions that convey the same optimization objective using varied semantics (Appendix C). To assess LLMs' ability to generalize beyond seen instructions, we hold out one instruction per task during training and use it only during inference.

2.6 IND and OOD Tasks

To rigorously evaluate instruction-tuned LLMs on both familiar and novel optimization scenarios, we split the 10 evaluation tasks into 2 groups:

In-Distribution (IND) Tasks: IND tasks are defined by property combinations that appear in the training set. Performance on these tasks assess how effectively the model can apply its learned modification strategies to the exact property combinations and objectives it was specifically trained on.

Out-of-Distribution (OOD) Tasks: OOD tasks involve novel multi-property combinations and novel multi-property objectives for each combination that are not used during training (i.e., unseen C-MuMO tasks). Note that although OOD property combinations are not used in training, each individual property is still used as part of other combinations in the training tasks. Success in OOD tasks demonstrates the model's ability to transfer its knowledge to novel property combinations and novel multi-objective tasks for each unseen property combination without task-specific fine-tuning. This ability is crucial in practice, where emerging therapeutic goals often necessitate adapting to previously unseen multi-property trade-offs.

3 GeLLM^{4}O-C Models

We introduce GeLLM^{4}O-Cs, a series of general-purpose LLMs instruction-tuned over C-MuMOInstruct. Each training sample in C-MuMOInstruct consists of a molecule pair (i.e., a hit molecule improved to a lead molecule via substructure modifications) and a corresponding natural language instruction that explicitly specifies the desired property-specific optimization objective (i.e., which properties to improve and up to how much, and which properties to maintain). Upon instruction-tuning on such samples, GeLLM^{4}O-C learns to associate the structural differences observed in molecule pairs with the desired property-specific objectives expressed via natural language instructions.

Thus, GeLLM^{4}O-C implicitly captures patterns of how specific substructure modifications correlate with change in multiple properties (i.e., structure-property relationships) (Hansch, 1969) and applies such knowledge to a given molecule during inference. Such knowledge of modification strategies and structure-property relationships is mapped to LLM parameters via instruction tuning. Unlike existing optimization methods that require explicitly designed scoring functions or reward shaping to balance multi-property trade-offs, GeLLM^{4}O-Cs implicitly learn these trade-offs by observing multiple paired molecules across diverse C-MuMO tasks. This allows for explicit control over each property with varying objectives.

We develop both specialist and generalist GeLLM^{4}O-Cs. Each specialist GeLLM^{4}O-C, denoted as GeLLM^{4}O-C-N, is fine-tuned on a single property combination of N properties, with multiple objectives in that specific combination. This enables them to learn focused modification strategies and capture specific trade-offs observed in that property combination. In contrast, each generalist GeLLM^{4}O-C, denoted as GeLLM^{4}O-C-P(N), is trained across multiple property combinations (i.e., all combinations in the power set of N properties) and multiple objectives within each combination. This multi-task training enables GeLLM^{4}O-C-P(N) to generalize learned structure-property relationships across multiple combinations and adapt to unseen property combinations and tasks at inference, without task-specific retraining.

Concretely, we develop a series of generalist GeLLM^{4}O-Cs, denoted as GeLLM^{4}O-C-P(N), each is jointly trained on multiple C-MuMO tasks involving diverse multi-property, multi-objective combinations with up to N properties. 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) on every projection layer and the language modeling head. We chose LoRA due to its widespread adoption and efficiency in fine-tuning large models under limited compute. While the choice between LoRA and full fine-tuning is orthogonal to the scientific contributions of this work, we use LoRA and set hyperparameters following well-established practices (Dey et al., 2025). We perform 0-shot evaluations (i.e., without in-context examples) for all GeLLM^{4}O-Cs. For each input molecule, we generate 20 candidates via beam search decoding (Appendix D.1).

4 Experimental Setup

4.1 Baselines

We compare GeLLM^{4}O-Cs against 2 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 and GPT-4o; and **(2)** foundational LLMs for chemistry: a Mistral-7B fine-tuned on diverse molecular tasks (Yu et al., 2024), denoted as LlaSMol$_{\text{Mistral}}$. We use few-shot prompting with only 1 in-context example for all general-purpose LLMs to balance generation quality with computational resources and expenses. For baselines that support beam-search decoding, we generate 20 candidate molecules per input using the same generation strategy as in GeLLM^{4}O-C. Additional details and prompts are in Appendix D.2 and Appendix E, respectively.

Our work introduces a new and practically motivated setting with C-MuMO tasks in which each property is associated with a distinct improvement or stability objective. To the best of our knowledge, existing reinforcement learning (Blaschke et al., 2020; You et al., 2018; Gao et al., 2022) and genetic algorithm-based methods (Jensen, 2019; Sun et al., 2022; Kim et al., 2024) are not directly applicable to this setting without significant redesign and engineering effort. These methods typically optimize scalarized multi-objective rewards, and do not support property-specific objectives or mixed improvement and stability constraints. Adapting these methods to our setting would require significant effort on reward shaping for each of the 28K C-MuMO tasks, which is non-trivial and beyond the scope of this work. Therefore, we considered strong off-the-shelf baselines capable of following natural-language instructions: SOTA general-purpose LLMs and LLMs for chemistry.

4.2 Evaluation Metrics

We employ multiple evaluation metrics (detailed in Appendix D.3) to enable a comprehensive assessment. For clarity and brevity, we report results primarily using the following metrics: **(1) Success Rate** (SR): the proportion of input molecules successfully optimized, such that all sub-optimal properties are improved, and all near-optimal ones are maintained within their corresponding Δ_p – reflecting the model's ability to follow property-specific objectives; **(2) Similarity with input** (Sim): the average Tanimoto similarity (Bajusz et al., 2015) between the optimized and corresponding input molecule; **(3) Relative Improvement** (RI): the relative improvement averaged across all sub-optimal properties. Higher SR, Sim, and RI are preferred, denoting more successful and effective optimizations. In Appendix G, we report results with a stricter notion of success, via SR$_\Theta$, measuring suc-

Table 3: Overall Performance in IND Tasks

Model	BPQ			ELQ			ACEP			BDPQ			DHMQ		
	SR$^{\uparrow}$	Sim$^{\uparrow}$	RI$^{\uparrow}$	SR$^{\uparrow}$	Sim$^{\uparrow}$	RI$^{\uparrow}$	SR$^{\uparrow}$	Sim$^{\uparrow}$	RI$^{\uparrow}$	SR$^{\uparrow}$	Sim$^{\uparrow}$	RI$^{\uparrow}$	SR$^{\uparrow}$	Sim$^{\uparrow}$	RI$^{\uparrow}$
General-purpose LLMs															
Mistral (0-shot)	28.80	**0.75**	1.24	21.60	0.72	0.16	26.20	0.75	1.10	2.40	**0.72**	0.49	4.80	0.71	0.76
Llama (0-shot)	33.60	0.70	0.78	16.60	**0.74**	0.10	17.20	0.74	0.69	8.80	**0.72**	1.67	6.00	**0.73**	1.35
Claude-3.5 (0-shot)	51.80	0.68	0.89	20.00	0.64	0.20	29.60	0.71	0.69	11.20	0.67	1.80	5.20	0.63	1.84
GPT-4o (0-shot)	30.20	0.72	0.55	16.60	0.72	0.10	22.20	0.74	0.52	4.20	**0.72**	3.98	5.80	0.72	0.88
Mistral (1-shot)	72.80	0.63	1.26	74.80	0.59	0.28	63.80	0.64	1.03	21.60	0.59	4.76	25.60	0.55	1.89
Llama (1-shot)	49.60	0.68	0.95	36.80	0.68	0.15	40.20	0.70	1.12	14.40	0.63	2.65	13.80	0.56	**3.39**
Claude-3.5 (1-shot)	61.80	0.65	1.31	29.20	0.63	0.21	32.60	0.71	1.24	15.60	0.58	3.99	8.40	0.65	1.38
GPT-4o (1-shot)	28.60	0.74	0.77	19.60	0.72	0.12	23.00	**0.76**	1.09	5.60	0.68	3.47	5.60	0.71	1.22
Foundational LLMs for Chemistry															
LlaSMol-M	78.20	0.64	0.92	81.40	0.62	0.28	68.60	0.66	1.00	22.60	0.68	2.22	24.80	0.62	1.44
Specialist LLMs															
GeLLM^{4}O-C-N$_{\text{Mistral}}$	71.00	0.57	2.59	81.80	0.55	0.39	85.60	0.54	**2.46**	**56.60**	0.50	5.48	44.60	0.57	2.96
GeLLM^{4}O-C-N$_{\text{Llama}}$	84.20	0.58	2.09	85.40	0.53	**0.41**	88.00	0.54	2.24	43.60	0.58	4.85	35.40	0.65	2.63
Impv-Spec (%)	7.7	-9.4	127.2	4.9	-14.5	46.4	28.3	-18.2	124.0	150.4	-26.5	146.8	74.2	3.6	56.6
Generalist LLMs															
GeLLM^{4}O-C-P(N)$_{\text{Mistral}}$	84.80	0.63	2.64	83.20	0.63	0.33	86.60	0.60	2.34	50.60	0.58	4.93	**53.40**	0.59	3.26
GeLLM^{4}O-C-P(N)$_{\text{Llama}}$	88.80	0.62	2.16	**90.80**	0.63	0.34	**92.80**	0.58	2.22	51.00	0.58	5.40	50.40	0.59	3.28
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**89.40**	0.62	2.30	88.40	0.59	**0.41**	74.60	0.61	1.92	48.40	0.58	5.05	52.20	0.61	2.24
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	79.40	0.57	**2.67**	79.00	0.56	**0.41**	72.60	0.57	2.27	42.60	0.55	**5.89**	41.80	0.57	3.32
Impv-Gen (%)	14.3	-3.1	150.0	11.5	1.6	21.4	35.3	-12.1	122.0	125.7	-14.7	143.2	108.6	7.3	72.5

$^{\uparrow}$ and $^{\downarrow}$ indicate whether a higher or lower metric is preferred, respectively. For each task, the best-performing model is in **bold**, and the best baseline is underlined. Impv-Spec and Impv-Gen represent the percentage improvement from the best specialist LLM and best generalist LLM over the best baseline, respectively. The best model in each group is selected based on SR for each task.

cess only if each property in the task exceeds Θ_p.

SR serves as the primary metric for evaluation, measuring the overall success rate of optimization. In contrast, Sim and RI offer complementary insights into the quality of successful optimizations. Importantly, Sim and RI are computed only over optimized molecules that contribute to SR, that is, over those that satisfy all specified property constraints, and not over all generated molecules. Thus, these metrics should be interpreted in conjunction with SR. For instance, a model achieving high Sim and RI but low SR indicates poor optimization quality since the model fails to optimize most input molecules. Moreover, high Sim is not an indicator of successful optimization. Rather, it reflects better scaffold preservation in the optimized molecules – a desirable criterion in hit-to-lead optimization. In fact, high Sim with low SR reflects minimal and ineffective modification, leading to failed optimization, which is why SR is the primary metric.

5 Experimental Results

Main Findings: The key findings are summarized as: **(1)** Both specialist and generalist GeLLM^{4}O-Cs consistently surpass general-purpose LLMs and foundational LLMs for chemistry across all IND (Section 5.1) and OOD tasks (Section 5.2), achieving up to 126% higher SR and 143% higher RI. **(2)** Generalist GeLLM^{4}O-Cs outperform specialist GeLLM^{4}O-Cs on 4 out of 5 IND combinations, with 26% more successful optimizations on challenging tasks, such as DHMQ (Section 5.1). **(3)** Generalist GeLLM^{4}O-Cs demonstrate remarkable 0-shot generalization to OOD tasks, surpassing the best general-purpose LLMs by 35% in SR and 76% in RI (Section 5.2). **(4)** Generalist GeLLM^{4}O-Cs exhibit strong generalization when prompted with unseen instructions across all IND tasks (Section 5.3).

5.1 IND Tasks

Table 3 presents the performance comparison of GeLLM^{4}O-Cs and baselines across all IND tasks. Detailed task-specific results are in Appendix G.1.

Overall Comparison: Across all IND tasks, all specialist and generalist GeLLM^{4}O-Cs consistently outperform all baselines. Notably, the generalist GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ outperforms the best baseline by 37% and 102% in SR and RI on average, indicating its superior ability as a foundational model to perform targeted modification across diverse C-MuMO tasks. On two challenging tasks, BDPQ and DHMQ, with a specific therapeutic endpoint (DRD2 inhibition), both specialist and generalist GeLLM^{4}O-Cs successfully optimize as much as 150% and 126% more input molecules than the baselines, with even 1-fold better RI. Such strong performance demonstrates the ability of

GeLLM^{4}O-Cs to tackle complex property trade-offs.

