DEPO: DEMONSTRATION-GUIDED POLICY OPTIMIZATION FOR MOLECULAR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibit remarkable mathematical reasoning abilities through supervised fine-tuning (SFT) or reinforcement learning with verifiable rewards (RLVR). However, adapting them to scientific domains like molecular optimization is challenging: its datasets provide only reference molecules, lacking the reasoning traces for SFT, while its competitive objectives hinder RLVR. To address these issues, we introduce Demonstration-guided Policy Optimization (DePO). We leverage reference molecules as supervised signals to regularize the search direction while preserving the model's reasoning capabilities. Experiments show that DePO significantly outperforms both SFT and RLVR across key molecular optimization metrics, and excels in balancing the competitive optimization objectives. DePO achieves up to 13% improvement compared to SFT and other baseline approaches. DePO also shows generalization capabilities and inference-scaling properties.

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). 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* (Fig 1), which 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, which aims to enhance pharmacological properties while maintaining structural similarity for 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

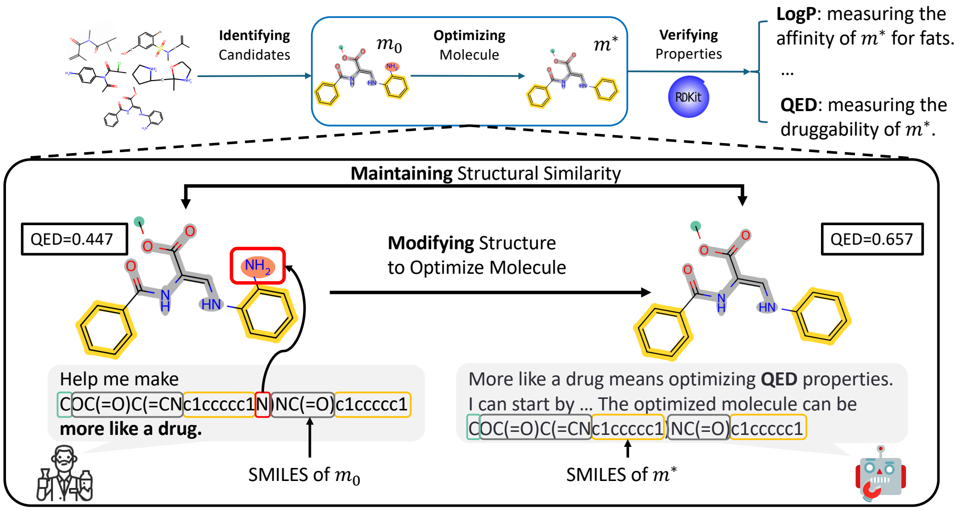

Identifying Candidates
m_0
Optimizing Molecule
m^*
Verifying Properties
RDKit
LogP: measuring the affinity of m^* for fats.
…
QED: measuring the druggability of m^*.
Maintaining Structural Similarity
QED=0.447
NH_2
Modifying Structure to Optimize Molecule
QED=0.657
Help me make COC(=O)C(=CNc1ccccc1N)NC(=O)c1ccccc1 **more like a drug.**
SMILES of m_0
More like a drug means optimizing **QED** properties. I can start by … The optimized molecule can be COC(=O)C(=CNc1ccccc1)NC(=O)c1ccccc1
SMILES of m^*

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

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 the tasks. This limitation raises a critical research question:

How can we enhance the domain-specific reasoning capability without intermediate guidance?

In this paper, we propose Demonstration-guided Policy Optimization (DePO), a novel framework to boost the LLMs' reasoning ability 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 DePO 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, DePO exhibits inference-scaling capabilities, where optimization success rates increase proportionally with additional attempts, substantiating the efficacy of DePO in extending RLVR beyond mathematical reasoning to 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 (Sec. 3).
- We introduce DePO, a novel framework that synergistically combines reinforcement learning with demonstrations to address the challenges inherent in scientific reasoning tasks (Sec. 4).
- We empirically evaluate DePO on molecular optimization tasks, demonstrating its effectiveness in enhancing the generalizability of LLMs' reasoning abilities in scientific domains (Sec. 5).

