DePO: Elicit Chemical Reasoning via Demonstration-Guided Policy Optimization

Xuan Li [1] **Zhanke Zhou** [1] **Zongze Li** [1] **Jiangchao Yao** [2 3] **Yu Rong** [4 5] **Lu Zhang** [1] **Bo Han** [1]

Abstract

Large language models (LLMs) have demonstrated impressive mathematical reasoning capabilities when trained with reinforcement learning with verifiable rewards (RLVR), particularly through Group Relative Policy Optimization (GRPO). However, extending these methods to scientific domains such as *molecular optimization* is challenging, as LLMs often lack the necessary domain-specific reasoning skills. Molecular optimization involves optimizing molecular properties while preserving structural similarity, leading to a complex combinatorial search. Existing models struggle due to conflicting objectives, limited chemical reasoning, and the scarcity of datasets with intermediate reasoning steps, which hinders learning effective strategies. To address these issues, we introduce **De**monstration-guided **P**olicy **O**ptimization (DePO). This framework leverages reference molecules as demonstrations to guide model exploration toward promising regions of chemical space. Specifically, DePO incorporates demonstrations as supervised signals for each reasoning chain, to regularize the search direction while preserving the model's reasoning capabilities. Experiments show that DePO significantly outperforms both supervised fine-tuning and GRPO approaches across key molecular optimization metrics, and excels in balancing the competitive optimization objectives. DePO also shows generalization capabilities and inference-scaling properties.

[1]Department of Computer Science, Hong Kong Baptist University [2]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University [3]Shanghai AI Laboratory [4]DAMO Academy, Alibaba Group, Hangzhou, China [5]Hupan Lab, Hangzhou, China. Correspondence to: Bo Han <bhanml@comp.hkbu.edu.hk>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

Large language models (LLMs) have revolutionized problem-solving by leveraging sophisticated reasoning capabilities and their vast knowledge repositories (Sun et al., 2023; Yu et al., 2024; Zhong et al., 2024; Chen et al., 2025a;b; Zhou et al., 2025). Conventional approaches employ manually designed prompts to enhance reasoning abilities, ranging from in-context learning (Tang et al., 2023) to chain-of-thought prompting (Wei et al., 2022) and its variants (Yao et al., 2023). In contrast, post-training methods such as supervised fine-tuning (SFT) further augment the reasoning capabilities of LLMs. By training with high-quality chain-of-thought demonstrations, LLMs acquire the capacity to perform deliberative reasoning before generating answers, a crucial ability to solve tasks requiring multiple reasoning steps, as evidenced by their effectiveness in tackling mathematical problems (Zelikman et al., 2022).

However, the curation of high-quality chain-of-thought demonstrations is resource-intensive and necessitates specialized domain expertise, rendering it impractical for scaling to domains beyond mathematics. Recent advances, notably DeepSeek-R1 (Guo et al., 2025), propose enhancing LLMs' generalizable reasoning capabilities through reinforcement learning with verifiable rewards (RLVR), requiring only question-answer pairs and a rule-based reward function. Specifically, DeepSeek-R1 employs Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to optimize models using reward signals derived from response accuracy and format adherence. This approach yields substantial improvements in generalizable reasoning capabilities, encouraging models to reason strategically by incorporating self-reflection and self-correction mechanisms when encountering complex tasks.

Besides mathematical reasoning, LLMs have achieved notable progress in scientific domains such as interdisciplinary literature analysis and scientific data interpretation (Zhang et al., 2023; AI4Science & Quantum, 2023; Gottweis et al., 2025). However, despite their broad domain knowledge and ability to process complex research articles, LLMs continue to face challenges with multi-step reasoning in specialized scientific tasks (Wang et al., 2023; Mirza et al., 2024). A pertinent example is *molecular optimization* (Figure 1), which

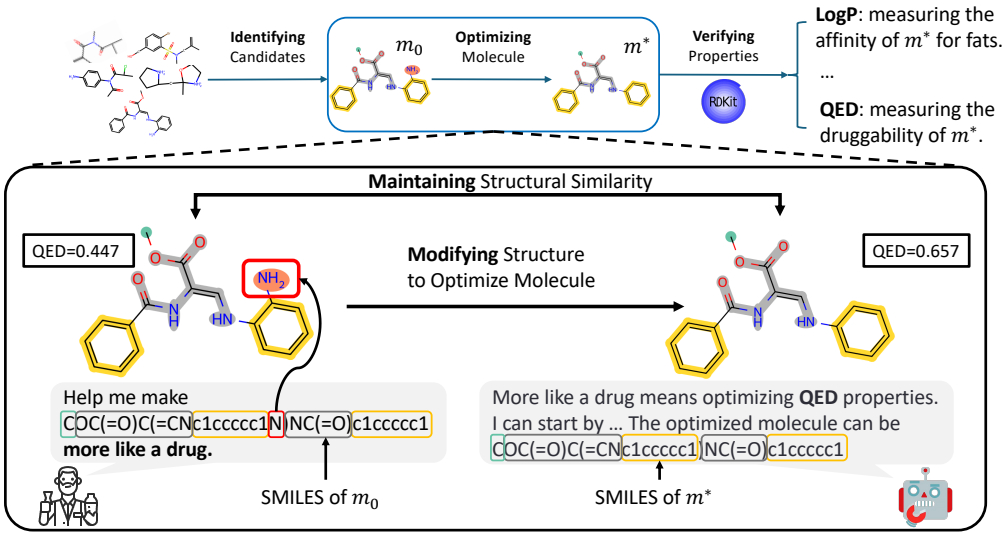

Figure 1. Molecular optimization aims to optimize the given molecule by modifying its components while maintaining the structural similarity of the original molecule after the modification. The molecule is represented as SMILES (Weininger, 1988), a sequence of symbols that represent atoms and bonds.

necessitates iterative analysis of molecular structures, the proposal and implementation of modifications, and evaluation of resultant properties (Talanquer, 2022; Guo et al., 2023; Liao et al., 2024). This task is crucial in drug discovery, where the goal is to enhance pharmacological properties while maintaining structural similarity to ensure biological activity (López-Pérez et al., 2024).

Despite the success of RLVR in mathematical domains, it often fails to transfer to scientific tasks such as molecular optimization, which require both specialized domain knowledge and sophisticated multi-step reasoning (Yue et al., 2025). Notably, the effectiveness of RLVR is inherently constrained by the model's pre-existing knowledge and reasoning capacity (Yue et al., 2025; Gandhi et al., 2025). Furthermore, molecular optimization datasets typically only contain final answers (demonstration molecules) without intermediate reasoning steps, and directly applying SFT on these demonstration molecules before RLVR can undermine the model's reasoning ability. This approach tends to encourage shortcut learning and deterministic outputs, rather than supporting the step-by-step reasoning process necessary for complex domain-specific tasks. This limitation raises a critical research question:

How can we leverage demonstrations to extend RLVR's knowledge frontier, enabling the robust multi-step reasoning required to handle complex challenges like molecular optimization?

In this paper, we propose Demonstration-guided Policy Optimization (DePO), a novel framework to boost the LLMs' chemical reasoning for molecular optimization tasks. DePO alleviates the limitations of conventional RLVR, which often suffers from unguided and inefficient exploration, by explicitly integrating reference molecules as demonstrations into the policy optimization process. Specifically, DePO augments the policy optimization objective with a demonstration-guided term that encourages the model to generate solutions consistent with demonstrations. During training, the model is supervised to match the demonstrated molecules, while being allowed to explore intermediate reasoning steps. DePO constrains the search space to chemically valid and promising regions, enabling the model to acquire domain knowledge and reasoning capabilities.

Empirically, we evaluate our method on instruction-based molecular optimization benchmarks, including TOMG-Bench (Li et al., 2024a) and MuMOInstruct (Dey et al., 2025). DePO achieves up to 13% improvement compared to SFT and other baseline approaches. Beyond instructions seen during training, we demonstrate the effectiveness of DePO on unseen instruction styles, highlighting its capacity to generalize to novel scenarios. Additionally, our approach exhibits strong inference-scaling capabilities, where optimization success rates proportionally increase with additional attempts, further substantiating the efficacy of DePO in extending RLVR beyond mathematical reasoning to complex scientific domains.

We summarize our contributions as follows:

- We identify the insufficiency of RLVR in scientific domains, which is limited by the model's capability in reasoning under domain-specific constraints (Section 3).
- We introduce DePO, a novel framework that synergistically combines reinforcement learning with expert demonstrations to address the challenges inherent in scientific reasoning tasks (Section 4).

- We empirically evaluate DePO on molecular optimization tasks, demonstrating its effectiveness in enhancing the generalizability of LLMs' reasoning abilities in scientific domains (Section 5).

2. Preliminary

In this section, we first introduce the basic idea of molecular optimization, followed by the existing approaches and the advantage of using LLMs for molecular optimization.