Furthermore, when evaluated under the stricter success criteria (via SR$_\Theta$) – which requires each property to exceed pharmaceutically relevant thresholds (i.e., Θ_p) – the performance gap between GeLLM^{4}O-Cs and baselines becomes even more pronounced. Table A2 demonstrates that generalist GeLLM^{4}O-Cs outperform the best baseline by as much as 218% in SR and 313% in RI. This highlights the ability of GeLLM^{4}O-Cs to not only optimize more molecules, but also to improve each desired property up to significant levels.

Comparison between specialist and generalist GeLLM^{4}O-C: Table 3 demonstrates that generalist GeLLM^{4}O-Cs outperform specialist ones on 4 out of 5 IND combinations, with particularly large gains on the challenging DHMQ tasks. This trend is prominent in tasks with fewer task-specific training pairs, such as BPQ, ELQ, and DHMQ, where generalist models outperform specialist ones by up to 26% in SR. Limited training pairs in these tasks hinder the specialist models to learn robust modification strategies. In contrast, generalist ones benefit from transferable knowledge of property trade-offs and learn optimization strategies from other diverse multi-property, multi-objective training tasks.

Interestingly, in the BDPQ tasks, despite having only 895 pairs, GeLLM^{4}O-C-N$_{\text{Mistral}}$ outperforms all generalist ones. The generalist variant, GeLLM^{4}O-C-P(N), – trained only on tasks involving BBBP, DRD2, PlogP and QED – remains competitive due to its focused training on these specific properties. In contrast, GeLLM^{4}O-C-P(10) – trained on all possible property combinations involving up to 10 properties – performs worse than GeLLM^{4}O-C-P(N) and specialist GeLLM^{4}O-C. This could be due to GeLLM^{4}O-C-P(10) encountering tasks with competing or conflicting objectives, which weakens its ability to specialize in BDPQ-specific trade-offs. This highlights a key challenge in developing foundational models: while multi-task tuning promotes cross-task knowledge transfer, it may also introduce conflicts that negatively impact performance on specialized tasks (e.g., BDPQ).

Comparison with general-purpose LLMs: Table 3 shows that all GeLLM^{4}O-Cs consistently outperform all general-purpose LLMs across all IND tasks, achieving up to 109% higher SR than the best general-purpose LLM, Mistral (1-shot). This strong performance gap underscores the benefit of instruction tuning on molecule pairs, which enables GeLLM^{4}O-Cs to learn robust and effective modification strategies that are difficult for general-purpose LLMs to learn through in-context examples alone. Moreover, general-purpose LLMs exhibit lower RI among the limited successfully optimized molecules, compared to GeLLM^{4}O-Cs. This demonstrates the ability of GeLLM^{4}O-Cs to perform more targeted modifications to yield substantial improvements on each sub-optimal property.

Comparison with foundational LLMs for chemistry: All GeLLM^{4}O-Cs substantially outperform the SoTA foundational LLM for chemistry, LlaSMol$_{\text{Mistral}}$, on all IND tasks. Another foundational LLM, ChemDFM, performs worse than LlaSMol (Appendix G). Notably, on BDPQ and DHMQ, GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ achieves a 126% and 115% higher SR, respectively, with higher RI by 143% and 126%, respectively, compared to LlaSMol$_{\text{Mistral}}$. While LlaSMol is instruction-tuned on a broad range of molecular tasks, GeLLM^{4}O-Cs are specifically instruction-tuned on different multi-property optimization tasks. This highlights the efficacy of instruction-tuning on optimization tasks to learn targeted modifications and navigate multi-property trade-offs. Appendix F presents 2 cases of such targeted modifications.

5.2 OOD Tasks

Table 4 presents the performance of GeLLM^{4}O-Cs and baselines across all OOD tasks. Since GeLLM^{4}O-C-Ns and GeLLM^{4}O-C-P(N) models use task-specific pairs, they are inapplicable to OOD tasks. Overall, generalist GeLLM^{4}O-Cs exhibit strong 0-shot generalization to novel C-MuMO tasks, consistently outperforming all baselines. Specifically, the best-performing generalist model, GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$, achieves an average SR of 63% across all tasks, outperforming the best baseline, Mistral (1-shot), by as much as 35% and 77% in SR and RI, respectively. These strong results demonstrate the remarkable ability of generalist GeLLM^{4}O-Cs to learn transferable optimization strategies and tackle unseen controllable property-specific objectives during inference. Such generalizability is crucial in practice, where evolving therapeutic goals often introduce novel property combinations and novel objectives.

5.3 Generalizability to Unseen Instructions

Table 5 compares specialist GeLLM^{4}O-Cs with generalist GeLLM^{4}O-Cs when evaluated with a hold-out

Table 4: Overall Performance in OOD Tasks

Model	CDE			ABMP			BCMQ			BDEQ			HLMPQ		
	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$
General-purpose LLMs															
Mistral (0-shot)	3.00	0.73	1.33	23.00	**0.77**	0.93	25.40	0.69	0.25	3.00	**0.71**	1.05	11.60	**0.79**	**1.76**
Llama (0-shot)	6.80	0.68	0.77	44.60	0.71	0.61	20.40	0.72	0.20	2.20	0.68	0.60	20.20	0.72	0.68
Claude-3.5 (0-shot)	6.80	0.70	1.07	43.60	0.70	0.80	30.00	0.64	0.26	4.80	0.62	0.57	21.00	0.66	0.59
GPT-4o (0-shot)	3.80	**0.74**	1.56	27.00	0.73	0.51	19.60	0.72	0.19	3.40	**0.71**	0.42	12.80	0.72	0.47
Mistral (1-shot)	30.60	0.62	**1.66**	73.20	0.64	1.09	63.80	0.60	0.31	21.60	0.58	1.16	55.60	0.62	0.77
Llama (1-shot)	18.20	0.55	1.51	60.80	0.70	0.83	41.60	0.67	0.23	11.40	0.51	**1.54**	28.00	0.70	0.75
Claude-3.5 (1-shot)	8.40	0.66	1.09	45.20	0.64	0.87	32.40	0.61	0.30	7.20	0.55	1.22	25.00	0.61	0.72
GPT-4o (1-shot)	7.00	0.72	1.04	34.40	0.74	0.65	23.40	**0.73**	0.21	2.20	0.70	0.83	13.40	0.71	0.65
Foundational LLMs for Chemistry															
LlaSMol$_{\text{Mistral}}$	29.80	0.61	1.28	72.40	0.67	0.78	72.80	0.63	0.30	18.20	0.60	0.65	37.80	0.68	0.66
Generalist LLMs															
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**39.80**	0.58	**1.66**	**86.60**	0.63	1.68	**84.20**	0.62	0.42	**29.20**	0.60	1.22	**74.60**	0.61	1.36
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	33.20	0.55	1.50	79.60	0.58	**1.81**	80.00	0.57	**0.44**	28.40	0.58	0.88	65.40	0.58	1.35
Impv-Gen (%)	30.1	-6.5	0.0	18.3	-1.6	54.1	15.7	-1.6	40.0	35.2	3.4	5.2	34.2	-1.6	76.6

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

Table 5: Overall Performance with Unseen Instructions in IND Tasks

Model GeLLM^{4}O-C	Instr	BPQ			ELQ			ACEP			BDPQ			DHMQ		
		SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$
Specialist LLMs																
-N$_{\text{Mistral}}$	seen	71.00	0.57	**2.59**	81.80	0.55	0.39	85.60	0.54	**2.46**	56.60	**0.50**	5.48	44.60	0.57	2.96
	unseen	68.60	0.55	2.33	84.60	0.53	**0.41**	86.80	0.53	2.28	59.40	0.47	**5.79**	**49.40**	0.56	**3.19**
-N$_{\text{Llama}}$	seen	**84.20**	0.58	2.09	85.40	0.53	0.41	88.00	0.54	2.24	**43.60**	0.58	4.85	35.40	0.65	2.63
	unseen	74.20	0.57	2.02	88.60	0.54	0.42	87.00	0.52	2.14	37.00	0.59	**5.27**	**37.60**	0.64	**2.77**
Generalist LLMs																
-P(10)$_{\text{Mistral}}$	seen	89.40	0.62	**2.30**	88.40	0.59	**0.41**	74.60	0.61	**1.92**	48.40	0.58	**5.05**	52.20	0.61	2.24
	unseen	89.60	0.62	2.01	87.60	0.60	0.37	78.00	0.63	1.75	46.60	0.60	4.57	50.20	0.61	**2.79**
-P(10)$_{\text{Llama}}$	seen	79.40	0.57	2.67	79.00	0.56	0.41	72.60	0.57	2.27	42.60	0.55	5.89	41.80	0.57	**3.32**
	unseen	**95.60**	0.55	2.63	**92.60**	0.55	0.42	**84.80**	0.57	2.21	**52.80**	0.55	5.67	**51.60**	0.55	2.96

'Seen' and 'unseen' indicate whether models are evaluated using instructions included during training or entirely novel instructions, respectively. $^\uparrow$ and $^\downarrow$ indicate whether higher or lower values of the corresponding metric are preferable. Within each row block, the best-performing model is highlighted in bold if the performance difference exceeds 5%.

instruction and property name (Appendix C). Overall, specialist GeLLM^{4}O-Cs exhibit a performance drop of over 5% in SR on 2 out of 5 IND combinations. In contrast, generalist GeLLM^{4}O-Cs retain consistent performance on all tasks. This indicates that generalist models – trained on more tasks and instructions – can generalize better to unseen instructions with different phrasings. Such generalizability is crucial in practice, where task instructions can vary widely. Notably, GeLLM^{4}O-C-P(10)$_{\text{Llama}}$ demonstrates more robustness than GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$, reflecting a reduced tendency to overfit to specific wordings.

6 Additional Analyses

We conduct ablation studies to examine the effects of (1) the number of properties included during instruction tuning of generalist models, and (2) instruction template diversity. We find that models trained on a moderate number of properties and diverse instruction templates outperform others across multiple IND tasks (Appendix G.3). Additionally, we perform a failure analysis of the most difficult objectives, revealing that DRD2 improvement consistently has the highest constraint violation rates, and that improving properties is more challenging than stabilizing them (Appendix G.4).

7 Conclusion

In this paper, we introduced C-MuMOInstruct, the first instruction-tuning dataset enabling controllable molecule optimization with property-specific objectives. Leveraging C-MuMOInstruct, we developed GeLLM^{4}O-Cs, that consistently and largely outperform strong general-purpose LLMs and foundational LLMs for chemistry across all IND and OOD tasks. Moreover, generalist GeLLM^{4}O-Cs exhibit strong generalization to unseen tasks, outperforming baselines by 27% on average. This indicates the potential of GeLLM^{4}O-C as a foundational model to tackle diverse tasks with realistic, controllable objectives reflecting real-world scenarios.

8 Limitations

While our work represents a significant step toward controllable, multi-objective molecule optimization, several limitations remain: **(1)** Our current framework is designed for single-step optimization. In practice, optimizing molecules to reach pharmaceutically meaningful thresholds for all properties may require multiple iterative modifications. Designing a feedback mechanism for GeLLM^{4}O-C or intermediate reward signal to guide iterative refinement is non-trivial and is a direction for future work. **(2)** We rely on computational predictors for molecular properties. Although they are well-established and widely used, they may introduce inaccuracies and may not always reflect exact experimental outcomes. Incorporating experimentally validated datasets or feedback to LLMs with wet-lab data is a promising direction for future work. **(3)** Although we demonstrate strong generalization to unseen instructions, our instruction templates are still synthetically generated. Future work could explore more diverse linguistic variation to test LLM robustness in truly open-ended settings.

9 Impact Statement

This work presents the first instruction-tuning dataset, C-MuMOInstruct, that explicitly supports property-specific objectives in multi-property molecule optimization – enabling models to selectively improve sub-optimal properties while preserving near-optimal ones. Built on this dataset, our developed instruction-tuned LLMs (GeLLM^{4}O-C) represent a substantial advancement toward controllable molecule optimization, addressing practical drug design requirements often overlooked by existing approaches. GeLLM^{4}O-Cs consistently outperform both strong general-purpose LLMs and foundational LLMs for chemistry across challenging optimization tasks involving conflicting objectives. By demonstrating robust generalization to novel property combinations and novel multi-property constraints, GeLLM^{4}O-C paves the way for scalable, general-purpose foundation LLMs that can flexibly handle diverse drug design constraints. We anticipate that GeLLM^{4}O-C will serve as a building block for future iterative LLM optimization frameworks.

Broader Impacts: The development of foundational LLMs for controllable multi-property molecule optimization represents a significant step toward AI-based molecular design tools. Their ability to follow property-specific instructions enables iterative optimization workflows, where molecules are refined over multiple steps based on intermediate feedback – a common and necessary paradigm in real-world lead optimization. Through natural language instructions, these models can be flexibly adapted to a variety of drug design scenarios without extensive retraining. Such flexibility lowers the barrier to deploying intelligent drug design pipelines, especially for researchers with limited computational or domain resources. Ultimately, such scalable and generalizable frameworks have the potential to accelerate early-stage drug development, reduce experimental burden, and democratize access to advanced drug design capabilities.