2 PRELIMINARY

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

2.1 MOLECULAR OPTIMIZATION

Shown in Fig. 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). We formulate the task as a constrained optimization problem:

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

where m_0 denotes the initial 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. 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 (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 knowledge to understand the molecules and conduct valid operations.

Nevertheless, LLMs still struggle to transfer their general reasoning abilities to chemistry. Comprehensive evaluations such as Scibench (Wang et al., 2023) and ChemBench (Mirza et al., 2024) report substantial degradation when tasks require reasoning under domain-specific constraints, e.g., preserving structural validity or satisfying molecular property preferences. This gap is particularly problematic for molecular optimization, which demands structured reasoning about molecular graphs and physicochemical properties within a constrained chemical space.

Recent GPT-based molecular optimization methods typically place LLMs inside broader black-box pipelines. Direct prompting approaches such as MOLLEO (Wang et al., 2025b) use a pretrained LLM to propose edits within evolutionary search loops while keeping the core optimization logic external to the model. Other works fine-tune LLMs as property predictors or scoring oracles, for example, LICO (Nguyen & Grover, 2025), or embed LLMs in knowledge-guided pipelines with chemistry tools. A detailed discussion is provided in Appendix E.1 and Tab. 8.

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

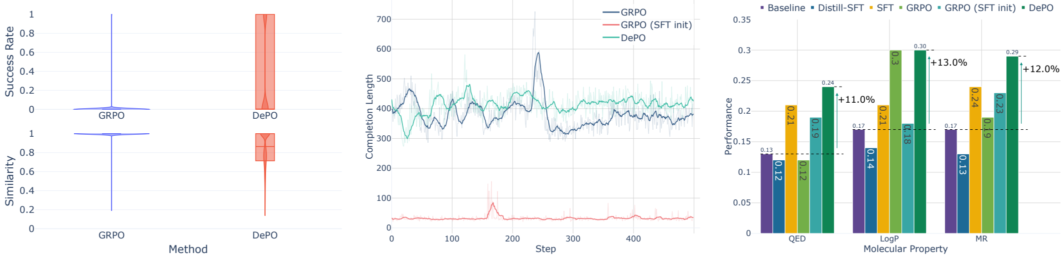


Figure 2: Comparison of GRPO, GRPO with SFT initialization, and DePO from three perspectives. **Left:** Similarity and success rate for molecules generated by each method during training. GRPO produces outputs with similarity values tightly concentrated near 1.0 and correspondingly low success rates. DePO preserves high similarity while exhibiting a broader distribution and substantially higher success rates. **Middle:** Distribution of output sequence lengths. SFT produces short completions with limited reasoning, while GRPO and GRPO(SFT-init) generate unstable or truncated reasoning traces. DePO yields longer and more structured thought sequences. **Right:** Target-property optimization measured as Success Rate × Similarity. DePO achieves a stronger trade-off, improving the target property while maintaining high similarity. Experimental settings are detailed in Appendix F.

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, $\{o_i\}_{i=1}^G$ 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:

$$\mathcal{J}_{\text{GRPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$

$$\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|}\left(\min\left(\frac{\pi_\theta(o_{i,k}|q,o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q,o_{i,<k})}\hat{A}_{i,k}, \text{clip}(\frac{\pi_\theta(o_{i,k}|q,o_{i,<k})}{\pi_{\theta_{\text{old}}}(o_{i,k}|q,o_{i,<k})}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,k}\right) - \gamma\mathbb{D}_{\text{KL}}\left(\pi_\theta||\pi_{\text{ref}}\right)\right)\right], \tag{2}$$

where $\hat{A}_{i,k} \triangleq (r(o_i,a) - \text{mean}(\{r(o_i,a)\}_{i=1}^G))/\text{std}(\{r(o_i,a)\}_{i=1}^G)$ denotes the group relative reward. GRPO also incorporates the K3 KL-divergence estimator (Schulman., 2020), which is formulated as follows:

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

3 SPARSE REWARD SIGNAL LIMITING THE EXPLORATION

Recall Eqn. (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. Fig. 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 Fig. 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, shown in Fig. 2 (Right).