2.1. Molecular Optimization

Shown in Figure 1, molecular optimization involves modifying molecular structures to enhance desired properties, such as drug-likeness measured by QED (Bickerton et al., 2012), while preserving structural similarity to the original molecule to retain its biological activity (López-Pérez et al., 2024; Lipinski & Hopkins, 2004). Molecular optimization can be formulated as a constrained optimization problem:

$$m^* = \arg\max_{m \in \mathcal{M}} F(m_0) \quad \text{s.t.} \quad \text{Similarity}(m^*, m_0) \geq \delta, \quad (1)$$

where m is an original molecular structure, $\mathcal{M}$ represents the set of valid molecules spanning the *chemical space*, $F : \mathcal{M} \to \mathbb{R}$ is a scalar-valued function evaluating the desired molecular property, such as drug-likeness or solubility. $\text{Similarity}(\cdot, \cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ quantifies structural similarity of two molecules, and $\delta \in [0, 1]$ is the threshold ensuring sufficient similarity. Notably, multiple valid solutions m^* may exist, as any molecule satisfying the objective function is considered optimal.

Conventional methods like Monte Carlo Tree Search (Yang et al., 2017) and Genetic Algorithms (Nigam et al., 2022; Fu et al., 2022) exhaustively search the chemical space for molecules with desired properties, but their computational inefficiency limits scalability (Stumpfe & Bajorath, 2012). Generative models address this by learning the chemical space distribution, enabling efficient exploration of valid regions. VAE (Liu et al., 2018) generates novel compounds via latent space navigation, GFlowNet (Bengio et al., 2021) optimizes molecular generation as a flow-matching problem, and diffusion models like EDMs (Hoogeboom et al., 2022) produce molecules through iterative denoising.

2.2. LLMs for Chemical Tasks

Despite their merits, conventional approaches exhibit inherent limitations in synthesizing molecules with precise, tailored properties (Li et al., 2024b). Furthermore, these methods demonstrate insufficient generalization capabilities when confronted with novel tasks, thereby impeding their practical utility in addressing emerging therapeutic challenges and pharmaceutical requirements (Dey et al., 2025; Li et al., 2024a). These limitations motivate us to explore the potential of LLMs for molecular optimization, which excels in generalizing to unseen tasks with limited demonstration or ambiguous instructions (Chang et al., 2024). Notably, LLMs have demonstrated remarkable capabilities in understanding molecular properties and their interactions (Guo et al., 2023). These investigations demonstrate that LLMs acquire sufficient chemical knowledge to understand the molecules and conduct valid operations.

Nevertheless, LLMs are limited in transferring their general reasoning capabilities to chemical domains. Comprehensive empirical evaluations, including Scibench (Wang et al., 2023) and ChemBench (Mirza et al., 2024), have systematically documented performance degradation when LLMs confront tasks requiring reasoning under domain-specific constraints, such as structural validity preservation or adherence to molecular property preferences. This limitation is particularly consequential for molecular optimization, which necessitates sophisticated reasoning about molecular structures and their associated physicochemical properties within a highly constrained chemical space.

2.3. Enhancing LLM Reasoning via RLVR

Recent advances in LLM reasoning capabilities, exemplified by DeepSeek-R1 (Guo et al., 2025), demonstrate that reinforcement learning (RL) through GRPO (Shao et al., 2024) with rule-based rewards can substantially enhance LLMs' reasoning faculties, particularly for complex mathematical reasoning tasks (Shao et al., 2024; Team, 2025; Guo et al., 2025). GRPO builds upon the Proximal Policy Optimization (PPO) (Schulman et al., 2017) but eliminates the critic model and Generalized Advantage Estimation (GAE), thus improving computational efficiency.

Given the question-answer pair (q, a) that is *i.i.d.* sampled from an underlying distribution $\mathcal{D}$, where q denotes the query and a represents the ground-truth answer. Let $\pi_\theta(\cdot|\cdot)$ be the current LLM policy parameterized by θ, $\{o_i\}_{j=1}^{G}$ denotes the G independent responses generated from the old policy model $\pi_{\text{old}}(\cdot|q)$, and $r(\cdot, \cdot)$ represents the reward function that quantifies the correctness of o_i with respect to q and a, and ϵ is the hyper-parameter. Let $\pi_{\text{ref}}(\cdot|q)$ denotes the reference policy model. Formally, GRPO optimizes the policy model π_θ by maximizing the following objective:

$$\begin{aligned}
\mathcal{J}_{\text{GRPO}}(\pi_\theta) \triangleq \mathbb{E}_{\substack{(q,a)\sim\mathcal{D}, \\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \\
\Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{k=1}^{|o_i|} \Bigg(\min \Bigg(\frac{\pi_\theta(o_{i,k}|q, o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q, o_{i,<k})} \hat{A}_{i,k}, \\
\text{clip} \left(\frac{\pi_\theta(o_{i,k}|q, o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q, o_{i,<k})}, \, 1-\varepsilon, \, 1+\varepsilon \right) \hat{A}_{i,k} \Bigg) \\
- \beta \, \mathbb{D}_{\text{KL}} \left(\pi_\theta \,\|\, \pi_{\text{ref}} \right) \Bigg) \Bigg],
\end{aligned} \quad (2)$$

where $\hat{A}_{i,k} \triangleq \frac{r(o_i,a) - \text{mean}(\{r(o_i,a)\}_{i=1}^{G})}{\text{std}(\{r(o_i,a)\}_{i=1}^{G})}$ denotes the group relative reward. GRPO also incorporates the K3 KL-

Figure 2. Comparison of GRPO, GRPO with SFT initialization, and DePO across three key metrics: training reward distribution (**left**), output sequence length (**middle**), and target property optimization performance (**right**). GRPO-based models, with or without SFT initialization, fail to balance property optimization and structural constraints. SFT-trained models further exhibit diminished reasoning ability, reflected by shorter completions. In contrast, DePO achieves a better trade-off between property and structure, and generates molecules with more detailed reasoning and improved target property performance. Experimental details are provided in Appendix A.

divergence estimator (Schulman., 2020), which is formulated as follows:

$$\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta||\pi_{\mathrm{ref}}\right) = \frac{\pi_{\mathrm{ref}}(o_{i,k}|q,o_{i,<k})}{\pi_\theta(o_{i,k}|q,o_{i,<k})} - \log\frac{\pi_{\mathrm{ref}}(o_{i,k}|q,o_{i,<k})}{\pi_\theta(o_{i,k}|q,o_{i,<k})} - 1. \tag{3}$$

3. Scare Reward Signal Limiting the Exploration

Recall Equation (2), the RL objective aims to optimize the policy model to obtain higher rewards, which heavily relies on the quality of the model's own generation results. However, LLMs struggle in generating effective optimization results for positive feedback, as well as exploring the search space efficiently. Without sufficiently informative feedback to guide the search process, the model's exploration trajectory becomes stochastic, failing to converge toward optimal solutions, which should satisfy the requirements of the target property while maintaining the structural constraints.

To empirically substantiate the above claims, we examine the training dynamics of models under various configurations. Specifically, we conduct RLVR with Qwen-2.5-3B-Instruct model and employ molecular property and structural constraints as the reward function. Figure 2 presents the results and we derive the following observations.

Observation 3.1 (*GRPO cannot balance the competitive molecular optimization constraints*)**.** Models trained with GRPO exhibit a conservative bias, generating molecules nearly identical to the input as Figure 2 (Left). While this approach easily satisfies structural similarity constraints, it prevents meaningful molecular modifications necessary for property enhancement. This leads to suboptimal property rewards and failure to meet the optimization objective. The resulting molecular outputs lack diversity and fail to adequately explore the space of improved structures, yielding poor optimization performance as Figure 2 (Right).

Observation 3.2 (*GRPO with SFT initialization cannot balance the trade-off between property optimization and structural constraints*)**.** While SFT effectively integrates domain-specific knowledge into LLMs (Mecklenburg et al., 2024), its application with simple question-answer pairs, which are prevalent in text-based molecular generation, appears insufficient for balancing the competing constraints inherent in molecular optimization tasks. As illustrated in Figure 2 (Left), models initialized with SFT tend to generate molecules that significantly deviate from the input structure, despite meeting the target property requirements. These models fail to maintain structural similarity, producing chemically valid but irrelevant optimizations that violate the constraints of molecular optimization tasks. Models trained with SFT-based initialization are also unable to generate molecules with detailed reasoning.

Observation 3.3 (*GRPO cannot recover the reasoning ability from SFT-initialized model*)**.** Crucially, applying GRPO to an SFT-initialized model fails to restore step-by-step reasoning. As illustrated in Figure 2 (Middle), the model persistently generates brief outputs during RL training, lacking substantive multi-step reasoning. While the model remains capable of generating chemically valid molecules, it fails to regain the reasoning ability to effectively balance the trade-offs required for successful optimization. Once SFT has induced a preference for direct responses, subsequent GRPO training is unable to restore the model's ability to engage in intermediate, deliberative reasoning.