10 Ethics Statement

Our work introduces instruction-tuning dataset, C-MuMOInstruct and GeLLM^{4}O-Cs tuned on C-MuMOInstruct for multi-property molecule optimization. While C-MuMOInstruct is curated with drug-like molecule and to improve pharmaceutically relevant and desirable properties, we cannot fully guarantee the absence of harmful compounds or the potential for misuse. Notably, 4 of the 10 properties in C-MuMOInstruct – carcinogenicity, hERG inhibition, drug-induced liver injury, and mutagenicity – are directly related to drug toxicity. Our models are explicitly tuned to minimize these property scores, and thus, to improve drug safety profiles aligned with widely accepted pharmacological desirability. The objective is to generate drug-like molecules with reduced toxicity, not to increase toxicity or discover harmful compounds.

Given that our models are fine-tuned on general-purpose open-source LLMs, they may still retain knowledge about toxic substructures or chemicals from the broader pretraining corpus. While our instruction-tuning encourages models to generate molecules with more pharmaceutically desirable profiles, we cannot fully eliminate the possibility of generating undesirable molecules if misused or prompted adversarially.

We strongly discourage any application of GeLLM^{4}O-Cs outside responsible drug discovery research. Deployment of these models should be accompanied by toxicity screening, expert review, and strong usage controls. We expect all users of our dataset and models to uphold the highest standards of ethical research and to take appropriate precautions to prevent unintended consequences.

Acknowledgements

This project was made possible, in part, by support from the National Science Foundation grant nos. IIS-2133650 and IIS-2435819, and the National Library of Medicine grant no. 1R01LM014385. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency. We would like to thank our colleagues, Hanwen Du and Frazier Baker, for insightful discussions and for providing feedback that helped enrich the quality of this paper.

References

2025. Rdkit: Open-source cheminformatics.

Mistral AI. 2023. Mistral 7b. *arXiv preprint*.

Jaqueline S. Angelo, Isabella A. Guedes, Helio J. C. Barbosa, and Laurent E. Dardenne. 2023. Multi-and many-objective optimization: present and future in de novo drug design. *Frontiers in Chemistry*, 11.

Reza Averly, Frazier N. Baker, and Xia Ning. 2025. Liddia: Language-based intelligent drug discovery agent. *Preprint*, arXiv:2502.13959.

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.

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.

Andrea Cavalli, Elisabetta Poluzzi, Fabrizio De Ponti, and Maurizio Recanatini. 2002. Toward a pharmacophore for drugs inducing the long qt syndrome: insights from a comfa study of herg k+ channel blockers. *Journal of medicinal chemistry*, 45(18):3844–3853.

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).

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.

Vishal Dey, Xiao Hu, and Xia Ning. 2025. Gellmˆ3o Generalizing large language models for multi-property molecule optimization. *arXiv preprint arXiv:2502.13398*.

Peter Ertl and Ansgar Schuffenhauer. 2009a. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1).

Peter Ertl and Ansgar Schuffenhauer. 2009b. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun 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. Mimosa: 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.

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, and 2 others. 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.

Corwin Hansch, Albert Leo, David Hoekman, and 1 others. 1995. *Exploring QSAR: hydrophobic, electronic, and steric constants*, volume 2. American Chemical Society Washington, DC.

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*.

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.

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, Saee 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*.

Paul D Leeson and Brian Springthorpe. 2007. The influence of drug-like concepts on decision-making in medicinal chemistry. *Nature reviews Drug discovery*, 6(11):881–890.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. 2001. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings 1pii of original article: S0169-409x(96)00423-1. the article was originally published in advanced drug delivery reviews 23 (1997) 3–25. 1. *Advanced Drug Delivery Reviews*, 46(1–3):3–26.

Shengchao Liu, Jiongxiao 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*.

Nicholas A Meanwell. 2011a. Improving drug candidates by design: a focus on physicochemical properties as a means of improving compound disposition and safety. *Chemical research in toxicology*, 24(9):1420–1456.

Nicholas A Meanwell. 2011b. Synopsis of some recent tactical application of bioisosteres in drug design. *Journal of medicinal chemistry*, 54(8):2529–2591.

Nicholas A Meanwell. 2016. Improving drug design: an update on recent applications of efficiency metrics, strategies for replacing problematic elements, and compounds in nontraditional drug space. *Chemical Research in Toxicology*, 29(4):564–616.

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.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Michael C. Sanguinetti and Martin Tristani-Firouzi. 2006. herg potassium channels and cardiac arrhythmia. *Nature*, 440(7083):463–469.

P. Seeman, T. Lee, M. Chau-Wong, and K. Wong. 1976. Antipsychotic drug doses and neuroleptic/dopamine receptors. *Nature*, 261(5562):717–719.

Philip Seeman. 2001. Antipsychotic drugs, dopamine receptors, and schizophrenia. *Clinical Neuroscience Research*, 1(1):53–60.

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.

Paul D. Shepard, Carmen C. Canavier, and Edwin S. Levitan. 2007. Ether-a-go-go–related gene potassium channels: What's all the buzz about? *Schizophrenia Bulletin*, 33(6):1263–1269.

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.

Mengying Sun, Jing Xing, Han Meng, Huijun Wang, Bin Chen, and Jiayu Zhou. 2022. 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, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 2 others. 2023. Llama 2: Open foundation and fine-tuned chat models.

Hicham Wahnou, Fouzia Hmimid, Ahmed Errami, Imane Nait Irahal, 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, Lingkai 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, and 3 others. 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, and 1 others. 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*.

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. Chemllm: A chemical large language model. *Preprint*, arXiv:2402.06852.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, Kai Yu, and Xin Chen. 2025. Developing chemdfm as a large language foundation model for chemistry. *Cell Reports Physical Science*, 6(4):102523.

A Related Work

Computational approaches have primarily focused on single- or double-property optimization tasks (You et al., 2018; Blaschke et al., 2020; Xie et al., 2021; Bung et al., 2022; Sun et al., 2022). Graph-based methods such as Modof (Chen et al., 2021), MIMOSA (Fu et al., 2021), and f-RAG (Lee et al., 2024) perform substructure modifications on molecular graphs, while sequence-based methods like Chemformer (Irwin et al., 2022) and Prompt-MolOpt (Wu et al., 2024), formulate optimization as translation tasks over SMILES strings. Genetic algorithm-based methods, GraphGA (Jensen, 2019) and MolLeo (Wang et al., 2025) can optimize multiple properties but generate entirely new molecular scaffolds, limiting their practical utility. Furthermore, existing methods (Jensen, 2019; Wang et al., 2025; Kim et al., 2024; Yang et al., 2021), require task-specific fine-tuning and expert-curated reward functions to model multi-property trade-offs, limiting their scalability and applicability.

Recently, LLMs have demonstrated great promise for molecule optimization through natural language instructions (Chang et al., 2024). ChatDrug (Liu et al., 2024) and Re3DF (Le and Chawla, 2024) adopt multi-turn dialogue frameworks for iterative optimization. However, their reliance on closed-source APIs leads to high costs. DrugAssist (Ye et al., 2025) developed task-specific instruction-tuned LLMs limited to optimization tasks with up to 2 properties. Dey et al. (2025) introduced `MuMOInstruct` – a large-scale instruction-tuning dataset specifically focused on multi-property optimization tasks involving 3 or more properties – and further demonstrated the remarkable generalization abilities of instruction-tuned LLMs. However, `MuMOInstruct` does not provide controllable property-specific objectives required to mimic realistic C-MuMO tasks.

B Details on C-`MuMOInstruct`

B.1 Details on Task Construction

Algorithm A1 presents a pseudocode for constructing all valid C-MuMO tasks with all possible property combinations involving up to $\mathcal{P}$ properties, given a molecule pair (M_x, M_y). This algorithm ensures that all selected molecule pairs meet both the similarity and stability constraints required for C-MuMO tasks. To construct C-`MuMOInstruct`, we run Algorithm A1 on a random sample of 100K molecule pairs sourced from Chen et al. (2021). To create training pairs for a given combination with N properties, we select only those tasks out of all C-MuMO tasks that have all N properties involved. For example, to create task-specific training pairs for `BDPQ`, we select only tasks that involve all 4 properties: $\mathcal{T}_{\text{BDPQ}} = \{t = (M_x, M_y, \mathcal{C}_i, \mathcal{C}_s) \in \mathcal{T} \mid (\mathcal{C}_i \cup \mathcal{C}_s) = \mathcal{P}\}$ where $\mathcal{P} = \{$BBBP, DRD2, PlogP and QED$\}$.

We use at most 100 molecule pairs for each C-MuMO task (i.e., a unique property combination with explicit property-specific objectives) to balance efficiency and task diversity. Given that C-`MuMOInstruct` contains over 28K such tasks, training a generalist model with all possible pairs would be computationally prohibitive and may overemphasize overrepresented tasks. Limiting the number of examples per task ensures that the instruction-tuned model is exposed to a broad spectrum of multi-property trade-offs without biasing toward specific tasks. This design supports better generalization across diverse optimization objectives while keeping training tractable.

B.2 Details on Quality Control

To ensure a high-quality instruction-tuning dataset, we applied a series of quality control procedures.

Molecule Deduplication and Canonicalization: All molecules in C-`MuMOInstruct` are represented using canonical SMILES strings (Weininger, 1988), standardized via RDKit (rdk, 2025). We remove molecules with identical canonicalized SMILES that are structurally equivalent, thereby eliminating redundancy and ensuring that each molecule appears only once.

Empirical Property Computation: C-`MuMOInstruct` uses computationally predicted scores to annotate each molecule with 10 pharmacologically relevant molecular properties. These scores are computed using well-established, high-performing tools widely used in the molecular machine learning community. Specifically, we adopt the official implementation from You et al. (2018) for computing DRD2 and PlogP scores, and leverage the ADMET-AI tool (Swanson et al., 2024) to compute all other properties. These tools rank among the top-performing predictors in the Therapeutics Data Commons (TDC) benchmark (Catacutan et al., 2024), and have been extensively validated and adopted in recent studies (Wei et al., 2024; Thomas et al., 2024; Wahnou et al., 2024; Dey et al., 2025; Averly et al.,

Algorithm A1: C-MuMO Task Construction from a Molecule Pair

```
Input: Molecule pair $(M_x, M_y)$, Pharmaceutically-relevant levels $\{\Theta_p\}$, Improvement thresholds
        $\{\Delta_p\}$, Set of properties $\mathcal{P}$
Output: List of valid C-MuMO tasks $\mathcal{T}$ for $(M_x, M_y)$ with at most $\mathcal{P}$ properties
Initialize $\mathcal{T} \leftarrow \emptyset$ ;
foreach $p \in \mathcal{P}$ do
    Compute change$[p] \leftarrow p(M_y) - p(M_x)$ ;
    Set dir$[p] \leftarrow$ (change$[p] > 0$) if higher $p$ is desirable, else negative ;
// Identify Sub-optimal and near-optimal Properties:
$\mathcal{P}_{\texttt{i}} \leftarrow \{p \in \mathcal{P}_{\texttt{i}} \mid \texttt{abs(change)}[p] > \Delta_p\}$ ;
$\mathcal{P}_{\texttt{s}} \leftarrow \{p \in \mathcal{P}_{\texttt{s}} \mid \texttt{abs(change)}[p] \leq \Delta_p$ and $p(M_x) \succeq \Theta_p\}$ ;
foreach property subset $\mathcal{C} \subseteq \mathcal{P}$ with $|\mathcal{C}| \geq 1$ do
    $\mathcal{C}_i \leftarrow C \cap \mathcal{P}_{\texttt{i}}$                // Identify sub-optimal subset ;
    if $\mathcal{C}_i = \emptyset$ then
        continue                              // Skip if no sub-optimal properties
    if not all dir$[p]$ in $\mathcal{C}_i$ are the same then
        continue                    // Require improvement in all sub-optimal ones
    NeedSwap $\leftarrow$ true if all dir$[p]$ in $\mathcal{C}_i$ are opposite of desired // Determine swap condition ;
    if NeedSwap then
        Swap $M_x \leftrightarrow M_y$          // Ensure correct direction of improvement ;
    $\mathcal{C}_s \leftarrow C \cap \mathcal{P}_{\texttt{s}}$                // Identify near-optimal subset ;
    Construct task $t = (M_x, M_y, \mathcal{C}_i, \mathcal{C}_s)$              // An optimization task ;
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$
return $\mathcal{T}$
```

2025). They provide a reliable, computationally efficient means to estimate property scores at scale, enabling the construction of high-quality datasets with broad coverage of chemical space.