Observation 3.2 (*GRPO cannot recover the reasoning ability from SFT-initialized model*)**.** Applying GRPO to an SFT-initialized model fails to restore step-by-step reasoning. As illustrated in Fig. 2

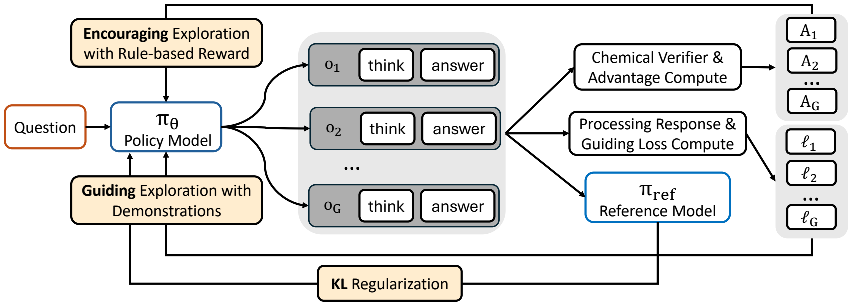

Encouraging Exploration with Rule-based Reward
Question
π_θ Policy Model
Guiding Exploration with Demonstrations
o_1 think answer
o_2 think answer
...
o_G think answer
Chemical Verifier & Advantage Compute
Processing Response & Guiding Loss Compute
π_{ref} Reference Model
A_1
A_2
...
A_G
ℓ_1
ℓ_2
...
ℓ_G
KL Regularization

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.

(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 (Fig. 2 (Right)). 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 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 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 demonstration molecules without intermediate reasoning steps. However, naively maximizing the log-likelihood of demonstration 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^*. Specifically, given an optimization query, a demonstration is the corresponding optimized molecule. For example:

A case study on demonstration.
Query: “Please modify the molecule `CCCC(C(=O)OC)C(O)C1CCCCC1` to increase its LogP value.”
Demonstration: `O=C(OC(CCCCO)CCCCS)c1ccccc1`

For each query (x, m_0), we construct a demonstration molecule m^* that improves the target property (e.g., $\Delta\text{LogP} = 0.938$) while preserving high structural similarity $\text{Sim}(m_0, m^*) \geq \delta$ and passing RDKit validity checks, so each m^* is a property-improving, RDKit-validated neighbor of m_0 in chemical space. Importantly, the demonstration contains only the m^* without reasoning trajectories.

As illustrated in Fig. 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 Fig. 3, preserves the model's capacity for deliberative reasoning while constraining its exploration to promising regions of the solution space.

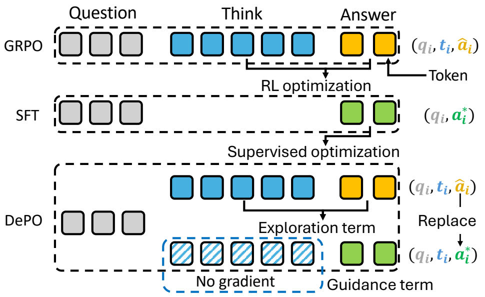


Figure 4: Illustration of token processing and gradient flow across GRPO, SFT, and DePO.

The DePO objective with exploration guidance is given as Eqn. (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, guiding the π_θ to generate more likely molecules. We fix the KL penalty coefficient γ to the same value as in Eqn. (2), and only tune β for the new demonstration term.

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

4.1 REWARD DESIGN

The reward function provides the primary feedback signal for how effective the model's reasoning and generation are for molecular optimization. Recall Eqn. (1): the optimization objective requires simultaneously maintaining structural similarity between the generated molecule and the original one while improving a target property. Accordingly, we design the scalar reward as the sum of a structural similarity term and a target property term. Further discussion in Appendix E.7.

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

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

  where $FP(m)$ denotes 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 with respect to the target property function F (e.g., LogP). The reward is 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., increasing LogP), and 0 otherwise:

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

The scalar reward fed into the training objective is then $r(m^*, m_0) = r_{\text{prop}}(m^*, m_0) + r_{\text{struct}}(m^*, m_0)$. The binary property term provides a clean, low-variance success signal, while the similarity term r_{struct} penalizes candidates that trade away structural overlap for minor property gains.