These observations suggest fundamental limitations of current approaches: they struggle with balancing the competing objectives in molecular optimization while maintaining the model's reasoning capabilities for better optimization. These limitations of GRPO and SFT-initialized models motivate the need for a more principled solution. Ideally, such an approach should guide exploration within the chemical

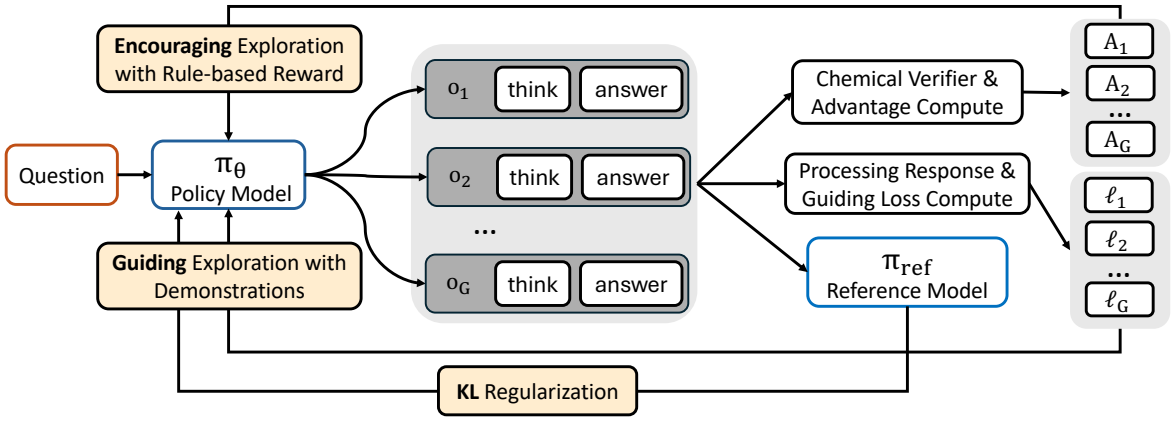

Figure 3. Schematic of the DePO framework. The policy model generates multiple completions, each containing reasoning steps ("think") and final answers. The model learns to reason in the chemical space through the advantage function computed on the full response's chemical validity, while being guided toward promising regions by the supervised loss applied to the processed responses.

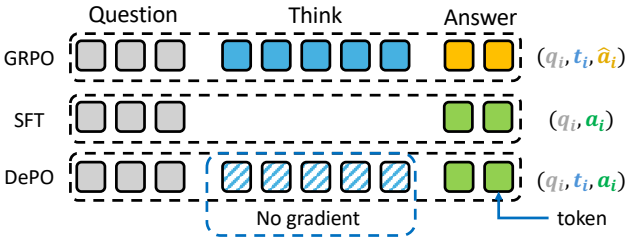

Figure 4. Illustration of token processing and gradient flow across GRPO, SFT, and DePO.

space, while balancing structural constraints and property optimization objectives. Moreover, it should navigate these trade-offs without diminishing the model's reasoning abilities, thereby enabling effective molecular optimization.

4. DePO: Demonstration-Guided Policy Optimization

Motivated by these findings, we propose a novel framework that leverages demonstrations to better direct the policy model's search process, namely Demonstration-guided Policy Optimization (DePO), and detail how it addresses the challenges identified above. Rather than depending exclusively on the model's knowledge, our approach constrains the exploration space by demonstrating the reference molecules. In the realm of molecular optimization, we can leverage the existing question-answer pairs to guide the model's exploration. This approach is especially useful for molecular optimization because the chemical space is enormous, and evaluating molecules often requires specialized knowledge that LLMs may lack from their pretraining (Kim et al., 2023; Jiang et al., 2023).

Conceptually, we incorporate demonstrations to the policy model by maximizing the log-likelihood of the reference response by $\arg\max_{\pi_\theta} \mathbb{E}_{(q,a)\sim\mathcal{D}} [\log \pi_\theta(a|q)]$, where $\mathcal{D}=\{(q_i, a_i)\}$ is the dataset of reference molecules without intermediate reasoning steps. However, naively maximizing the log-likelihood of reference molecules risks inducing deterministic behavior, wherein the model bypasses intermediate reasoning processes in favor of direct answer generation. To address this limitation, we introduce the exploration guidance term to the objective function that replaces each generation's final answer $\hat{a}_i$ with the demonstrated solution a_i.

As illustrated in Figure 4, DePO resembles the standard RLVR procedure, with an additional supervised guidance term where the model-generated answer ($\hat{a}_i$) is substituted with the demonstrated solution (a_i). In our approach, we decompose the model's output o_i into two components: the intermediate reasoning tokens t_i and the final answer $\hat{a}_i$, such that $o_i = [t_i; \hat{a}_i]$. This decomposition allows us to selectively replace only the final answer while preserving the model's reasoning process. We formally represent this process as $\pi_\theta(a_i|q, t_i)$, where t_i denotes the sequence of intermediate reasoning tokens. Furthermore, we employ gradient masking for the intermediate reasoning steps, effectively excluding these tokens from parameter updates during optimization. This approach prevents the model from learning potentially erroneous reasoning patterns while preserving its capacity for exploratory thinking.

The resulting framework, shown in Figure 3, preserves the model's capacity for deliberative reasoning while simultaneously constraining its exploration to chemically valid and promising regions of the solution space.

The DePO objective with exploration guidance is given as Equation (4). It offers a balanced approach to molecular optimization by allowing the policy model to learn from both its own exploration (via $\hat{A}_{i,k}$) and from expert demonstrations. Notably, the external guidance is essentially imposing supervision on the π_θ's final predictions, which guides the

π_θ to generate more likely molecules.

$$\mathcal{J}_{\text{DePO}}(\pi_\theta) \triangleq \mathbb{E}_{\substack{(q,a)\sim\mathcal{D}, \\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|}\min\left(\frac{\pi_\theta(o_{i,k}|q,o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q,o_{i,<k})}\hat{A}_{i,k},\right.$$
$$\left.\text{clip}\left(\frac{\pi_\theta(o_{i,k}|q,o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q,o_{i,<k})},\ 1-\varepsilon,\ 1+\varepsilon\right)\hat{A}_{i,k}\right)$$
$$+\gamma\cdot\underbrace{\log\pi_\theta(a_i|q,t_i)}_{\text{Guiding Exploration}}$$
$$-\ \beta\cdot\left(\frac{\pi_{\text{ref}}(o_{i,k}|q,o_{i,<k})}{\pi_\theta(o_{i,k}|q,o_{i,<k})}-\log\frac{\pi_{\text{ref}}(o_{i,k}|q,o_{i,<k})}{\pi_\theta(o_{i,k}|q,o_{i,<k})}-1\right)\Bigg]. \tag{4}$$

4.1. Reward Design

The reward function serves as the primary feedback signal for the effectiveness of the reasoning process of the model. Recall Equation (1), molecular optimization requires maintaining the structural similarity between the generated molecule and the original ones, along with the target property. We define the reward function as follows:

- **Structural similarity** r_{struct}: We employ the Tanimoto similarity (Bajusz et al., 2015) to measure the similarity between the generated molecule and the original ones.

$$r_{\text{struct}}(m^*, m_0) = \frac{|FP(m^*)\cap FP(m_0)|}{|FP(m^*)\cup FP(m_0)|}\in[0,1], \tag{5}$$

  where $FP(m)$ represents the molecular fingerprint of molecule m. This similarity metric quantifies the structural overlap between two molecular fingerprints, with values ranging from 0 (completely distinct structures) to 1 (structurally identical molecules).

- **Target property** r_{prop}: We define a binary reward function with respect to the target property value function F (e.g., logP, QED). The reward is assigned a value of 1 if the generated molecule m^* achieves a favorable change in the target property relative to the original molecule m_0, as specified by the optimization objective (e.g., minimizing LogP). Otherwise, the reward is 0.

$$r_{\text{prop}}(m^*, m_0) = \begin{cases} 1, & \text{if } F(m^*)\succeq F(m_0), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

5. Experiments

In this section, we evaluate the performance of DePO. We first outline the experimental setup (Section 5.1), followed by a detailed discussion of the results (Section 5.2). Lastly, we provide case studies and investigate the DePO's inference-scaling property (Section 5.3).

5.1. Experiment Settings

In what follows, we describe the setting of the experiments, including the dataset, baselines, and evaluation metrics. Detailed settings are provided in Appendix A.

Datasets. We employ two instruction-based molecular optimization benchmarks, TOMG-Bench (Li et al., 2024a) and MuMOInstruct (Dey et al., 2025), to evaluate the knowledge of LLM on molecular structure and properties.

- **TOMG-Bench** is a single-objective molecular optimization benchmark encompassing two primary task types: structure-based and property optimization. Structure-based tasks require the model to modify designated functional groups within molecules, whereas property tasks involve adjusting molecular structures to improve specific chemical properties. We covers three structure-based tasks (AddComponent, DelComponent, SubComponent) and three property tasks (QED, LogP, MR).
- **MuMOInstruct** is a multi-objective molecular optimization benchmark. Each task involves optimizing several pharmaceutical properties at once, increasing the challenge for the model. To systematically evaluate generalization, MuMOInstruct includes both seen and unseen instruction styles to assess the model's robustness to instruction variation. Four pharmaceutical properties are considered: plogP (lipophilicity), QED (drug-likeness), BBBP (blood-brain barrier permeability), and DRD2 (dopamine receptor D2 binding affinity). Notably, we evaluate three tasks with seen and unseen instruction styles with different combinations of optimization tasks, including BDP (BBBP+DRD2+plogP), BDQ (BBBP+DRD2+QED), and BPQ (BBBP+DRD2+plogP).

Baselines. We use `Qwen-2.5-3B-Instruct` as our backbone and primary *baseline* model. For comparison, we evaluate the following approaches: *Distill-SFT*, which performs SFT on the s1.1K dataset (Muennighoff et al., 2025) to impart extended reasoning abilities by leveraging distilled responses from DeepSeek-R1 on predominantly mathematical tasks; *SFT*, which refers to training the backbone model on the training split of the target molecular optimization dataset in an supervised manner; *GRPO*, which applies RLVR to the backbone model, following the objective in Equation (2) and the reward function described in Section 4.1; and *GRPO (SFT init)*, which is identical to GRPO except that it is trained starting from the *SFT* model. We conduct experiments on 3 Nvidia A100s.