While these predictors are not experimentally validated, they demonstrate strong alignment with experimentally measured values and are widely accepted as practical surrogates in virtual screening pipelines. Notably, experimentally validated measurements are severely limited for many key pharmacological properties. For instance, public datasets contain fewer than 2,000 experimentally measured BBBP values – orders of magnitude below what is needed to train large-scale deep learning models or instruction-tuned LLMs. Given these constraints, the use of empirical predictors is not only standard but necessary for enabling scalable dataset creation and evaluation.

Instruction Diversity and Generation: To avoid LLM overfitting to specific phrasings and to promote generalization to natural word variations in task formulation, we ensure that each optimization task is associated with a diverse set of instructions. Starting from a manually written seed prompt, we use GPT-4o (OpenAI, 2024) to generate several paraphrased variants that preserve the semantic intent while differing in structure and wording. From these, we select 30 semantically equivalent but syntactically diverse instructions per task to include in the training data.

To explicitly assess the models' ability to generalize to new instructions, we hold out one instruction per task as unseen during instruction-tuning. This unseen instruction is then used during evaluation to measure robustness to novel phrasings. This design allows us to evaluate not only task-level generalization but also linguistic flexibility in following diverse natural language instructions. All instructions used in training and testing are provided in Appendix C.

B.3 Details on IND Tasks

1. `BPQ` (BBBP, PlogP, QED): This task involves 7 diverse combinations of property-specific objectives across BBBP, PlogP, and QED – three properties central to CNS drug design. Each optimization task may involve improving one or more of these properties while maintaining

or improving the others. Optimizing 7 diverse multi-objective combinations of BBBP, PlogP, and QED simulates early-stage filtering of CNS-active hits.

2. ELQ (hERG, LIV, QED): Here, the focus is on toxicity-related properties and overall drug-likeness. hERG inhibition and liver toxicity are two major causes of clinical trial failures, while QED ensures retained drug-like features. A good optimizer must reduce toxicity signals while preserving beneficial characteristics, reflecting real-world needs in late-stage lead optimization, where safety issues are addressed without sacrificing potency.

3. ACEP (AMP, CARC, hERG, PlogP): This task consists of 15 optimization combinations focused on absorption and toxicity-related properties. Each task may require improving any subset of AMP (permeability), CARC (carcinogenicity), hERG (cardiotoxicity), or PlogP (lipophilicity), while stabilizing the rest. It captures the complex trade-offs typical in preclinical candidate refinement, where ADME and safety must be simultaneously addressed.

4. BDPQ (BBBP, DRD2, PlogP, QED): This combination includes 13 challenging optimization tasks for antipsychotic drug design. These require optimization for BBB penetration and DRD2 activity – two critical endpoints for efficacy – while maintaining lipophilicity and drug-likeness. It embodies a highly targeted CNS design task and is one of the most challenging due to strong interdependencies among all properties.

5. DHMQ (DRD2, HIA, MUT, QED): This combination involves optimization of 9 different multi-objective tasks to optimize a CNS drug target that must bind to DRD2 receptors while exhibiting high intestinal absorption and low mutagenicity. Each task selectively improves or maintains a subset of these properties. It simulates a realistic challenge in optimizing orally active CNS agents under ADMET and pharmacological constraints.

B.4 Details on OOD Tasks

1. CDE (CARC, DRD2, hERG): These tasks target CNS drug candidates, especially antipsychotics, requiring high DRD2 inhibition. However, many such drugs are known to block the hERG potassium channel, raising serious cardiotoxicity concerns. Additionally, reducing carcinogenicity is essential for long-term drug safety. Each task may involve increasing DRD2 inhibition while reducing or preserving carcinogenicity and cardiotoxicity. This mirrors real-world lead optimization, where enhancing efficacy must be carefully balanced against major safety liabilities.

2. ABMP (AMP, BBBP, MUT, PlogP): Tasks in this combination target oral CNS-targeted drug design. AMP and BBBP capture permeability at intestinal and blood-brain barriers, respectively, essential for drugs acting on the brain after oral administration. Mutagenicity must be minimized or maintained to prevent genotoxic effects, while plogP should be improved or maintained to balance lipophilicity, solubility, and synthetic accessibility. The task requires coordinated improvement of absorption and brain penetration while constraining safety and physicochemical properties, posing a nontrivial optimization challenge.

3. BCMQ (BBBP, CARC, MUT, QED): These tasks comprise 15 multi-objective combinations requiring improvements in BBB permeability while maintaining or minimizing toxicity (CARC, MUT) and retaining or improving drug-likeness (QED). Each task emphasizes safety-aware design for CNS-targeting molecules without degrading overall molecular quality.

4. BDEQ (BBBP, DRD2, hERG, QED): This combination consists of 11 diverse optimization objectives. High BBBP and DRD2 inhibition are necessary for efficacy, while low hERG inhibition is essential to avoid cardiotoxicity. QED must remain high to ensure overall molecular quality. This combination embodies the classic efficacy-safety trade-off, making it one of the most realistic multi-objective scenarios.

5. HLMPQ (HIA, LIV, MUT, PlogP, QED): This combination includes 21 broad-spectrum ADMET-focused multi-objective tasks aimed at orally administered drugs. Each task challenges the model to find precise modifications that jointly optimize oral bioavailability and structural quality while minimizing major toxicity risks – reflecting a realistic early-phase development setting.

C Diverse Instructions

Figure A1 presents the prompt template used for instruction-tuning. Each prompt has three parts: (1) '{general instruction}', (2) input source molecule and properties to adjust for the specific optimization task, and (3) target optimized molecule.

The '{general instruction}' will be replaced with one of 6 diverse task instructions, which are presented below. The first instruction is manually written, and is provided as the seed instruction to GPT-4o to generate 5 more differently phrased instructions. The last one is the hold-out instruction for inference. Below are 6 diverse instructions:

1. "Your task is to modify the given molecule to adjust specific molecular properties so that the resulting molecule satisfies the given target thresholds. Keep structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed in <SMILES> </SMILES> tags. The property values of the new molecule should meet or exceed the specified targets enclosed in <THRESHOLD> </THRESHOLD> tags."

2. "Adjust the molecular structure to ensure that each specified property reaches the corresponding threshold listed in <THRESHOLD> </THRESHOLD>. Minimize structural changes and try to maintain the core scaffold. Return the resulting molecule using <SMILES> </SMILES> tags."

3. "Alter the molecule to satisfy the provided property thresholds in <THRESHOLD> </THRESHOLD>. Preserve the core scaffold and make as few structural changes as possible. Output the SMILES of the new molecule, enclosed in <SMILES> </SMILES>."

4. "Update the given molecule so that the specified properties fall within acceptable ranges defined by the values in <THRESHOLD> </THRESHOLD>. Maintain as much of the original structure as possible. Output only the modified molecule enclosed in <SMILES> </SMILES> tags."

5. "Edit the molecular structure so that all required properties match or exceed the threshold values defined in <THRESHOLD> </THRESHOLD>. Try to retain the core scaffold. Output only the SMILES representation of the optimized molecule enclosed in <SMILES> </SMILES>."

6. "Modify the molecule to bring its properties to at least the levels defined in <THRESHOLD> </THRESHOLD>. Avoid excessive modifications and preserve the core scaffold. Output only the resulting molecule's SMILES wrapped in <SMILES> </SMILES>."

In the 2nd part of the prompt template, multiple properties to be adjusted are described via the task-specific '{adjust_i}' (Figure A1). Each '{adjust_i}' is randomly replaced with one of the following 5 adjustment templates for each sub-optimal property improvement:

1. "change property to be direction <THRESHOLD> value </THRESHOLD>",

2. "change the value of property to be direction <THRESHOLD> value </THRESHOLD>",

3. "change property aiming for direction <THRESHOLD> value </THRESHOLD>",

4. "change property so it is direction <THRESHOLD> value </THRESHOLD>",

5. "change property with a goal of direction <THRESHOLD> value </THRESHOLD>"

Thus, 6 diverse general instruction templates and 5 diverse adjustment templates together lead to 30 different templates for instruction tuning.

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. AMP: "membrane permeability", "Parallel Artificial Membrane Permeability (PAMPA)"