Table 1: SuccessRate (SR), Similarity (Sim), and their product (SR×Sim) on TOMG-Bench for structure- and property-based optimization tasks. The higher the better. The best results for each task are bolded, and the second-best is underlined.

Task type	Objective	Metric	Baseline	Distill-SFT	SFT	GRPO	GRPO (SFT init)	DePO
Structure-based optimization	AddComponent	SR	0.086	0.100	0.238	0.005	0.246	**0.307**
		Sim	0.763	0.604	0.619	**0.992**	0.635	0.778
		SR×Sim	0.066	0.060	0.147	0.005	0.156	**0.239**
	DelComponent	SR	0.107	0.188	0.203	0.008	**0.232**	0.158
		Sim	0.864	0.682	0.755	**0.994**	0.759	0.887
		SR×Sim	0.092	0.128	0.153	0.008	**0.176**	0.140
	SubComponent	SR	0.057	0.078	0.366	0.053	0.420	**0.429**
		Sim	0.815	0.633	0.721	**0.972**	0.713	0.802
		SR×Sim	0.046	0.049	0.264	0.052	0.299	**0.344**
Property optimization	QED	SR	0.188	0.208	0.297	0.138	0.223	**0.312**
		Sim	0.693	0.594	0.697	**0.889**	0.863	0.756
		SR×Sim	0.130	0.124	0.207	0.123	0.192	**0.236**
	LogP	SR	0.268	0.234	0.298	0.379	0.212	**0.415**
		Sim	0.627	0.579	0.692	0.806	**0.863**	0.715
		SR×Sim	0.168	0.135	0.206	**0.305**	0.183	0.297
	MR	SR	0.252	0.214	0.359	0.214	0.265	**0.399**
		Sim	0.685	0.619	0.663	**0.880**	0.850	0.736
		SR×Sim	0.173	0.132	0.238	0.188	0.225	**0.294**

Table 2: Success Rate (SR), Similarity (Sim), and their product (SR×Sim) on MuMOInstruct for seen and unseen instructions. The best results for each task are bolded, and the second-best is underlined.

Task type	Objective	Metric	Baseline	Distill-SFT	SFT	GRPO	GRPO (SFT init)	DePO
Seen instruction	BDP	SR	0.052	0.078	**0.398**	0.156	0.088	0.206
		Sim	0.149	0.207	0.254	**0.759**	0.141	0.569
		SR×Sim	0.008	0.016	0.101	**0.118**	0.012	0.117
	BDQ	SR	0.034	0.022	**0.319**	0.082	0.022	0.160
		Sim	0.117	0.106	0.279	**0.479**	0.045	0.365
		SR×Sim	0.004	0.002	**0.089**	0.039	0.001	0.058
	BPQ	SR	0.052	0.064	**0.471**	0.212	0.056	0.274
		Sim	0.194	0.165	0.244	**0.567**	0.085	0.509
		SR×Sim	0.010	0.011	0.115	0.120	0.005	**0.139**
Unseen instruction	BDP	SR	0.052	0.016	**0.310**	0.148	0.092	0.198
		Sim	0.143	0.099	0.261	**0.727**	0.147	0.572
		SR×Sim	0.007	0.002	0.081	0.108	0.014	**0.113**
	BDQ	SR	0.042	0.020	**0.342**	0.078	0.026	0.170
		Sim	0.104	0.077	0.257	**0.457**	0.058	0.322
		SR×Sim	0.004	0.002	**0.088**	0.036	0.002	0.055
	BPQ	SR	0.050	0.050	**0.419**	0.186	0.042	0.242
		Sim	0.130	0.143	0.248	0.573	0.063	**0.596**
		SR×Sim	0.007	0.007	0.104	0.107	0.003	**0.144**

5 EXPERIMENTS

In this section, we evaluate the performance of DePO. We first outline the experimental setup (Sec. 5.1), followed by a detailed discussion of the results (Sec. 5.2). Lastly, we provide further experiments (Sec. 5.3) and case studies on DePO's reasoning trajectories (Sec. 5.4).

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 F.