Evaluation Metrics. We evaluate model performance using three complementary metrics. `Success Rate` quantifies the proportion of molecular optimization tasks in which the model successfully achieves the specified property objectives. `Similarity` is measured by the Tanimoto coefficient (Bajusz et al., 2015), which assesses

Task type	Objective (↑)	Baseline	Distill-SFT	SFT	GRPO	GRPO (SFT init)	DePO
Structure-based optimization	AddComponent	0.065	0.060	<u>0.147</u>	0.005	<u>0.156</u>	**0.239**
	DelComponent	0.092	0.128	<u>0.154</u>	0.008	**0.176**	<u>0.140</u>
	SubComponent	0.047	0.050	0.264	0.052	<u>0.300</u>	**0.344**
Property optimization	QED	0.130	0.123	0.207	0.123	<u>0.193</u>	**0.236**
	LogP	0.168	0.135	0.206	**0.305**	<u>0.183</u>	**0.297**
	MR	0.173	0.132	<u>0.238</u>	0.188	<u>0.225</u>	**0.293**

Table 1. Comparison of different methods on TOMG-Bench target on structural and property optimization. The best results for each task are bolded, and the second-best is underlined.

Task type	Objective (↑)	Baseline	Distill-SFT	SFT	GRPO	DePO
Seen Instruction	BDP	0.008	0.016	<u>0.101</u>	**0.118**	**0.117**
	BDQ	0.004	0.002	**0.089**	0.039	<u>0.058</u>
	BPQ	0.010	0.011	<u>0.115</u>	<u>0.120</u>	**0.139**
Unseen Instruction	BDP	0.007	0.002	0.081	<u>0.108</u>	**0.113**
	BDQ	0.004	0.002	**0.088**	0.036	<u>0.054</u>
	BPQ	0.006	0.007	0.104	<u>0.107</u>	**0.144**

Table 2. Overall Performance in MuMOInstruct benchmark with seen and unseen instructions. The best results for each task are bolded, and the second-best is underlined.

the structural similarity between the input and optimized molecules. To jointly capture both optimization effectiveness and structural preservation, we report the product `Success Rate` × `Similarity`, which reflects the model's ability to balance property improvement with maintenance of molecular integrity.

5.2. Quantitative Results

We summarize the empirical observations *w.r.t.* the experimental results in Tables 1 and 2.

DePO elicits the model's chemical reasoning on single-objective optimization tasks. Table 1 summarizes the results for single-objective molecular optimization. For structure-based tasks, DePO achieves the best performance on `AddComponent` and `SubComponent`, corresponding to improvements of 8.3% and 4% over the next best method, respectively. For property-based optimization, DePO achieves superior or competitive performance compared to all baselines, highlighting its effectiveness and robustness across evaluation settings, achieving up to 13.0% absolute improvement over the base model. Notably, GRPO without SFT initialization performs markedly worse, particularly on structure-based tasks, underscoring the challenges of unconstrained exploration in the vast chemical space. In contrast, DePO, which integrates demonstration guidance, consistently outperforms SFT and GRPO, yielding more effective molecular optimization.

DePO helps the model to balance multi-objective optimization problems. Table 2 presents the results for multi-objective molecular optimization. Notably, DePO outperforms baseline methods on `BDP` and `BPQ` tasks, achieving up to 4% improvements over baseline methods, highlighting its ability to effectively balance multiple competing objectives simultaneously.

DePO elicit model's generalization ability on unseen instruction styles. Shown in Table 2, the performance advantage of DePO is maintained for unseen instructions, achieving superior results despite the model encountering novel instruction formats. The most significant gains are observed in the `BDP` task, where DePO's approach to guided exploration proves particularly effective at navigating the complex optimization landscape involving multiple constraints. These results collectively validate that DePO's demonstration-guided approach constrains the exploration space while maintaining the model's reasoning capabilities across scenarios of multi-objective optimization.

5.3. Case Studies

In this section, we present case studies to demonstrate the effectiveness of DePO. We begin by showcasing molecular optimization outcomes on optimizing molecular MR value from TOMG-Bench, followed by an analysis of DePO's inference-time scaling behavior.

Chemically-validated reasoning. Figure 5 illustrates the qualitative differences in reasoning approaches between DePO and GRPO on a molecular optimization task. The

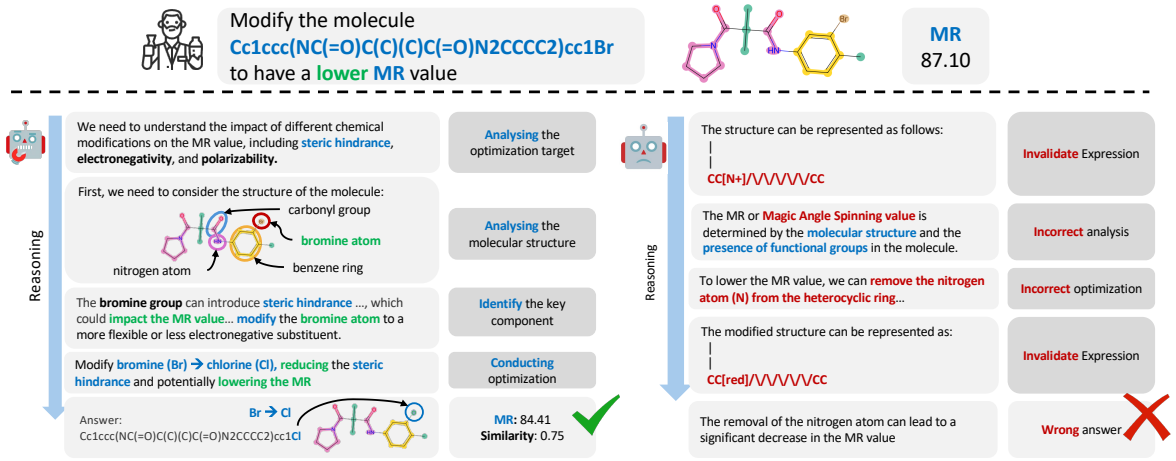

Figure 5. Comparative analysis of molecular optimization strategies employed by DePO (left) and GRPO (right). DePO applies chemically principled reasoning, accurately identifying key structural motifs and recommending a validated and effective substitution. In contrast, GRPO generates chemically unsound reasoning and suggests invalid structural changes.

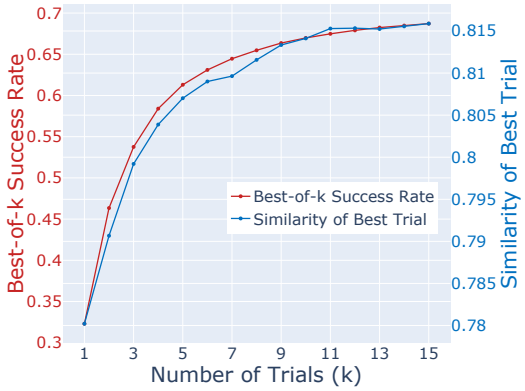

Figure 6. Inference-scaling effect of DePO.

left panel demonstrates DePO's chemically sound reasoning process: it correctly identifies structural elements (bromine, carbonyl groups, nitrogen atoms), articulates the relationship between steric hindrance and MR values, and proposes a valid transformation (substituting Br with Cl) that successfully reduces the MR value while maintaining molecular similarity. In contrast, the right panel reveals GRPO's flawed approach, which exhibits invalid chemical expressions, incorrect structural analysis, and proposes chemically implausible modifications (removing nitrogen from a heterocyclic ring). This comparison underscores DePO's capacity to generate not only structurally valid molecules, but also to produce coherent reasoning that captures the underlying chemical principles governing validated and robust molecular property optimization.

Inference-scaling properties. Figure 6 details DePO's inference-scaling characteristics. We experiment by sampling multiple times from the same task, namely optimizing the molecular MR value. The plot reveals that as the number of sampling trials (k) increases, DePO's best-of-k success rate (red curve) and the similarity of the trials (blue curve) both demonstrate marked improvements. These results underscore DePO's proficiency in leveraging increased computational budgets at inference.

6. Conclusion and Further Discussion

Conclusion. In this work, we identified key challenges in applying LLMs to molecular optimization tasks, particularly the difficulty in balancing competing objectives while maintaining reasoning capabilities. Conventional reinforcement learning approaches like GRPO struggle with sparse reward signals, leading to suboptimal exploration of the chemical space. We introduced DePO, a novel framework that effectively guides LLM exploration through expert demonstrations while preserving the model's reasoning abilities. Our empirical evaluations on TOMG-Bench and MuMOInstruct benchmarks demonstrate that DePO consistently outperforms existing methods across various tasks, achieving superior performance in both structure-based and property-based optimization scenarios. These results highlight the importance of guided exploration in complex domains and establish DePO as an effective approach for enhancing LLM reasoning for scientific tasks.

Limitations. Despite DePO's promising results, limitations remain. First, the framework relies on the availability of demonstrations, which may be scarce for novel or complex molecular optimization tasks. In addition, while our approach improves LLM reasoning for molecular optimization, the black-box nature of LLM still presents challenges for domain experts seeking to understand the precise reasoning behind specific structural modifications.