2. BBBP: "BBB permeability", "Blood-brain barrier permeability (BBBP)"

3. CARC: "carcinogenicity", "potential to disrupt cellular metabolic processes"

4. DRD2: "DRD2 inhibition", "inhibition probability of Dopamine receptor D2"'

5. "hERG": "hERG inhibition", "potential to block hERG channel",

```
[INST]
{general instruction}

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

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

Figure A1: Prompt template used for instruction-tuning GeLLM^{4}O-Cs

6. HIA: “Intestinal adsorption", “human intestinal adsorption ability"
7. "DILI": "liver injury risk", "potential to cause liver disease",
8. MUT: “Mutagenicity", “probability to induce genetic alterations (mutagenicity)"
9. PlogP: “Penalized octanol-water partition coefficient (penalized logP)", “Penalized logP which is logP penalized by synthetic accessibility score and number of large rings"
10. QED: “QED", “drug-likeness quantified by QED score"

D Details on Experimental Setup

D.1 GeLLM^{4}O-Cs

We develop specialist and generalist GeLLM^{4}O-Cs by instruction-tuning general-purpose LLMs on C-MuMOInstruct using specific and multiple property combinations, respectively. The generalist GeLLM^{4}O-C-P(N) refers to a generalist model that is trained on property combinations, each with up to N properties. For backbone models, we use Mistral-7B-Instruct-v0.3 (AI, 2023) and Llama3.1-8B-Instruct (Grattafiori et al., 2024), and apply parameter-efficient fine-tuning using LoRA (Hu et al., 2022) through the Huggingface Transformers framework (Wolf et al., 2020). All models are fine-tuned with a learning rate of 1×10^{-4}, and a cosine scheduler with 5% warm-up. Specialist models are trained with a batch size of 32 for 10 epochs; GeLLM^{4}O-C-P(N) models are trained with a batch size of 128 for 5 epochs when $N <= 4$, and for 1,800 steps when $N = 10$. The difference in training steps/epochs is to strike a balance between training cost and overfitting. LoRA is configured with rank 16, $\alpha = 16$, dropout rate of 0.05, and is applied to all projection layers and the language modeling head. We conduct 0-shot evaluation for all GeLLM^{4}O-Cs, where no in-context examples are provided. For each test molecule, we generate 20 candidate molecules using beam search decoding with a beam width of 20.

Upon applying LoRA, the number of trainable parameters vary from 42M for Mistral-7B-v0.3 to 44M for Llama3.1-8B-Instruct. Training time on a single NVIDIA A100 GPU (40 GB) ranges from 1 hour for specialist models to 8–20 hours for generalist models, depending on the total number of tasks and molecule pairs – going up to 28K tasks and 1M pairs for GeLLM^{4}O-C-P(N) with N=10. The entire training consumed approximately 150 GPU hours.

D.2 Baselines

In this section, we detailed the baselines selected for our comparison. Table A1 lists the sources and licenses of all the source datasets and models (i.e., artifacts) used in this work. We ensured that all artifacts were utilized in accordance with the usage guidelines specified by their original authors or licensors. For the models we developed, we have considered relevant ethical implications, which are discussed in Section 10.

General-purpose LLMs: We benchmark 4 publicly available general-purpose LLMs, including 2 open-weights LLMs: Mistral-7B Instruct-v0.3 (AI, 2023), Llama-3.1 8B-Instruct (Touvron et al., 2023), and 2 closed-weights LLMs: Claude-3.5, and GPT-4o to assess their performance in molecule optimization tasks. For open-weights LLMs, we utilize their official HuggingFace checkpoints, while for closed-weights ones, we access the checkpoints via their official APIs.

We perform 0-shot and 1-shot inference (i.e., with 0 and 1 in-context examples, respectively) using the prompt templates, detailed in Appendix E.1. While few-shot prompting can improve performance, we selected 1-shot as a practical trade-off to control inference cost, especially for closedsourced API-based models. Moreover, we found negligible performance improvement using 5-shots in our preliminary experiments. We generate up to 20 molecules per input molecule using the same generation strategy for open-source LLMs as in GeLLM^{4}O-Cs. Since Claude and GPT do not support the beam-search decoding strategy or any customized strategy for multiple sequence generations, we generate only one molecule per input prompt.

Foundational LLMs for Chemistry: We adopt LlaSMol$_{\texttt{Mistral}}$, the Mistral-7B variant of LlaSMol, as the foundational LLM for chemistry due to its strong performance across diverse molecular tasks. In comparison to other instruction-tuned LLMs for chemistry, such as ChemDFM (Zhao et al., 2025), MolInst (Fang et al., 2024) and ChemLLM (Zhang et al., 2024), LlaSMol$_{\texttt{Mistral}}$ consistently achieves state-of-the-art results. For evaluation, we adopt 0-shot inference. Our preliminary experiments indicated that incorporating in-context examples did not lead to consistent improvements, rather impacted performance. Furthermore, we employ a simplified prompt format (as shown in Appendix E.2) after observing that LlaSMol struggles to follow more complex and structured instruction formats. For ChemDFM, we use 0-shot inference using the same prompt template and generation configuration as of general-purpose LLMs.

Non-LLM Domain-expert Methods: Existing non-LLM methods(Fu et al., 2021; Sun et al., 2022; Angelo et al., 2023; Kim et al., 2024) rely on genetic algorithms or reinforcement learning. These methods typically require carefully curated fitness or reward functions to balance multiple properties. Such functions are often difficult to design and require significant domain expertise, limiting their flexibility and generalizability.

Furthermore, these methods follow a fundamentally different experimental setting: given an initial pool of candidates, these methods iteratively modify molecules based on oracle feedback. This often leads to generating molecules with entirely new scaffolds. In contrast, our setting closely aligns with lead optimization in drug discovery, where the goal is to minimally modify an input molecule while preserving its core scaffold.

D.3 Evaluation Metrics

We adopt multiple evaluation metrics to comprehensively assess model performance. The metrics are defined as follows:

1. **Success Rate (SR):** SR denotes the proportion of test cases where at least one of the 20 generated candidate molecules satisfies all specified property objectives – i.e., improving all sub-optimal properties while preserving all near-optimal ones. When multiple candidates are optimized, the molecule exhibiting the highest cumulative improvement is selected for evaluation. A higher SR reflects the model's effectiveness in achieving task-specific optimization goals.

2. **Strict Success Rate (SR$_\Theta$):** SR$_\Theta$– a stricter variant of SR – measures the proportion of test cases where at least one generated molecule not only improves all sub-optimal properties but also brings each of them above the pharmaceutically relevant threshold Θ_p, while still preserving all near-optimal properties within their respective Δ_p bounds. This metric reflects whether the model can generate molecules with desirable properties as specified.

3. **Validity (Val):** Validity refers to the percentage of test instances for which at least one of the generated molecules is chemically valid, determined via successful parsing by RDKit. High Valensures the model's ability to generate syntactically correct and chemically valid structures.

4. **Similarity (Sim):** Sim measures the average Tanimoto similarity between optimized and input molecules based on binary Morgan fingerprints (with radius of 2 and dimension of 2048). Higher Sim indicates better preservation of the similarity constraint – a key requirement in lead optimization, where maintaining the core molecular scaffold is essential.

5. **Novelty (Nov):** Novelty quantifies the fraction of optimized molecules that are not present in the training set. This indicates the model's ability to generate novel and previously unseen drug candidates, crucial for exploration in drug discovery pipelines.

6. **Synthetic Accessibility Score (SAS):** SAS evaluates how easy a molecule is to synthesize, with scores ranging from 1 (easily synthesizable) to 10 (difficult to synthesize) (Ertl and Schuffenhauer, 2009a). Lower scores indicate simpler, more synthesizable molecules.

7. **Relative Improvement** (RI)**:** RI is computed as the average relative gain in each sub-optimal property compared to the input molecule. This metric reflects the magnitude of property-level improvements achieved by the model. Formally, for a task improving $\mathcal{P}_{\mathtt{i}}$ properties, RI is computed as the average of relative change ($\mathtt{RI}_p$) in each property $p \in \mathcal{P}_{\mathtt{i}}$ as:

$$\mathtt{RI} = \frac{\sum_{p \in \mathcal{P}_{\mathtt{i}}} \mathtt{RI}_p}{|\mathcal{P}_{\mathtt{i}}|},$$

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

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

where $\mathbb{D}[p]$ is an indicator function denoting whether higher scores of p is desirable, $p(M_x)$ and $p(M_y)$ denote the score of property p in the input molecule M_x and generated molecule M_y, respectively.

8. **Average Property Score** (APS)**:** APS is computed as the average property score for each molecular property across all successfully optimized molecules. Higher or lower APS, depending on the desired direction for each property, indicates that the model consistently generates better molecules with property scores aligned with pharmaceutical objectives.

E Prompt Templates

The prompt templates for general-purpose LLMs and for LlaSMol are provided below.

E.1 Prompt Template for General-purpose LLMs

We use a structured and detailed prompt template with a system prompt, task instruction, and in-context examples for few-shot prompting. Figure A2 shows an example.

E.2 Prompt Template for LlaSMol

Unlike general-purpose language models, LlaSMol was instruction-tuned on a range of chemistry-specific tasks using a dedicated prompt structure. In our preliminary experiments, we found that applying the general-purpose prompt format led to suboptimal performance, as LlaSMol often failed to interpret the task correctly. To address this, we adopted a simplified prompt format that omits the system message and does not explicitly separate the instruction, input, and expected output. Additionally, we restrict our evaluation of LlaSMol to 0-shot inference only. Figure A3 illustrates the simplified prompt used for the same task as above.

F Case Studies

F.1 Case from ACEP

Figure A4a and Figure A4b show optimization examples generated by $\mathtt{GeLLM^4O\text{-}C\text{-}P(10)_{Mistral}}$ and $\mathtt{LlaSMol_{Mistral}}$ on the IND task ACEP. The hit molecule features a central urea scaffold with a carboxamide and a morpholine ring. The goal is to improve AMP and PlogP while maintaining CARC and hERG.

$\mathtt{GeLLM^4O\text{-}C\text{-}P(10)_{Mistral}}$ accomplishes this by replacing the morpholine with a para-chlorophenyl group (Figure A4a). This modification eliminates a polar heterocycle and introduces a planar, lipophilic aromatic ring bearing a chlorine atom. This leads to notable improvements in AMP (+0.29) and PlogP (+0.85), while CARC and hERG remain within acceptable ranges. The increased hydrophobicity introduced by the chlorinated aromatic ring contributes to a higher PlogP, as aromatic chlorides are known to enhance lipophilicity due to both the non-polar nature of the phenyl group and the electron-withdrawing effect of chlorine (Hansch et al., 1995). The rigid aromatic system may reduce the molecule's conformational flexibility, which in turn lowers conformational entropy. This structural constraint can limit the number of unintended binding interactions, thereby reducing the likelihood of off-target liabilities (Meanwell, 2011b, 2016)

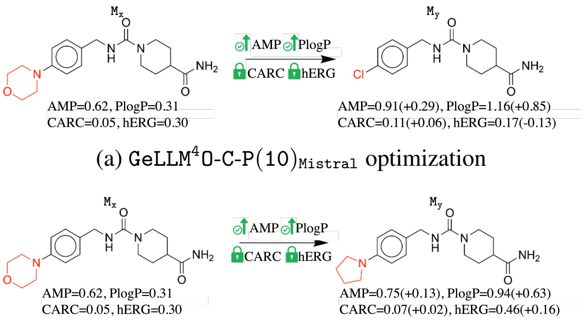
$\mathtt{LlaSMol_{Mistral}}$'s modification replaces the morpholine with a pyrrolidine ring. This change maintains a basic nitrogen atom but removes the oxygen, slightly reducing polarity compared to morpholine. Although this approach achieves a moderate PlogP improvement (+0.63), it shows a concerning increase in hERG liability (+0.16). The pyrrolidine ring, while structurally similar to morpholine (Figure A4b), introduces greater basicity and conformational flexibility. These properties are known risk factors for hERG channel binding in medicinal chemistry, explaining the less favorable safety profile (Cavalli et al., 2002).

```
<<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> O=C(Cc1cccc([N+](=O)[O-])c1)NC1CCN(Cc2ccccc2)CC1 </SMILES>
%%% Adjust: increase DRD2 inhibition with a goal of at least <THRESHOLD> 0.54 </
THRESHOLD>, decrease Mutagenicity with a goal of at most <THRESHOLD> 0.1 </
THRESHOLD> and increase QED aiming for at least <THRESHOLD> 0.89 </THRESHOLD>
while keeping Intestinal adsorption unchanged.
%%% Response: <SMILES> O=C(Cc1ccc(O)cc1)NC1CCN(Cc2ccccc2)CC1 </SMILES>

Task:
%%% Input : <SMILES> C#Cc1ccc(C2CC3CCC(C2C(=O)OC)N3C)cc1 </SMILES>
%%% Adjust: decrease Mutagenicity with a goal of at most <THRESHOLD> 0.2 </
THRESHOLD>, increase QED with a goal of at least <THRESHOLD> 0.8 </THRESHOLD> and
increase the value of DRD2 inhibition to be at least <THRESHOLD> 0.2 </THRESHOLD>
while keeping Intestinal adsorption unchanged.
[/INST]