Impact Statement

This paper introduces DePO, a novel framework for enhancing large language model (LLM) reasoning in molecular optimization. By leveraging demonstration-guided policy optimization, our work aims to accelerate the discovery and design of new molecules, which could have significant positive impacts in fields such as medicine, materials science, and sustainable chemistry.

References

AI4Science, M. R. and Quantum, M. A. The impact of large language models on scientific discovery: a preliminary study using gpt-4. In *arXiv*, 2023.

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

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. In *NeurIPS*, 2021.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024.

Chen, A., Song, Y., Zhu, W., Chen, K., Yang, M., Zhao, T., et al. Evaluating o1-like llms: Unlocking reasoning for translation through comprehensive analysis. In *arXiv*, 2025a.

Chen, Q., Qin, L., Liu, J., Peng, D., Guan, J., Wang, P., Hu, M., Zhou, Y., Gao, T., and Che, W. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. In *arXiv*, 2025b.

Dey, V., Hu, X., and Ning, X. Gellm3o: Generalizing large language models for multi-property molecule optimization. In *arXiv*, 2025.

Edwards, C., Lai, T., Ros, K., Honke, G., Cho, K., and Ji, H. Translation between molecules and natural language. In *EMNLP*, 2022.

Fan, L., Tan, L., Chen, Z., Qi, J., Nie, F., Luo, Z., Cheng, J., and Wang, S. Haloperidol bound d2 dopamine receptor structure inspired the discovery of subtype selective ligands. *Nature communications*, 11(1):1074, 2020.

Fu, T., Gao, W., Coley, C., and Sun, J. Reinforced genetic algorithm for structure-based drug design. In *NeurIPS*, 2022.

Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., and Goodman, N. D. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. In *arXiv*, 2025.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., et al. Towards an ai co-scientist. In *arXiv*, 2025.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. In *arXiv*, 2025.

Guo, T., Nan, B., Liang, Z., Guo, Z., Chawla, N., Wiest, O., Zhang, X., et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. In *NeurIPS*, 2023.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *ICML*, 2022.

Jiang, S., Wang, Y., and Wang, Y. Selfevolve: A code evolution framework via large language models. *arXiv preprint arXiv:2306.02907*, 2023.

Kim, G., Baldi, P., and McAleer, S. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*, 2023.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Le Fevre, R. J. W. Molecular refractivity and polarizability. In *Advances in Physical Organic Chemistry*, volume 3, pp. 1–90. Elsevier, 1965.

Li, J., Li, J., Liu, Y., Zhou, D., and Li, Q. Tomg-bench: Evaluating llms on text-based open molecule generation. In *arXiv*, 2024a.

Li, J., Liu, Y., Liu, W., Le, J., Zhang, D., Fan, W., Zhou, D., Li, Y., and Li, Q. Molreflect: Towards in-context fine-grained alignments between molecules and texts. In *arXiv*, 2024b.

Liao, C., Yu, Y., Mei, Y., and Wei, Y. From words to molecules: A survey of large language models in chemistry. In *arXiv*, 2024.

Lipinski, C. and Hopkins, A. Navigating chemical space for biology and medicine. *Nature*, 432(7019):855–861, 2004.

Lipinski, C. A., Lombardo, F., Dominy, B. W., and Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25, 1997.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. In *NeurIPS*, 2018.

López-Pérez, K., Avellaneda-Tamayo, J. F., Chen, L., López-López, E., Juárez-Mercado, K. E., Medina-Franco, J. L., and Miranda-Quintana, R. A. Molecular similarity: Theory, applications, and perspectives. *Artificial Intelligence Chemistry*, 2(2):100077, 2024.

Mecklenburg, N., Lin, Y., Li, X., Holstein, D., Nunes, L., Malvar, S., Silva, B., Chandra, R., Aski, V., Yannam, P. K. R., et al. Injecting new knowledge into large language models via supervised fine-tuning. In *arXiv*, 2024.

Mirza, A., Alampara, N., Kunchapu, S., Ríos-García, M., Emoekabu, B., Krishnan, A., Gupta, T., Schilling-Wilhelmi, M., Okereke, M., Aneesh, A., et al. Are large language models superhuman chemists? In *arXiv*, 2024.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling. In *arXiv*, 2025.

Nigam, A., Pollice, R., and Aspuru-Guzik, A. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digital Discovery*, 1(4): 390–404, 2022.

Pei, Q., Zhang, W., Zhu, J., Wu, K., Gao, K., Wu, L., Xia, Y., and Yan, R. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *EMNLP*, 2023.

Schulman., J. Approximating kl divergence, 2020. URL http://joschu.net/blog/kl-approx.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. In *arXiv*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. In *arXiv*, 2024.

Sterling, T. and Irwin, J. J. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Stumpfe, D. and Bajorath, J. Exploring activity cliffs in medicinal chemistry: miniperspective. *Journal of medicinal chemistry*, 55(7):2932–2942, 2012.

Sun, J., Zheng, C., Xie, E., Liu, Z., Chu, R., Qiu, J., Xu, J., Ding, M., Li, H., Geng, M., et al. A survey of reasoning with foundation models. In *arXiv*, 2023.

Sundar, R., Jain, M. R., and Valani, D. Mutagenicity testing: Regulatory guidelines and current needs. In *Mutagenicity: assays and applications*, pp. 191–228. Elsevier, 2018.

Talanquer, V. The complexity of reasoning about and with chemical representations. *JACS Au*, 2(12):2658–2669, 2022.

Tang, X., Zheng, Z., Li, J., Meng, F., Zhu, S.-C., Liang, Y., and Zhang, M. Large language models are in-context semantic reasoners rather than symbolic reasoners, 2023.

Team, Q. Qwq-32b: Embracing the power of reinforcement learning. *URL: https://qwenlm. github. io/blog/qwq-32b*, 2025.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A. R., Zhang, S., Sun, Y., and Wang, W. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In *arXiv*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

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

Wu, D., Chen, Q., Chen, X., Han, F., Chen, Z., and Wang, Y. The blood–brain barrier: Structure, regulation and drug delivery. *Signal transduction and targeted therapy*, 8(1): 217, 2023.

Yang, X., Zhang, J., Yoshizoe, K., Terayama, K., and Tsuda, K. Chemts: an efficient python library for de novo molecular generation. *Science and technology of advanced materials*, 18(1):972–976, 2017.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. R. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.

Yu, F., Zhang, H., Tiwari, P., and Wang, B. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12): 1–39, 2024.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *arXiv*, 2025.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

Zhang, X., Wang, L., Helwig, J., Luo, Y., Fu, C., Xie, Y., Liu, M., Lin, Y., Xu, Z., Yan, K., et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. In *arXiv*, 2023.

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *ACL*, 2024.

Zhong, T., Liu, Z., Pan, Y., Zhang, Y., Zhou, Y., Liang, S., Wu, Z., Lyu, Y., Shu, P., Yu, X., et al. Evaluation of openai o1: Opportunities and challenges of agi. In *arXiv*, 2024.

Zhou, Z., Feng, X., Zhu, Z., Yao, J., Koyejo, S., and Han, B. From passive to active reasoning: Can large language models ask the right questions under incomplete information? In *ICML*, 2025.

Appendix

A. Experiment Settings

In this section, we provide the detailed experimental settings for all the experiments.

Pharmacological metrics. We employ the following pharmacological metrics for the molecular optimization tasks:

- `QED` (Quantitative Estimation of Drug-likeness) (Bickerton et al., 2012): QED provides a composite score that quantifies the drug-likeness of a molecule by integrating multiple molecular properties, such as molecular weight, logP, topological polar surface area, counts of hydrogen bond donors and acceptors, aromatic rings, rotatable bonds, and the presence of undesirable chemical functionalities.
- `LogP` (lipophilicity) (Lipinski et al., 1997): LogP quantifies the lipophilicity of a compound, reflecting its tendency to partition into non-polar (lipid-like) versus polar (aqueous) environments. Higher LogP values indicate greater solubility in non-polar solvents, which is relevant for drug absorption.
- `plogP` (penalized logP) denotes the logP penalized by the ring size and synthetic accessibility.
- `MR` (molar refractivity) (Le Fevre, 1965): MR is a physicochemical descriptor that quantifies molecular size and polarizability, both of which are critical for modeling molecular interactions with biological targets and membranes.
- `BBBP` (blood-brain barrier permeability) (Wu et al., 2023): BBBP quantifies a molecule's ability to permeate the blood-brain barrier (BBB), a selective interface that regulates molecular exchange between the systemic circulation and the central nervous system. The BBB is formed by specialized endothelial cells with tight junctions, minimal vesicular transport, and absence of fenestrations, collectively restricting passive diffusion of most compounds.

  This barrier protects neural tissue from toxins and maintains brain homeostasis, but also limits drug delivery to the brain. BBB permeability is modulated by interactions among endothelial cells, astrocytes, pericytes, and the extracellular matrix, which together constitute the neurovascular unit.
- `Mutag` (mutagenicity) (Sundar et al., 2018): Mutag refers to the induction of permanent transmissible changes in the amount or structure of the genetic material of cells or organisms.
- `DRD2` (dopamine receptor D2 binding affinity) (Fan et al., 2020): DRD2 measures the binding affinity of a molecule to the D2 subtype of dopamine receptors, which are G-protein-coupled receptors primarily located in brain regions such as the striatum, nucleus accumbens, and prefrontal cortex. These receptors are central to regulating reward, motivation, and motor control. Higher DRD2 affinity indicates stronger ligand-receptor binding, which can modulate dopaminergic signaling and is relevant for the treatment of neurological disorders such as Parkinson's disease.

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: **task**. Assistant:

Table 3. System prompt adopted for training. **task** will be replaced with the specific molecular optimization task.

Models	AddComponent		DelComponent		SubComponent	
	SR	Similarity	SR	Similarity	SR	Similarity
Baseline	0.086	0.763	0.107	0.864	0.057	0.815
Distill-SFT	0.100	0.604	0.188	0.682	0.078	0.633
SFT	0.238	0.619	0.203	0.755	0.366	0.721
GRPO	0.005	0.992	0.008	0.994	0.053	0.972
GRPO (SFT init)	0.246	0.635	0.232	0.759	0.420	0.713
DePO	0.307	0.778	0.158	0.887	0.429	0.802

Table 4. Performance comparison of various methods on structure-based optimization tasks from TOMG-Bench. For each task, the best result is bolded and the second best is underlined. We report Success Rate (SR) and Similarity; higher values indicate better performance.

Datasets. We detailed the dataset used in the experiments, including the construction of the dataset, the training splits.

- **TOMG-Bench** is derived from Zinc-250K (Sterling & Irwin, 2015) and PubChem (Kim et al., 2019), comprising two task categories: structure-based and single-property optimization. In structure-based tasks, the LLM is instructed to operate on specific functional groups within molecules. Single-property optimization tasks require the LLM to modify molecules to enhance target properties such as `QED` (Bickerton et al., 2012) (drug-likeness), `LogP` (Lipinski et al., 1997) (lipophilicity), and `MR` (Le Fevre, 1965) (molecular size and polarizability).
- **MuMOInstruct** is a multi-objective molecular optimization benchmark designed to reflect the complexity of real-world drug discovery. Derived from Zinc-250K, it requires models to optimize multiple molecular properties concurrently, thereby increasing task difficulty. It incorporates both seen and unseen instruction styles to evaluate the model's instruction-following robustness. The benchmark covers five critical pharmaceutical properties: `plogP` (lipophilicity, balancing permeability, solubility, and metabolic stability; higher is better), `QED` (drug-likeness), `BBBP` (blood-brain barrier permeability, relevant for central nervous system drugs), `Mutag` (mutagenicity, where lower values indicate reduced toxicity), and `DRD2` (dopamine receptor D2 binding affinity, with higher values indicating greater specificity).

For TOMG-Bench, we utilize the light training split, comprising 1,500 samples (500 per subtask for both structure-based and property-based optimization). The full TOMG-Bench test set is used for evaluation. To ensure comparability in training data volume, we randomly select 500 samples from MuMOInstruct for training, which resulting 1500 samples for training. All the training samples only contain the instruction and the target molecule, without any intermediate reasoning process.

Supervised Fine-Tuning. We configure the training process as follows. We employ the Llama-Factory (Zheng et al., 2024) to SFT the model. All the SFT models are trained using two A100 GPU. Each device processes a batch size of 2, and gradients are accumulated over 2 steps before an update. The learning rate is set to 1.0×10^{-5} and optimized using a cosine scheduler, with a warmup ratio of 0.05 to stabilize early training. The model is trained for 5 epochs using BF16 precision on the training split of TOMG-Bench and 1 epoch on the training split of MuMOInstruct.

Reinforcement Learning. We utilize the Transformer Reinforcement Learning (TRL) library (von Werra et al., 2020) for model training. All reinforcement learning approaches, including *GRPO*, *GRPO (SFT init)*, and *DePO*, are trained using a unified system prompt (see Table A), consistent with the DeepSeek-R1 protocol (Guo et al., 2025). Unless otherwise specified, we adopt the default TRL hyperparameters, with the following exceptions: the learning rate is set to 5.0×10^{-6}, and the maximum prompt length is limited to 256 tokens. We use a group size of 4 per input and a maximum completion length of 1024 tokens. Training is conducted for 1 epoch with a per-device batch size of 2 for training and 1 for evaluation. To ensure reproducibility, we fix the random seed to 42 and apply a warmup ratio of 0.1. Model generation is performed on a single GPU, hosted by vLLM (Kwon et al., 2023), while two additional GPUs are allocated for training.

Evaluation. We employ vLLM to host the model to accelerate the generation process. For all generation tasks, we set the

Models	QED		LogP		MR	
	SR	Similarity	SR	Similarity	SR	Similarity
Baseline	0.188	0.693	0.268	0.627	0.252	0.685
Distill-SFT	0.208	0.594	0.234	0.579	0.214	0.619
SFT	0.297	0.697	0.298	0.692	0.359	0.663
GRPO	0.138	0.889	0.379	0.806	0.214	0.880
GRPO (SFT init)	0.223	0.863	0.212	0.863	0.265	0.850
DePO	0.312	0.756	0.415	0.715	0.399	0.736

Table 5. Performance comparison of various methods on property-based optimization tasks from TOMG-Bench. For each task, the best result is bolded and the second best is underlined. We report Success Rate (SR) and Similarity; higher values indicate better performance.

Models	BDP		BDQ		BPQ	
	SR	Similarity	SR	Similarity	SR	Similarity
Baseline	0.052	0.149	0.034	0.117	0.052	0.194
Distill-SFT	0.078	0.207	0.022	0.106	0.064	0.165
SFT	**0.456**	0.390	**0.344**	0.321	**0.484**	0.327
GRPO	0.156	**0.759**	0.082	**0.479**	0.212	**0.567**
GRPO (SFT init)	0.088	0.141	0.022	0.045	0.056	0.085
DePO	<u>0.206</u>	<u>0.569</u>	<u>0.16</u>	<u>0.365</u>	<u>0.274</u>	<u>0.509</u>

Table 6. Performance on seen instruction on MuMOInstruct benchmark. The best result is bolded, and the second best is underlined. We report Success Rate (SR) and Similarity; higher values indicate better performance.

temperature to 0.75 and `top_p` to 0.85 to balance diversity and relevance in the generated outputs. We use a single beam (`num_beams = 1`) and limit the maximum number of new tokens to 512. These hyper-parameters are chosen to ensure consistent and controlled generation quality across experiments.

Licenses. The MuMOInstruct dataset is released under the MIT License. Qwen-2.5-3B-Instruct is distributed under the Qwen Research License Agreement. vLLM, TRL, and Llama-Factory are all licensed under Apache 2.0.

B. Full Experiment Results and Further Analysis

In this section, we provide the full results of all the experiments. Notably, for TOMG-Bench, we provide the full results for the structure-based optimization tasks in Table 4 and the property-based optimization tasks in Table 5. For MuMOInstruct, we provide the full results for the seen instruction in Table 6 and the unseen instruction in Table 7. We also provide a discussion on the infeasibility of GRPO with SFT initialization on the multi-objective tasks in Appendix B.2. Finally, we conduct the empirical analysis on the performance of domain-specific LMs in Appendix B.3.

B.1. Molecular Optimization Performance

Single-objective optimization tasks. Tables 4 and 5 present the performance of each model on the structure-based and property-based tasks of TOMG-Bench. Performance is measured using Success Rate (SR) and molecular Similarity. Several key patterns are observed:

- **DePO consistently achieves a strong trade-off between SR and molecular similarity across tasks.** In the AddComponent task (Table 4), DePO attains an SR of 0.307 and a similarity of 0.778. In QED optimization (Table 5), it leads with an SR of 0.312 and a similarity of 0.756. These results underscore DePO's capacity to generate molecules that are both successful in meeting task objectives and structurally faithful to the input.

- **SFT improves SR but sacrifices similarity.** Supervised Fine-Tuning (SFT) markedly increases SR relative to the baseline (e.g., from 0.057 to 0.366 for SubComponent in Table 4), but this improvement often comes at the expense of molecular similarity, which remains lower than that of DePO (e.g., SFT similarity of 0.721 vs. DePO's 0.802 for SubComponent).

  GRPO with SFT initialization can achieve competitive SRs in certain cases (e.g., 0.420 for SubComponent), but its similarity is less consistent (0.713 for SubComponent). Distill-SFT generally underperforms SFT in both SR and similarity.

Models	BDP		BDQ		BPQ	
	SR	Similarity	SR	Similarity	SR	Similarity
Baseline	0.052	0.143	0.042	0.104	0.050	0.130
Distill-SFT	0.016	0.099	0.020	0.077	0.050	0.143
SFT	**0.400**	0.409	**0.356**	0.299	**0.376**	0.277
GRPO	0.148	**0.727**	0.078	**0.457**	0.186	**0.573**
GRPO (SFT init)	0.092	0.147	0.026	0.058	0.042	0.063
DePO	<u>0.198</u>	<u>0.572</u>	<u>0.170</u>	<u>0.322</u>	<u>0.242</u>	<u>0.596</u>

Table 7. Performance on unseen instruction on MuMOInstruct benchmark. The best result is bolded, and the second best is underlined. We report Success Rate (SR) and Similarity; higher values indicate better performance.