%%% Response:
```

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

```
Modify the molecule <SMILES> C#Cc1ccc(C2CC3CCC(C2C(=O)OC)N3C)cc1 <SMILES> to
decrease the value of Mutagenicity to be at most <THRESHOLD> 0.2 </THRESHOLD>,
increase QED to be at least <THRESHOLD> 0.8 </THRESHOLD> and increase DRD2
inhibition to be at least <THRESHOLD> 0.2 </THRESHOLD> while keeping Intestinal
adsorption unchanged.
%%% Response:
```

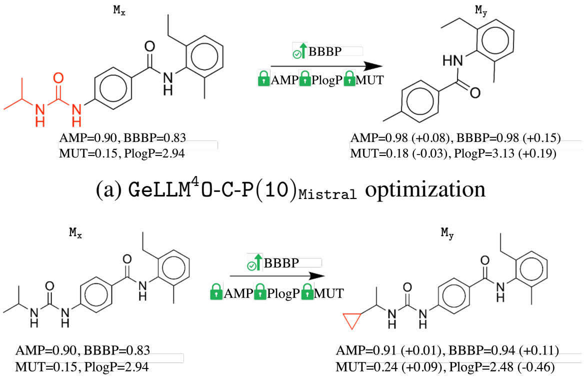
Figure A3: An example of a prompt used for LlaSMol


M_x
M_y
AMP PlogP
CARC hERG
AMP=0.62, PlogP=0.31
CARC=0.05, hERG=0.30
AMP=0.91(+0.29), PlogP=1.16(+0.85)
CARC=0.11(+0.06), hERG=0.17(-0.13)
AMP=0.62, PlogP=0.31
CARC=0.05, hERG=0.30
AMP=0.75(+0.13), PlogP=0.94(+0.63)
CARC=0.07(+0.02), hERG=0.46(+0.16)

(a) GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ optimization

(b) LlaSMol$_{\text{Mistral}}$ optimization

Figure A4: An example from ACEP. Modifications are highlighted in red.


M_x
M_y
BBBP
AMP PlogP MUT
AMP=0.90, BBBP=0.83
MUT=0.15, PlogP=2.94
AMP=0.98 (+0.08), BBBP=0.98 (+0.15)
MUT=0.18 (-0.03), PlogP=3.13 (+0.19)
AMP=0.90, BBBP=0.83
MUT=0.15, PlogP=2.94
AMP=0.91 (+0.01), BBBP=0.94 (+0.11)
MUT=0.24 (+0.09), PlogP=2.48 (-0.46)

(a) GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ optimization

(b) LlaSMol$_{\text{Mistral}}$ optimization

Figure A5: An example from ABMP. Modifications are highlighted in red.

F.2 Case from ABMP

Figure A5a and Figure A5b present optimization examples produced by GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ and LlaSMol$_{\text{Mistral}}$ on the OOD task ABMP. The hit molecule is a symmetric tri-amide structure, composed of three carbonyl linkers connecting aromatic and aliphatic moieties. The goal is to im-

Table A1: Licenses and Sources of Artifacts

Artifact	Source	License Type	Accessibility
Modof	https://github.com/ziqi92/Modof	PolyForm Noncommercial License 1.0.0	Open Source
LlaSMol$_{\text{Mistral}}$	https://huggingface.co/datasets/osunlp/SMolInstruct	Creative Commons Attribution 4.0	Checkpoint
ChemDFM$_{\text{Llama}}$	https://huggingface.co/OpenDFM/ChemDFM-v1.5-8B	GNU Affero General Public License v3.0	Checkpoint
Claude 3.5 (Sonnet)	https://docs.anthropic.com/claude/reference/getting-started-with-the-api	Proprietary	API
GPT-4o	https://openai.com/api/	Proprietary	API
Llama-3.1 8B-Instruct	https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct	Llama 3.1 Community	Checkpoint
Mistral-7B-Instruct-v0.3	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3	Apache license 2.0	Checkpoint

prove BBBP, while keeping AMP, MUT, and PlogP stable.

GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ introduces a substantial simplification by collapsing the tri-amide backbone into a more compact structure containing a single central amide and two substituted aromatic rings (Figure A5a). This transformation removes several polar functional groups and incorporates lipophilic features such as methyl and aryl substitutions. These changes are well-aligned with medicinal chemistry strategies for enhancing membrane permeability – primarily through increased lipophilicity and reduced polarity (Meanwell, 2011a; Leeson and Springthorpe, 2007). As a result, GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ achieves a favorable outcome, yielding a significant improvement in BBBP (+0.15), along with a modest increase in PlogP (+0.19), while keeping AMP and MUT values stable.

In contrast, LlaSMol$_{\text{Mistral}}$ applies a conservative modification by retaining the tri-amide scaffold and appending an isopropyl group to the left-hand side of the molecule (Figure A5b). This change preserves the molecule's original polarity and structural complexity, while introducing additional steric bulk. Crucially, it fails to reduce polarity or increase hydrophobicity – both essential for maintaining or improving PlogP (Ertl and Schuffenhauer, 2009b). As a result, despite a small gain in BBBP (+0.11), the model suffers a substantial drop in PlogP (–0.46) and an increase in toxicity (MUT), indicating an unfavorable optimization outcome.

G Complete Experimental Results

G.1 IND Evaluation

Tables A3, A4, A5, A6 and A7 presents the performance comparison of GeLLM^{4}O-Cs with general-purpose LLMs and LlaSMol$_{\text{Mistral}}$ under all evaluation metrics for each IND task.

Table A2 presents the overall performance comparison of GeLLM^{4}O-Cs with all baselines under the strict success criteria. This requires each sub-optimal property to exceed its predefined pharmaceutically relevant threshold, Θ_p, in the optimized molecule. We use Θ_p to reflect realistic drug design objectives, where each property is expected to reach a clinically meaningful level. However, this is a highly challenging setting, particularly because our evaluation involves only a single-step molecule modification. Starting molecules may be significantly sub-optimal, and a single structural change may not be sufficient to reach such high thresholds. This explains the significantly lower success rates for all models compared to the looser success criteria in Table 3.

G.2 OOD Evaluation

Tables A8, A9, A10, A11 and A12 presents the performance comparison of GeLLM^{4}O-Cs with general-purpose LLMs and LlaSMol$_{\text{Mistral}}$ under all evaluation metrics for each OOD task.

G.3 Ablation Studies

To better understand what drives the performance gains of our instruction-tuned models, we conducted two key ablations: (1) the impact of the number of properties seen during instruction tuning of generalist GeLLM^{4}O-Cs, and (2) the impact of instruction template diversity.

Properties in Tuning Generalist GeLLM^{4}O-Cs: We compare multiple variants of generalist GeLLM^{4}O-Cs with the specialist GeLLM^{4}O-C across 3 IND tasks (BPQ, BDPQ, and DHMQ): **(1)** Generalist GeLLM^{4}O-C-P(N)s trained on the power set

Table A2: Overall Performance in IND Tasks with stricter success criteria

Model	BPQ			ELQ			ACEP			BDPQ			DHMQ		
	SR$_\theta$$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$_\theta$$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$_\theta$$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$_\theta$$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$	SR$_\theta$$^\uparrow$	Sim$^\uparrow$	RI$^\uparrow$
General-purpose LLMs															
Mistral (0-shot)	3.40	0.71	1.60	3.40	**0.70**	0.38	2.80	0.70	0.88	0.00	-	-	0.00	-	-
Llama (0-shot)	3.80	0.69	0.39	2.20	0.69	0.27	1.00	0.71	0.53	0.00	-	-	0.20	0.75	3.00
Claude-3.5 (0-shot)	4.40	0.65	0.56	3.00	0.63	0.40	1.60	0.60	0.72	0.00	-	-	0.00	-	-
GPT-4o (0-shot)	1.60	**0.73**	0.48	1.40	0.67	0.33	1.60	0.72	0.34	0.00	-	-	0.40	0.71	2.51
Mistral (1-shot)	14.20	0.53	1.45	16.20	0.57	0.49	10.20	0.54	1.31	3.40	0.32	18.68	3.40	0.39	3.87
Llama (1-shot)	6.40	0.63	0.62	4.80	0.61	0.39	3.00	0.63	0.47	0.40	0.15	18.71	2.20	0.28	**14.00**
Claude-3.5 (1-shot)	9.20	0.59	0.95	3.20	0.63	0.42	3.60	**0.73**	0.72	0.60	0.38	4.16	0.40	0.69	2.73
GPT-4o (1-shot)	2.60	0.70	0.45	2.00	0.67	0.28	1.20	**0.73**	0.25	0.00	-	-	1.00	0.71	2.72
Foundational LLMs for Chemistry															
LlaSMol-M	14.80	0.61	0.88	17.60	0.60	0.48	10.80	0.62	0.67	0.60	**0.68**	9.42	1.40	0.70	4.12
ChemDFM$_{\text{Llama}}$	3.20	0.63	0.33	3.00	0.65	0.38	1.40	0.69	0.40	0.20	0.55	0.78	0.60	**0.81**	5.44
Specialist LLMs															
GeLLM^{4}O-C-N$_{\text{Mistral}}$	25.40	0.51	2.57	28.80	0.51	0.56	28.00	0.50	**4.00**	**9.40**	0.35	13.24	6.40	0.52	9.92
GeLLM^{4}O-C-N$_{\text{Llama}}$	29.60	0.53	2.06	31.40	0.50	**0.58**	31.40	0.50	3.14	4.60	0.48	16.89	4.20	0.65	10.68
Impv-Spec (%)	100.0	-13.1	134.1	78.4	-16.7	20.8	190.7	-19.4	368.7	176.5	9.4	-29.1	88.2	33.3	156.3
Generalist LLMs															
GeLLM^{4}O-C-P(N)$_{\text{Mistral}}$	27.60	0.59	2.43	23.40	0.62	0.51	31.20	0.57	3.42	5.40	0.55	11.30	**9.00**	0.54	11.53
GeLLM^{4}O-C-P(N)$_{\text{Llama}}$	30.60	0.57	2.15	25.60	0.60	0.51	**34.40**	0.55	2.77	6.40	0.50	19.46	6.80	0.60	13.35
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**32.60**	0.59	2.32	**32.00**	0.57	0.55	23.40	0.58	1.88	3.80	0.59	13.26	4.80	0.64	11.14
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	32.40	0.54	**2.59**	27.60	0.56	0.54	25.20	0.56	3.11	5.00	0.51	**22.70**	5.40	0.56	13.70
Impv-Gen (%)	120.3	-3.3	163.6	81.8	-5.0	14.6	218.5	-11.3	313.4	88.2	56.2	4.2	164.7	38.5	197.9

$^\uparrow$ and $^\downarrow$ indicate whether a higher or lower value of the metric is preferred, respectively. For each task, we underline the best baseline performance and highlight in **bold** the best performing model for each metric. Impv-Spec and Impv-Gen represent the relative percentage improvement from the best specialist LLM and best generalist LLM over the best baseline, respectively. The best model in each group is selected based on SR for each task.

Table A3: Overall Performance on BPQ

Model	SR$^\uparrow$	Val$^\uparrow$	Sim$^\uparrow$	Nov$^\uparrow$	SAS$^\downarrow$	RI$^\uparrow$	APS		
							BBBP$^\uparrow$	PlogP$^\uparrow$	QED$^\uparrow$
General-purpose LLMs									
Mistral (0-shot)	28.80	85.80	**0.75**	**100.00**	2.87	1.24	0.92	0.41	0.77
Llama (0-shot)	33.60	99.00	0.70	**100.00**	2.86	0.78	0.92	0.65	0.76
Claude-3.5 (0-shot)	51.80	96.80	0.68	99.61	2.75	0.89	0.91	0.70	0.75
GPT-4o (0-shot)	30.20	88.00	0.72	**100.00**	2.70	0.55	0.90	0.65	0.76
Mistral (1-shot)	72.80	99.20	0.63	97.53	2.58	1.26	0.91	1.07	0.77
Llama (1-shot)	49.60	**100.00**	0.68	99.19	2.71	0.95	0.91	0.89	0.75
Claude-3.5 (1-shot)	61.80	96.60	0.65	**100.00**	2.68	1.31	**0.93**	0.90	0.77
GPT-4o (1-shot)	28.60	86.20	0.74	**100.00**	2.76	0.77	0.90	0.70	0.76
Foundational LLMs for Chemistry									
LlaSMol$_{\text{Mistral}}$	78.20	**100.00**	0.64	99.74	2.65	0.92	0.91	0.87	0.77
ChemDFM$_{\text{Llama}}$	27.00	92.00	0.66	99.26	2.82	0.65	**0.93**	0.68	0.77
Specialist LLMs									
GeLLM^{4}O-C-3$_{\text{Mistral}}$	71.00	98.40	0.57	98.87	2.45	2.59	**0.93**	1.51	**0.79**
GeLLM^{4}O-C-3$_{\text{Llama}}$	84.20	**100.00**	0.58	99.05	2.46	2.09	0.92	1.44	**0.79**
Impv-Spec	7.7	0.0	-9.4	-0.7	7.2	127.2	1.1	65.5	2.6
Generalist LLMs									
GeLLM^{4}O-C-P(3)$_{\text{Mistral}}$	84.80	**100.00**	0.63	99.06	2.46	2.64	0.92	1.47	0.78
GeLLM^{4}O-C-P(3)$_{\text{Llama}}$	88.80	**100.00**	0.62	99.10	**2.38**	2.16	0.92	1.48	**0.79**
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**89.40**	99.00	0.62	98.43	2.49	2.30	**0.93**	1.39	**0.79**
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	79.40	88.80	0.57	97.48	2.42	**2.67**	**0.93**	**1.56**	**0.79**
Impv-Gen	14.3	-1.0	-3.1	-1.3	6.0	150.0	2.2	59.8	2.6