Task type	Objective (↑)	Baseline	Distill-SFT	SFT	GRPO	GRPO (SFT init)	DePO
Seen Instruction	BDP	0.008	0.016	0.101	**0.118**	0.012	**0.117**
	BDQ	0.004	0.002	**0.089**	0.039	0.001	<u>0.058</u>
	BPQ	0.010	0.011	0.115	<u>0.120</u>	0.005	**0.139**
Unseen Instruction	BDP	0.007	0.002	0.081	<u>0.108</u>	0.014	**0.113**
	BDQ	0.004	0.002	**0.088**	0.036	0.002	<u>0.054</u>
	BPQ	0.006	0.007	0.104	<u>0.107</u>	0.003	**0.144**

Table 8. Overall Performance in MuMOInstruct benchmark with seen and unseen instructions. The best results for each task are bolded, and the second-best is underlined.

Task type	Objective (↑)	BioT5-base	MolT5-large	Baseline	SFT	GRPO	GRPO (SFT init)	DePO
Structure-based optimization	AddComponent	0.054	0.031	0.065	<u>0.147</u>	0.005	<u>0.156</u>	**0.239**
	DelComponent	0.027	0.027	0.092	<u>0.154</u>	0.008	**0.176**	<u>0.140</u>
	SubComponent	0.011	0.016	0.047	0.264	0.052	<u>0.300</u>	**0.344**
Property optimization	QED	0.080	0.055	0.130	0.207	0.123	<u>0.193</u>	**0.236**
	LogP	0.079	0.043	0.168	0.206	**0.305**	<u>0.183</u>	**0.297**
	MR	0.081	0.048	0.173	<u>0.238</u>	0.188	<u>0.225</u>	**0.293**

Table 9. Comparison of different methods on TOMG-Bench target on structural and property optimization. The best results for each task are bolded, and the second-best is underlined.

- **GRPO without SFT init preserves similarity but has low SR.** GRPO without SFT initialization adopts a conservative modification strategy, frequently yielding the highest similarity scores across tasks (e.g., >0.97 in structure-based tasks in Table 4).

  However, this preservation of structural integrity results in very low SRs for most structure-based tasks (e.g., 0.005 for AddComponent). GRPO does exhibit task-specific strengths, such as a high SR of 0.379 for LogP optimization.

Multi-objective optimization tasks. Tables 6 and 7 present the performance of each model on the MuMOInstruct benchmark, evaluating both instructions encountered during training and those not seen previously. Performance is measured using Success Rate (SR) and molecular Similarity. The results reveal several key patterns:

- **Clear trade-off exhibit between SR and Similarity is apparent across methods.** Notably, SFT often yields high SR, particularly on seen instructions (e.g., SR of 0.456 for BDP in Table 6), but typically results in lower molecular similarity (e.g., SFT Similarity scores in Table 6 are 0.390, 0.321, 0.327, while DePO's are 0.569, 0.365, 0.509). This suggests that SFT can aggressively modify molecules to meet property targets, sometimes at the expense of significant structural deviation.

- **DePO consistently demonstrates a more balanced performance profile.** While its standalone SR might occasionally be surpassed by SFT (e.g., the SR for SFT on BDQ with seen instruction is 0.344 vs DePO's 0.160 in Table 6), DePO generally maintains higher similarity scores than SFT (compare DePO and SFT in Table 6 and Table 7). This ability to achieve competitive SR while preserving structural similarity contributes to its strong performance in the combined metric (SR $\times$ Similarity) reported in Table 2.

- **The GRPO variants exhibit distinct behaviors.** GRPO without SFT initialization tends to preserve molecular structure effectively, achieving high similarity scores (e.g., GRPO Similarity for BDP seen is 0.759 in Table 6). However, its SR can be variable (e.g., SR of 0.156 for BDP with seen instruction vs 0.082 for BDQ with seen instruction in Table 6). Conversely, GRPO initialized with SFT performs poorly on the MuMOInstruct benchmark, with notably low SR and often low similarity, leading to very low scores (e.g., 0.012 for BDP with seen instruction).

B.2. GRPO with SFT Initialization cannot Generate Readable Outputs

While GRPO with SFT initialization demonstrates noteworthy performance on single-objective tasks (as detailed in Table 1), its efficacy significantly diminishes on the more complex multi-objective tasks within the MuMOInstruct benchmark. The combined SR $\times$ Similarity scores presented in Table 8 for GRPO (SFT init) are markedly low across all evaluated multi-objective settings.

This quantitative underperformance aligns with qualitative observations of problematic generation behavior, such as those illustrated in Section C.2, where the model may produce multiple, unreasoned molecular outputs or invalid SMILES strings. These issues suggest that while SFT initialization can be beneficial for simpler tasks, it may hinder the model's reasoning ability to effectively navigate the chemical space of multi-objective molecular optimization, leading to a failure to generate both valid and high-quality solutions.

B.3. Comparison with Domain-Specific LMs

We report the SR $\times$ Similarity scores for BioT5-base (Pei et al., 2023) and MolT5-large (Edwards et al., 2022) as provided in (Li et al., 2024a). BioT5 leverages biochemical text to enhance both molecular understanding and generation, while MolT5-large utilizes large-scale pretraining to improve SMILES generation from textual descriptions. We report the results in Table 9.

Notably, the results demonstrate that fine-tuned generalist language models can perform competitively, and often surpass, domain-specific models in molecular optimization tasks. Notably, DePO consistently outperforms both BioT5-base and MolT5-large across all evaluated objectives. For example, in QED optimization, DePO achieves a score of 0.236, substantially higher than BioT5-base (0.080) and MolT5-large (0.055). Moreover, the baseline generalist LLM, without additional task-specific fine-tuning, often matches or exceeds the performance of domain-specific models (e.g., Baseline LogP score of 0.168 vs. 0.079 for BioT5-base and 0.043 for MolT5-large).

These findings suggest that general-purpose LLMs, when adapted with DePO, are highly effective for molecular optimization and can match or outperform models pre-trained specifically on biomedical and chemical corpora.

C. Case Study

In this section, we provide case studies of different approaches to both single-objective and multi-objective optimization tasks.

C.1. Case Studies on Single-Objective Optimization

In this part, we present the case studies of different approaches to single-objective optimization tasks. In general, for the single-objective optimization task, the baseline model lacks detailed domain knowledge. SFT and GRPO (SFT init) offer simple modifications without elucidating the underlying chemical rationale, potentially sacrificing optimization opportunities for directness. GRPO, when unguided, is prone to errors in chemical understanding and reasoning.

Case study of Baseline model (Qwen-2.5-3B-Instruct).
Prompt: Modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR.
LLM Output:
I'm sorry, but it seems there might be some confusion in your request.
"MR" can refer to different things depending on the context (e.g., molecular formula, molar ratio, mass-to-charge ratio, etc.).
To modify a chemical structure, such as the given propane sulfonate derivative, we typically perform chemical modifications that aim to modify the functionality, functional groups, or introduce new ones.
However, without more specific instructions or the chemical context, it's challenging to propose a clear and direct modification that would result in a lower MR (assuming it refers to the Molecular Formula or molar mass of the compound).
If you could provide more details or specify what "lower MR" means in your case, I would be better able to assist you with a specific modification.
Alternatively, if you meant to work on the structural formula of the compound, I can help you customize it or identify potential modifications.

Case study of SFT.
Prompt: Modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR.
LLM Output:
Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1

Case study of GRPO (SFT init).
Prompt: Modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR.
LLM Output:
Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1O

Case study of GRPO.
Prompt: Modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR.
LLM Output:
To modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR value, we need to consider the structure of the molecule and the impact of the structural changes on the molecular properties.
The original molecule has a heterocyclic ring with a nitrogen atom (N) and a carbonyl group (C=O). The structure can be represented as follows: "' | | CC[N+]/CC "'
The MR or Magic Angle Spinning value is determined by the molecular structure and the presence of functional groups in the molecule. To lower the MR value, we can remove the nitrogen atom (N) from the heterocyclic ring, as the absence of the nitrogen atom can lead to a lower MR value.
The modified structure with the nitrogen atom removed can be represented as follows: "' | | CC[red]/CC "'
The removal of the nitrogen atom can lead to a significant decrease in the MR value, making the modified molecule more suitable for certain applications.
In summary, to lower the MR value of the molecule, we can remove the nitrogen atom from the heterocyclic ring.

Case study of DePO.
Prompt: Modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to have a lower MR.
LLM Output:
To determine how to modify the molecule Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br to achieve a lower Magnetization Transfer Ratio (MTR) value, we need to understand the impact of different chemical modifications on the MR value.
MTR is responsible for the transfer of magnetization between molecules, and such transfer can be influenced by several aspects of the molecule's structure, including steric hindrance, electronegativity, and polarizability.
First, we need to consider the structure of the molecule:
<think> Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br can be broken down into a carbonyl group attached to a benzene ring, which is then attached to a nitrogen atom, and connected to a bromine atom.
The presence of the bromine group can introduce steric hindrance and create a polarizable environment, which could impact the MR value. To decrease the MR value, we might consider removing any bulky or electronegative substituents, which would reduce the steric hindrance and permit more efficient interactions or resonance structures contributing to higher MR.</think>
One straightforward approach to do this is to modify the bromine atom to a more flexible or less electronegative substituent, such as hydrogen (H) or another non-polar atom or a less electronegative halogen like chlorine (Cl).
However, since chlorine is known to be more electronegative and might increase MR, we might want to opt for a less electronegative and flexible substituent.
<answer> Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Cl </answer>
Here, chlorine is substituted for the bromine, reducing the steric hindrance and potentially lowering the MR value by changing the electronegativity and flexibility of the molecule.
However, for more definitive optimization, computational methods (like molecular dynamics simulations, quantum chemical calculations, or a more extensive computational search) would be necessary.
This modification shows a reasonable effect based on the goal of lowering MR value.