$^\uparrow$ and $^\downarrow$ indicate whether a higher or lower value of the metric is preferred, respectively. For each task, we underline the best baseline performance and highlight in **bold** the best performing model for each metric. Impv-Spec and Impv-Gen represent the relative percentage improvement from the best specialist LLM and best generalist LLM over the best baseline, respectively. The best model in each group is selected based on SR for this task.

Table A4: Overall Performance on ELQ

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS		
							hERG$^{\downarrow}$	LIV$^{\downarrow}$	QED$^{\uparrow}$
General-purpose LLMs									
Mistral (0-shot)	21.60	89.20	0.72	**100.00**	2.82	0.16	0.37	0.55	0.77
Llama (0-shot)	16.60	97.40	**0.74**	**100.00**	2.90	0.10	0.44	0.56	0.80
Claude-3.5 (0-shot)	20.00	96.40	0.64	**100.00**	2.67	0.20	0.41	0.60	0.76
GPT-4o (0-shot)	16.60	90.80	0.72	**100.00**	2.83	0.10	0.39	0.53	0.74
Mistral (1-shot)	74.80	99.80	0.59	94.92	2.77	0.28	0.38	0.55	0.78
Llama (1-shot)	36.80	99.40	0.68	97.83	2.90	0.15	0.45	0.56	0.77
Claude-3.5 (1-shot)	29.20	97.60	0.63	**100.00**	2.73	0.21	0.48	0.58	0.76
GPT-4o (1-shot)	19.60	90.00	0.72	**100.00**	2.85	0.12	0.46	0.53	0.76
Foundational LLMs for Chemistry									
LlaSMol$_{\text{Mistral}}$	81.40	99.80	0.62	99.26	2.71	0.28	0.38	0.56	0.77
ChemDFM$_{\text{Llama}}$	15.00	91.20	0.68	**100.00**	2.91	0.19	0.38	0.52	0.79
Specialist LLMs									
GeLLM^{4}O-C-3$_{\text{Mistral}}$	81.80	99.40	0.55	99.27	2.85	0.39	0.32	**0.46**	0.79
GeLLM^{4}O-C-3$_{\text{Llama}}$	85.40	**100.00**	0.53	99.53	2.87	**0.41**	**0.29**	**0.46**	0.79
Impv-Spec	4.9	0.2	-14.5	0.3	-5.9	46.4	23.7	17.9	2.6
Generalist LLMs									
GeLLM^{4}O-C-P(3)$_{\text{Mistral}}$	83.20	99.80	0.63	98.80	2.64	0.33	0.33	0.53	0.78
GeLLM^{4}O-C-P(3)$_{\text{Llama}}$	**90.80**	**100.00**	0.63	98.90	2.60	0.34	0.33	0.52	0.80
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	88.40	99.80	0.59	99.55	2.64	**0.41**	**0.29**	0.50	**0.81**
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	79.00	90.60	0.56	99.49	**2.58**	**0.41**	0.30	0.48	**0.81**
Impv-Gen	11.5	0.2	1.6	-0.4	4.1	21.4	13.2	7.1	3.9

The metrics, notations, and formatting have the same meanings as those in Table A3.

Table A5: Overall Performance on ACEP

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS			
							AMP$^{\uparrow}$	CARC$^{\downarrow}$	hERG$^{\downarrow}$	PlogP$^{\uparrow}$
General-purpose LLMs										
Mistral (0-shot)	26.20	87.20	0.75	**100.00**	2.77	1.10	0.90	0.18	0.38	0.70
Llama (0-shot)	17.20	98.00	0.74	**100.00**	2.74	0.69	0.90	0.20	0.47	0.76
Claude-3.5 (0-shot)	29.60	96.20	0.71	**100.00**	2.78	0.69	0.91	0.17	0.38	0.64
GPT-4o (0-shot)	22.20	91.40	0.74	99.10	2.77	0.52	0.90	0.17	0.36	0.54
Mistral (1-shot)	63.80	99.80	0.64	95.92	2.56	1.03	0.92	0.18	0.43	0.92
Llama (1-shot)	40.20	99.00	0.70	98.51	2.64	1.12	0.92	0.20	0.46	0.87
Claude-3.5 (1-shot)	32.60	96.60	0.71	**100.00**	2.74	1.24	0.94	0.16	0.42	0.60
GPT-4o (1-shot)	23.00	88.80	**0.76**	**100.00**	2.79	1.09	0.93	0.17	0.40	0.63
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	68.60	**100.00**	0.66	99.71	2.65	1.00	0.93	0.17	0.43	0.90
ChemDFM$_{\text{Llama}}$	22.00	93.00	0.72	**100.00**	2.85	1.03	0.93	0.16	0.44	0.84
Specialist LLMs										
GeLLM^{4}O-C-4$_{\text{Mistral}}$	85.60	**100.00**	0.54	99.53	2.39	**2.46**	0.95	0.14	**0.33**	1.24
GeLLM^{4}O-C-4$_{\text{Llama}}$	88.00	99.80	0.54	99.55	2.38	2.24	0.95	0.14	0.34	1.25
Impv-Spec	28.3	-0.2	-18.2	-0.2	10.2	124.0	2.2	17.6	20.9	38.9
Generalist LLMs										
GeLLM^{4}O-C-P(4)$_{\text{Mistral}}$	86.60	**100.00**	0.60	98.61	2.38	2.34	**0.96**	0.15	0.36	1.25
GeLLM^{4}O-C-P(4)$_{\text{Llama}}$	**92.80**	99.80	0.58	98.92	**2.34**	2.22	0.95	0.15	0.35	1.26
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	74.60	**100.00**	0.61	99.20	2.44	1.92	0.95	**0.13**	0.35	1.11
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	72.60	93.60	0.57	98.62	2.38	2.27	**0.96**	0.15	0.38	**1.33**
Impv-Gen	35.3	-0.2	-12.1	-0.8	11.7	122.0	2.2	11.8	18.6	40.0

The metrics, notations, and formatting have the same meanings as those in Table A3.

of property combinations for only the N properties involved in each IND task; **(2)** Generalist GeLLM^{4}O-C-P(6)s trained on the power set of 6 properties (BBBP, DRD2, HIA, mutagenicity, PlogP, and QED) common across the 3 tasks; and **(3)** Generalist GeLLM^{4}O-C-P(10)s trained on all property combinations involving all 10 properties across all C-MuMO tasks.

Table A13 presents the performance of these models tuned on the Mistral checkpoint. Overall, GeLLM^{4}O-C-P(6) consistently outperforms GeLLM^{4}O-C-P(N) and GeLLM^{4}O-C-P(10), and per-

Table A6: Overall Performance on BDPQ

Model	SR$^\uparrow$	Val$^\uparrow$	Sim$^\uparrow$	Nov$^\uparrow$	SAS$^\downarrow$	RI$^\uparrow$	APS			
							BBBP$^\uparrow$	DRD2$^\uparrow$	PlogP$^\uparrow$	QED$^\uparrow$
General-purpose LLMs										
Mistral (0-shot)	2.40	75.60	**0.72**	**100.00**	2.83	0.49	**0.96**	0.09	0.66	0.82
Llama (0-shot)	8.80	97.00	**0.72**	**100.00**	3.24	1.67	**0.96**	0.06	0.03	0.79
Claude-3.5 (0-shot)	11.20	96.80	0.67	**100.00**	2.78	1.80	0.93	0.09	0.60	0.78
GPT-4o (0-shot)	4.20	84.80	**0.72**	**100.00**	2.92	3.98	0.93	0.07	0.51	0.82
Mistral (1-shot)	21.60	99.20	0.59	92.59	2.65	4.76	0.94	0.18	0.94	0.80
Llama (1-shot)	14.40	99.40	0.63	91.67	3.01	2.65	0.94	0.11	0.63	0.78
Claude-3.5 (1-shot)	15.60	95.20	0.58	**100.00**	2.66	3.99	0.94	0.11	1.26	0.80
GPT-4o (1-shot)	5.60	87.20	0.68	**100.00**	2.65	3.47	0.95	0.09	1.09	**0.85**
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	22.60	**100.00**	0.68	**100.00**	2.85	2.22	0.93	0.09	0.63	0.78
ChemDFM$_{\text{Llama}}$	6.20	93.00	0.67	**100.00**	2.85	3.51	0.92	0.07	0.64	0.80
Specialist LLMs										
GeLLM^{4}O-C-4$_{\text{Mistral}}$	**56.60**	**100.00**	0.50	97.88	**2.45**	5.48	0.95	**0.22**	1.25	0.79
GeLLM^{4}O-C-4$_{\text{Llama}}$	43.60	99.80	0.58	99.08	2.52	4.85	0.95	0.16	1.14	0.79
Impv-Spec	150.4	0.0	-26.5	-2.1	14.0	146.8	2.2	144.4	98.4	1.3
Generalist LLMs										
GeLLM^{4}O-C-P(4)$_{\text{Mistral}}$	50.60	**100.00**	0.58	99.21	2.51	4.93	0.95	0.17	1.23	0.79
GeLLM^{4}O-C-P(4)$_{\text{Llama}}$	51.00	**100.00**	0.58	98.43	2.49	5.40	0.95	0.17	1.19	0.78
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	48.40	99.40	0.58	99.17	2.55	5.05	0.95	0.16	1.22	0.79
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	42.60	88.60	0.55	98.59	2.47	**5.89**	0.94	0.17	**1.37**	0.79
Impv-Gen	125.7	0.0	-14.7	-1.6	12.6	143.2	2.2	88.9	88.9	0.0

The metrics, notations, and formatting have the same meanings as those in Table A3.

Table A7: Overall Performance on DHMQ

Model	SR$^\uparrow$	Val$^\uparrow$	Sim$^\uparrow$	Nov$^\uparrow$	SAS$^\downarrow$	RI$^\uparrow$	APS			
							DRD2$^\uparrow$	HIA$^\uparrow$	MUT$^\downarrow$	QED$^\uparrow$
General-purpose LLMs										
Mistral (0-shot)	4.80	86.80	0.71	**100.00**	2.88	0.76	0.05	**1.00**	0.29	0.80
Llama (0-shot)	6.00	97.40	**0.73**	**100.00**	3.09	1.35	0.06	**1.00**	0.28	0.79
Claude-3.5 (0-shot)	5.20	95.20	0.63	**100.00**	2.73	1.84	0.10	**1.00**	0.20	0.75
GPT-4o (0-shot)	5.80	87.80	0.72	**100.00**	2.89	0.88	0.07	**1.00**	0.22	**0.82**
Mistral (1-shot)	25.60	99.80	0.55	86.72	2.89	1.89	**0.18**	**1.00**	0.21	0.78
Llama (1-shot)	13.80	99.40	0.56	85.51	3.06	**3.39**	**0.18**	**1.00**	0.24	0.79
Claude-3.5 (1-shot)	8.40	95.20	0.65	**100.00**	2.77	1.38	0.12	**1.00**	0.21	0.78
GPT-4o (1-shot)	5.60	87.40	0.71	**100.00**	2.78	1.22	0.10	**1.00**	0.22	0.81
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	24.80	**100.00**	0.62	**100.00**	2.93	1.44	0.08	0.99	0.20	0.78
ChemDFM$_{\text{Llama}}$	6.80	86.40	0.67	**100.00**	3.03	1.72	0.07	**1.00**	**0.17**	**0.82**
Specialist LLMs										
GeLLM^{4}O-C-4$_{\text{Mistral}}$	44.60	99.80	0.57	99.10	2.81	2.96	0.14	0.99	0.19	0.78
GeLLM^{4}O-C-4$_{\text{Llama}}$	35.40	**100.00**	0.65	**100.00**	2.73	2.63	0.12	0.99	0.20	0.79
Impv-Spec	74.2	0.0	3.6	14.3	2.8	56.6	-22.2	-1.0	9.5	0.0
Generalist LLMs										
GeLLM^{4}O-C-P(4)$_{\text{Mistral}}$	**53.40**	**100.00**	0.59	99.25	2.76	3.26	0.15	0.99	0.19	0.78
GeLLM^{4}O-C-P(4)$_{\text{Llama}}$	50.40	**100.00**	0.59	**100.00**	2.67	3.28	0.13	0.99	0.19	0.79
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	52.20	99.60	0.61	**100.00**	2.76	2.24	0.12	0.99	0.19	0.79
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	41.80	83.20	0.57	**100.00**	**2.65**	3.32	0.15	0.99	0.20	0.79
Impv-Gen	108.6	0.2	7.3	14.4	4.5	72.5	-16.7	-1.0	9.5	0.0

The metrics, notations, and formatting have the same meanings as those in Table A3.

forms on par or better than the specialist GeLLM^{4}O-C-N. While GeLLM^{4}O-C-P(6) benefits from limited exposure to fewer properties and thus fewer property trade-offs, the generalist model GeLLM^{4}O-C-P(10) learns to tackle significantly more diverse and conflicting property trade-offs. These additional trade-offs potentially make learning more challenging, especially for tasks requiring fine-grained control and with conflicting objectives. Nonetheless, GeLLM^{4}O-C-P(10) achieves highly competitive performance across all tasks – demonstrating its robustness and flexibility as a

Table A8: Overall Performance on CDE

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS		
							CARC$^{\downarrow}$	DRD2$^{\uparrow}$	hERG$^{\downarrow}$
General-purpose LLMs									
Mistral (0-shot)	3.00	86.00	0.73	**100.00**	3.13	1.33	0.15	**0.14**	0.65
Llama (0-shot)	6.80	96.60	0.68	**100.00**	3.32	0.77	0.20	0.06	0.57
Claude-3.5 (0-shot)	6.80	97.80	0.70	**100.00**	2.98	1.07	0.16	0.08	0.52
GPT-4o (0-shot)	3.80	89.80	**0.74**	**100.00**	3.01	1.56	0.15	0.05	**0.39**
Mistral (1-shot)	30.60	99.60	0.62	93.46	3.00	**1.66**	0.15	0.09	0.50
Llama (1-shot)	18.20	99.40	0.55	76.92	3.50	1.51	0.14	0.12	0.47
Claude-3.5 (1-shot)	8.40	98.40	0.66	**100.00**	2.91	1.09	0.12	0.08	0.47
GPT-4o (1-shot)	7.00	88.20	0.72	**100.00**	3.10	1.04	0.16	0.05	0.53
Foundational LLMs for Chemistry									
LlaSMol$_{\text{Mistral}}$	29.80	**99.80**	0.61	97.99	**2.79**	1.28	0.14	0.06	0.46
ChemDFM$_{\text{Llama}}$	8.20	90.60	0.64	**100.00**	3.16	0.84	0.17	0.08	0.53
Generalist LLMs									
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**39.80**	98.60	0.58	**100.00**	2.85	**1.66**	**0.11**	0.08	0.42
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	33.20	86.80	0.55	**100.00**	2.86	1.50	**0.11**	0.08	0.48
Impv-Gen	30.1	-1.0	-6.5	7.0	5.0	0.0	26.7	-11.1	16.0

The metrics, notations, and formatting have the same meanings as those in Table A3.

Table A9: Overall Performance on ABMP

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS			
							AMP$^{\uparrow}$	BBBP$^{\uparrow}$	MUT$^{\downarrow}$	PlogP$^{\uparrow}$
General-purpose LLMs										
Mistral (0-shot)	23.00	83.00	**0.77**	**100.00**	2.76	0.93	0.90	0.87	0.24	0.86
Llama (0-shot)	44.60	98.40	0.71	**100.00**	2.85	0.61	0.92	0.90	0.25	1.17
Claude-3.5 (0-shot)	43.60	96.20	0.70	**100.00**	2.73	0.80	0.95	0.89	0.24	0.81
GPT-4o (0-shot)	27.00	87.40	0.73	**100.00**	2.72	0.51	0.93	0.89	0.25	0.93
Mistral (1-shot)	73.20	99.60	0.64	94.81	2.62	1.09	0.93	0.90	0.23	1.10
Llama (1-shot)	60.80	99.60	0.70	99.01	2.76	0.83	0.92	0.89	0.24	1.02
Claude-3.5 (1-shot)	45.20	96.40	0.64	**100.00**	2.67	0.87	0.95	0.91	0.23	1.04
GPT-4o (1-shot)	34.40	87.80	0.74	**100.00**	2.73	0.65	0.93	0.89	0.28	1.03
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	72.40	**100.00**	0.67	**100.00**	2.75	0.78	0.94	0.89	0.24	0.93
ChemDFM$_{\text{Llama}}$	39.60	92.40	0.67	**100.00**	2.95	0.98	0.94	0.89	0.23	1.40
Generalist LLMs										
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**86.60**	99.40	0.63	98.85	2.48	1.68	0.95	**0.92**	0.20	1.63
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	79.60	89.60	0.58	98.99	**2.42**	**1.81**	**0.96**	0.91	**0.19**	**1.81**
Impv-Gen	18.3	-0.2	-1.6	4.3	5.3	54.1	2.2	2.2	13.0	48.2

The metrics, notations, and formatting have the same meanings as those in Table A3.

single unified model capable of handling diverse C-MuMO tasks without retraining.

Instruction Diversity: We further ablate the effect of instruction diversity during instruction tuning. We compare GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$ trained with either a single instruction template or a set of 30 diverse templates. Both models were evaluated using an unseen instruction. Table A14 presents the SR of these 2 variants when evaluated on 5 IND tasks. Clearly, the variant trained with diverse instructions significantly outperforms the other variant across all tasks. This suggests that instruction diversity during training improves robustness to varied phrasings and generalization to unseen instructions at inference.

G.4 Failure Analysis

To better understand the limitations of our instruction-tuned models, we perform a targeted failure analysis on challenging multi-property tasks. Specifically, we focus on the 4 property combinations with the lowest overall success rates – BDPQ, DHMQ, CDE, and BDEQ – across both IND and OOD tasks. For each of these combinations, we identify the most dominant multi-property objective (i.e., one with the largest test-set size) and analyze failures for that objective. We define two metrics to characterize failures: **(1)** Failure Rate (FR) of an objective as the average percentage of generated molecules that fail to satisfy at least one property constraint (either improvement or stability require-

Table A10: Overall Performance on BCMQ

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS			
							BBBP$^{\uparrow}$	CARC$^{\downarrow}$	MUT$^{\downarrow}$	QED$^{\uparrow}$
General-purpose LLMs										
Mistral (0-shot)	25.40	89.60	0.69	**100.00**	2.84	0.25	0.92	0.16	0.25	0.77
Llama (0-shot)	20.40	98.60	0.72	**100.00**	2.86	0.20	0.90	0.18	0.24	0.79
Claude-3.5 (0-shot)	30.00	96.00	0.64	**100.00**	2.66	0.26	0.91	0.16	0.22	0.77
GPT-4o (0-shot)	19.60	90.60	0.72	**100.00**	2.66	0.19	0.90	0.18	0.21	0.77
Mistral (1-shot)	63.80	99.60	0.60	93.10	2.61	0.31	0.90	0.16	0.20	0.78
Llama (1-shot)	41.60	99.80	0.67	95.67	2.78	0.23	0.91	0.17	0.23	0.77
Claude-3.5 (1-shot)	32.40	95.00	0.61	**100.00**	2.69	0.30	0.91	0.15	0.23	0.78
GPT-4o (1-shot)	23.40	86.40	**0.73**	**100.00**	2.63	0.21	0.90	0.18	0.20	0.76
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	72.80	**100.00**	0.63	98.90	2.71	0.30	0.90	0.16	0.20	0.77
ChemDFM$_{\text{Llama}}$	18.20	87.00	0.67	98.90	2.90	0.27	0.90	0.14	0.23	0.76
Generalist LLMs										
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**84.20**	99.20	0.62	99.52	2.55	0.42	**0.93**	**0.12**	**0.17**	0.81
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	80.00	91.20	0.57	99.00	**2.49**	**0.44**	**0.93**	**0.12**	**0.17**	**0.82**
Impv-Gen	15.7	-0.8	-1.6	0.6	5.9	40.0	3.3	25.0	15.0	5.2

The metrics, notations, and formatting have the same meanings as those in Table A3.

Table A11: Overall Performance on BDEQ

Model	SR$^{\uparrow}$	Val$^{\uparrow}$	Sim$^{\uparrow}$	Nov$^{\uparrow}$	SAS$^{\downarrow}$	RI$^{\uparrow}$	APS			
							BBBP$^{\uparrow}$	DRD2$^{\uparrow}$	hERG$^{\downarrow}$	QED$^{\uparrow}$
General-purpose LLMs										
Mistral (0-shot)	3.00	78.00	**0.71**	**100.00**	2.97	1.05	0.88	0.06	**0.40**	0.75
Llama (0-shot)	2.20	96.00	0.68	**100.00**	3.46	0.60	**0.96**	0.07	0.48	0.78
Claude-3.5 (0-shot)	4.80	96.60	0.62	**100.00**	2.76	0.57	0.92	0.04	0.52	0.79
GPT-4o (0-shot)	3.40	87.60	**0.71**	**100.00**	**2.75**	0.42	0.93	0.07	0.55	**0.82**
Mistral (1-shot)	21.60	99.80	0.58	84.26	3.11	1.16	0.91	0.15	0.49	0.77
Llama (1-shot)	11.40	99.60	0.51	68.42	3.48	1.54	0.92	**0.19**	0.49	0.79
Claude-3.5 (1-shot)	7.20	97.60	0.55	**100.00**	2.88	1.22	0.95	0.08	0.53	0.79
GPT-4o (1-shot)	2.20	86.00	0.70	**100.00**	2.81	0.83	0.95	0.09	0.57	0.80
Foundational LLMs for Chemistry										
LlaSMol$_{\text{Mistral}}$	18.20	**100.00**	0.60	**100.00**	2.86	0.65	0.92	0.07	0.49	0.80
ChemDFM$_{\text{Llama}}$	3.00	87.40	0.68	**100.00**	3.13	**1.64**	0.94	0.08	0.49	0.79
Generalist LLMs										
GeLLM^{4}O-C-P(10)$_{\text{Mistral}}$	**29.20**	98.40	0.60	**100.00**	2.78	1.22	0.92	0.08	0.45	0.80
GeLLM^{4}O-C-P(10)$_{\text{Llama}}$	28.40	92.20	0.58	**100.00**	**2.75**	0.88	0.92	0.07	0.47	0.80
Impv-Gen	35.2	-1.4	3.4	18.7	10.6	5.2	1.1	-46.7	8.2	3.9

The metrics, notations, and formatting have the same meanings as those in Table A3.

ment) among all the valid generations given the input. **(2)** Constraint Violation Rate (CVR) of a property as the average percentage of generated molecules that fail to satisfy a specific individual property constraint among all the valid generations given the input.

Our detailed failure analysis reveals distinct patterns in the model's limitations across different property combinations: **(1)** DRD2 improvement emerges as the most challenging constraint, with CVR consistently exceeding 59% across all examined objectives (BDPQ: 59.25%, DHMQ: 67.32%, CDE: 69.38%, BDEQ: 83.36%). This suggests a fundamental difficulty in optimizing dopamine receptor binding while satisfying other constraints. This may be due to the weaker correlation between DRD2 activity and the other properties, limiting the model's ability to identify mutually beneficial modifications. **(2)** Improvement constraints generally exhibit higher CVR than stability constraints (e.g., 67.32% for improving DRD2 and 29.58% for improving QED, while only 3.49% for stabilizing HIA in DHMQ). This indicates that it is more challenging for the model to improve properties – especially DRD2 – than to maintain existing favorable properties in multi-property optimization. One possible explanation is that property improvements often require more substantial molecular modifications, which can easily disrupt the satisfaction of other concurrent constraints. **(3)** Stability requirements

Table A12: Overall Performance on HLMPQ

Model	SR↑	Val↑	Sim↑	Nov↑	SAS↓	RI↑	APS				
							HIA↑	LIV↓	MUT↓	PlogP↑	QED↑
General-purpose LLMs											
Mistral (0-shot)	11.60	82.40	**0.79**	**100.00**	2.91	**1.76**	0.99	**0.38**	0.20	0.51	0.77
Llama (0-shot)	20.20	99.40	0.72	98.02	2.82	0.68	**1.00**	0.54	0.23	0.70	**0.79**
Claude-3.5 (0-shot)	21.00	97.00	0.66	99.05	2.72	0.59	**1.00**	0.46	0.24	0.69	**0.79**
GPT-4o (0-shot)	12.80	87.60	0.72	**100.00**	2.78	0.47	**1.00**	0.48	0.20	0.49	0.75
Mistral (1-shot)	55.60	99.80	0.62	97.12	2.59	0.77	0.99	0.54	0.21	1.08	0.77
Llama (1-shot)	28.00	99.60	0.70	97.86	2.72	0.75	**1.00**	0.56	0.24	0.83	0.78
Claude-3.5 (1-shot)	25.00	95.00	0.61	97.60	2.60	0.72	**1.00**	0.53	0.25	0.89	0.78
GPT-4o (1-shot)	13.40	87.40	0.71	**100.00**	2.82	0.65	**1.00**	0.50	0.21	0.61	0.73
Foundational LLMs for Chemistry											
LlaSMol$_{Mistral}$	37.80	**100.00**	0.68	**100.00**	2.66	0.66	**1.00**	0.58	0.22	0.92	0.73
ChemDFM$_{Llama}$	10.80	90.60	0.68	98.15	3.01	1.04	0.98	0.43	0.19	0.68	0.77
Generalist LLMs											
GeLLM^{4}O-C-P(10)$_{Mistral}$	**74.60**	99.80	0.61	99.46	2.49	1.36	**1.00**	0.53	**0.18**	1.43	**0.79**
GeLLM^{4}O-C-P(10)$_{Llama}$	65.40	90.80	0.58	99.69	**2.41**	1.35	**1.00**	0.53	**0.18**	**1.53**	**0.79**
Impv-Gen	34.2	0.0	-1.6	2.4	3.9	76.6	1.0	1.9	14.3	32.4	2.6

The metrics, notations, and formatting have the same meanings as those in Table A3.

Table A13: Ablation on Property Combinations

Model GeLLM^{4}O-C	BPQ			BDPQ			DHMQ		
	SR↑	Sim↑	RI↑	SR↑	Sim↑	RI↑	SR↑	Sim↑	RI↑
Specialist LLMs									
-N$_{Mistral}$	71.00	0.57	2.59	**56.60**	0.50	**5.48**	44.60	0.57	2.96
Generalist LLMs									
-P(N)$_{Mistral}$	84.80	**0.63**	**2.64**	50.60	**0.58**	4.93	53.40	0.59	3.26
-P(6)$_{Mistral}$	**93.80**	0.60	2.53	54.40	**0.58**	4.76	**58.20**	0.60	**3.52**
-P(10)$_{Mistral}$	89.40	0.62	2.30	48.40	**0.58**	5.05	52.20	**0.61**	2.24

For each task, the best-performing model is in **bold**, and the next best model is underlined. The best generalist LLM is marked in blue. The best model in each group is selected based on SR for each task.

Table A14: Ablation on Instruction Diversity

Model	BPQ	ELQ	ACEP	BDPQ	DHMQ
Single	77.8	75.6	67.4	35.2	36.4
Diverse	**89.6**	**87.6**	**78.0**	**46.6**	**50.2**

’Single’ and ’Diverse’ indicate whether models are trained with a single or 30 diverse instruction template(s). The best-performing model is highlighted in bold if the performance difference exceeds 5%.

Table A15: Overview of FR and CVR for challenging objectives

Task	#Inp	FR	Improvement CVR							Stability CVR						
			BBBP↑	CARC↓	DRD2↑	hERG↓	HIA↑	PlogP↑	QED↑	BBBP	CARC	DRD2	hERG	HIA	PlogP	QED
BDPQ	96	80.22	-	-	59.25	-	-	25.85	-	6.66	-	-	-	-	-	21.92
DHMQ	202	85.39	-	-	67.32	-	31.76	-	29.58	-	-	-	-	3.49	-	-
CDE	130	80.91	-	-	69.38	-	29.88	-	-	-	34.69	-	-	-	-	-
BDEQ	122	91.06	12.30	-	83.36	12.39	-	-	6.59	-	-	-	-	-	-	-

“#Inp" denotes the number of test molecules for the specific objective. ↑ and ↓ indicate whether higher or lower scores of a given property are desirable for improvement; - indicates properties not involved in the objective.

show more variable performance: while some properties like BBBP and HIA demonstrate relatively low CVR (6.66% and 3.49% respectively), others like CARC stabilization seems more challenging (34.69%). This suggests that certain properties are more sensitive to structural edits, and may require more careful regularization in training.