DePO, in contrast, successfully combines guided exploration with preserved reasoning capabilities, leading to chemically sound analysis and a reasoned modification. This highlights DePO's ability to effectively navigate the complexities of molecular optimization by leveraging demonstrations to inform its search process, resulting in more reliable and interpretable outputs.

We summarize the key findings as follows:

The baseline model fails to understand the chemical context. The baseline model fails to understand the term "MR" in the chemical context, requesting clarification. This highlights a lack of specialized knowledge required for targeted molecular optimization, a common issue with general-purpose LLMs.

SFT provides a direct answer without intermediate reasoning. Model after SFT provides a direct answer by removing the bromine atom: `Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1`. While this modification is chemically valid and likely reduces MR (by removing a heavy atom), the output lacks any reasoning process. This aligns with Observation 3.2 from Section 3, which highlights that SFT can diminish reasoning ability, leading to shorter completions.

GRPO (SFT init) exhibits similar behavior as SFT. GRPO (SFT init) substitutes the bromine with an oxygen atom: `Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1O`. Similar to SFT, this is a direct modification without explicit reasoning. While potentially effective, it underscores Observation 3.3, indicating that GRPO may not fully recover the detailed reasoning capabilities when initialized from an SFT model that favors direct answers.

GRPO shows flawed chemical reasoning. GRPO misinterprets "MR" as "Magic Angle Spinning," incorrectly analyzes the molecular structure (e.g., its depiction of the molecule and the claim about the heterocyclic ring), and proposes a chemically implausible modification (removing a nitrogen atom from the heterocyclic ring). This behavior is consistent with Observation 3.1, where GRPO, without proper guidance, struggles to navigate the chemical space and may generate suboptimal or incorrect solutions. The output shown is directly referenced in Figure 5 (right) of Section 5.3.

DePO demonstrates a more robust and chemically sound approach. DePO demonstrates a more robust and chemically sound approach. The model correctly interprets the task in the context of MTR (Magnetization Transfer Ratio), identifies key structural features (bromine, carbonyl groups, nitrogen), and reasons about their impact on MR (steric hindrance, electronegativity). The DePO output shown is related to Figure 5 (left) of Section 5.3.

It proposes substituting bromine with chlorine (`Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Cl`), a chemically plausible modification aimed at reducing steric hindrance and altering electronegativity to lower MR. This case study aligns with the principles of DePO outlined in Section 4, where demonstration guidance helps constrain exploration to chemically valid and promising regions.

C.2. Case Study on Multi-Objective Optimization

In this part, we present the case studies of different approaches to multi-objective optimization tasks.

We summarize the key findings as follows:

Case study of Baseline model (Qwen-2.5-3B-Instruct).
Prompt: Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.
Input : <SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc
(OC)c(C)c1 </SMILES>
Adjust: increase BBB permeability, increase DRD2 inhibition and increase Penalized octanol-water partition coefficient (penalized logP)
LLM Output:
without large structural modifications.
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)Nc1ccc(C)nc1 </SMILES>

The baseline model struggles with complex instructions and multiple objectives. The baseline model generates a molecule by modifying the end part of the SMILES string: `CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)Nc1ccc(C)nc1`. While it produces a valid SMILES string, the output lacks explicit reasoning regarding how the modification addresses the three specified objectives (increasing BBB permeability, DRD2, and penalized logP). The effectiveness of the modification is unclear without analysis. This indicates a potential difficulty in handling multi-faceted optimization goals simultaneously and providing transparent reasoning.

Case study of SFT.
Prompt: Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.
Input : <SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc
(OC)c(C)c1 </SMILES>
Adjust: increase BBB permeability, increase DRD2 inhibition and increase Penalized octanol-water partition coefficient (penalized logP)
LLM Output:
without increasing liver toxicity.
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/C1CCN(C)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(C)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccccc2)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccccc2Cl)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccc(Cl)c(Cl)c2)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccccc2F)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccc(Cl)c(Cl)c2)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccc(Cl)c(Cl)c2)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc2ccc(Cl)c(Cl)c2)CC1
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N1CCN(Cc

Case study of GRPO (SFT init).
Prompt: Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.
Input : <SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc
(OC)c(C)c1 </SMILES>
Adjust: increase BBB permeability, increase DRD2 inhibition and increase Penalized octanol-water partition coefficient (penalized logP)
LLM Output:
without increasing hydrophobicity.
CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1
</CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Br)cc1

SFT generates multiple candidates without clear selection criteria or reasoning for multi-objective tasks. The SFT model outputs a list of different SMILES strings, such as `CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/C1CCN(C)CC1` and others involving substitutions like `N1CCN(Cc2ccccc2)CC1`. It prefaces its output with "without increasing liver toxicity," which was not part of the original prompt, suggesting potential misalignment or hallucination of constraints. Notably, the SFT model does not provide any reasoning for the modifications or how they address the multiple objectives of increasing BBB permeability, DRD2 inhibition, and penalized logP. This behavior aligns with Observation 3.2, where SFT models tend to provide direct answers, often at the cost of detailed reasoning, making it difficult to assess their strategy for balancing multiple objectives.

GRPO (SFT init) provides a single modification but lacks a detailed rationale for multi-objective trade-offs. The GRPO (SFT init) model suggests substituting a part of the molecule to `...N/N=C/c1ccc(Br)cc1`. While it offers a single, concrete modification, the output is repetitive and lacks the step-by-step reasoning required to understand how this change is intended to concurrently satisfy the three diverse optimization goals. This is consistent with Observation 3.3, indicating that GRPO, when starting from an SFT model, may not fully restore the capacity for elaborate reasoning, especially when dealing with the complexities of multi-objective optimization.

Case study of GRPO.
Prompt: Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.
Input : <SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc
(OC)c(C)c1 </SMILES>
Adjust: increase BBB permeability, increase DRD2 inhibition and increase Penalized octanol-water partition coefficient (penalized logP)
LLM Output:
without changing the molecular structure significantly.
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES> </SMILES>
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES> </SMILES>
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES> </SMILES>
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES> </SMILES> </SMILES>
Note: For the given properties and constraints, no SMILES alteration will change the molecule significantly without altering its structure. Thus, the SMILES structure remains the same. However, for illustration, 4 SMILES structures have been presented.
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES>
</SMILES>
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES>
</SMILES>
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H]
(O)N/N=C/c1ccc(OC)c(C)c1</SMILES>
</SMILES>
<SMILES>
CCOc1cc([C@H]2NC(=O)NC(C)=C2"

Case study of DePO.
Prompt: Your task is to modify the given molecule to adjust specific molecular properties while keeping structural changes as minimal as possible. Your response should only contain a valid SMILES representation of the modified molecule enclosed with <SMILES> </SMILES> tag.
Input : <SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc
(OC)c(C)c1 </SMILES>
Adjust: increase BBB permeability, increase DRD2 inhibition and increase Penalized octanol-water partition coefficient (penalized logP)
LLM Output:
without increasing lipophilicity (logP)
Adjusted SMILES:
<SMILES> CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Cl)c(Cl)c1 </SMILES>
Explanation:
- Penalized LogP Increase: The removal of the hydroxyl (-O-) group from one of the amide nitrogens (from CCCO to CCCN) increases the octanol-water partition coefficient, which is beneficial for BBB permeability and penalized logP. The change to an alkyl side chain (CCCNC1CCOc2ccc(Cl)c(Cl)c2) can also contribute to this effect.
- DRD2 Inhibition: No specific alterations are made to the scaffold that directly affect DRD2 inhibition.
- Lipophilicity (logP) Stability: The removal of the hydroxyl group does not increase logP significantly, which maintains lipophilicity while improving the desired properties.

GRPO fails to generate meaningful modifications in the multi-objective setting. When presented with this task, GRPO repeatedly outputs the original molecule, stating: "For the given properties and constraints, no SMILES alteration will change the molecule significantly without altering its structure. Thus, the SMILES structure remains the same." This behavior indicates that GRPO is unable to effectively engage with the optimization objective, likely due to limitations in its reward structure or an excessive preference for minimal structural changes. As highlighted in Observation 3.1, GRPO can struggle to explore chemical space without explicit guidance, often resulting in conservative outputs, particularly in complex multi-objective scenarios.

DePO exhibits systematic reasoning and targeted molecular modification for multi-objective optimization. In contrast, DePO proposes a modified molecule, `CCOc1cc([C@H]2NC(=O)NC(C)=C2C(=O)OC)ccc1OC[C@H](O)N/N=C/c1ccc(Cl)c(Cl)c1`, and provides a clear rationale for its design. The model explains its chemical modifications in complete sentences. For example, it states that the dichlorination of the terminal phenyl ring is intended to influence the desired properties. It also notes that the removal of the hydroxyl group from the molecule increases the octanol-water partition coefficient. This change is beneficial for both blood-brain barrier permeability and penalized logP.

Although DePO acknowledges that it did not make direct changes to improve DRD2 inhibition, it demonstrates an understanding of the multiple objectives and justifies its design choices accordingly. This structured and interpretable approach aligns with DePO's use of demonstration-based guidance, as described in Section 4. The effectiveness of this method is also evident in DePO's superior performance on multi-objective tasks, as shown in Table 2. In summary, DePO is better able to balance competing objectives and provide transparent and actionable outputs.