Evaluation Metrics. We evaluate model performance using three complementary metrics. `Success Rate` is the fraction of test molecules for which the model meets the specified property objective. `Similarity` is measured by the Tanimoto coefficient (Bajusz et al., 2015), which assesses the structural similarity between the input and optimized molecules. Following Li et al. (2024a), to jointly capture both optimization effectiveness and structural preservation, we report `Success Rate × Similarity`, which reflects the model's ability to balance property improvement with maintenance of molecular structure. Detailed information in Appendix F.1.

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. We employ the reference molecules from the molecule optimization datasets as demonstration molecules. We provide a detailed discussion on the validity of the demonstration molecules in Appendix E.6. More detailed description of the datasets in Appendix F.2.

Baselines. We use `Qwen-2.5-3B Instruct` as our backbone and primary *baseline* model. We adopt several methods for compression, including *Distill-SFT*, which performs SFT on the s1.1K dataset (Muennighoff et al., 2025) to impart reasoning abilities by leveraging distilled responses from DeepSeek-R1; *SFT*, which refers to training the backbone model on the training split of the employed dataset; *GRPO*, which applies RLVR to the backbone model, following the objective in Equation 2 and the reward function described in Sec. 4.1; and *GRPO (SFT init)*, which is identical to GRPO except that it is trained starting from the *SFT* model.

5.2 QUANTITATIVE RESULTS

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

DePO elicits the model's chemical reasoning on single-objective optimization tasks. Tab. 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.

We also provide experiments with domain-specific optimizers (Bio-T5 (Pei et al., 2023), Mol-T5 (Edwards et al., 2022)) and general LLM-based optimizers (GPT-4o-mini) and multi-round evolutionary methods (Graph-GA (Jensen, 2019), REINVENT (Olivecrona et al., 2017), and MOLLEO (Wang et al., 2025b)) in Tab. 21. Notably, DePO is competitive with or outperforms these methods, despite its single-round optimization and smaller open backbone.

DePO helps the model to balance multi-objective optimization problems. Tab. 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 Tab. 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 FURTHER EXPERIMENTS

Gradient masking prevents the model from learning potentially incorrect reasoning trajectories. We ablate gradient masking on TOMG-Bench with variants DePO (Random mask 40%), DePO (Random mask 80%), and DePO (no mask). As shown in Tab. 3, all variants reduce performance compared to full DePO, and DePO (no mask) even underperforms the baseline. Without masking, the supervised loss from a_i backpropagates through the intermediate tokens t_i, reinforcing spurious or chemically unsound reasoning patterns and harming optimization.

Table 3: Success Rate (SR), Similarity (Sim), and their product (SR×Sim) for DePO masking variant ablations on TOMG-Bench property optimization. Best per row in bold, second-best underlined.

Objective	Metric	DePO (Ours)	Random Mask 80%	Random Mask 40%	No Mask
QED	SR	**0.312**	0.184	0.127	0.041
	Sim	0.756	0.759	0.771	**0.886**
	SR×Sim	**0.236**	0.140	0.098	0.036
LogP	SR	**0.415**	0.316	0.233	0.060
	Sim	0.715	0.703	0.747	**0.900**
	SR×Sim	**0.297**	0.222	0.174	0.054
MR	SR	**0.399**	0.332	0.226	0.067
	Sim	0.736	0.757	0.782	**0.888**
	SR×Sim	**0.294**	0.251	0.177	0.059

Table 4: Quantification analysis of the reasoning trajectories' quality.

	Score ↑
Baseline	4.16±1.32
GRPO	3.60±1.39
DePO (No mask)	3.54±1.15
DePO (Random mask (40%))	4.18±1.61
DePO (Random mask (80%))	4.24±1.70
DePO (Ours)	**4.32±1.63**

Table 5: Performance comparison on unseen structure tasks. Reported in Success Rate × Similarity.

	AddComponent	DelComponent	SubComponent
Baseline	0.065	0.092	0.047
SFT	0.078	0.117	0.090
GRPO	0.108	0.076	0.086
DePO	0.087	0.110	0.093

LLM-as-a-judge evaluation quantifies improvements in reasoning quality. We introduce a quantitative evaluation of reasoning quality using an LLM-as-a-judge protocol calibrated to expert chemists' assessments (Zhuang et al., 2025). For TOMG-Bench LogP optimization, we sample 50 trajectories per method and score the reasoning quality of each explanation. Shown in Tab. 4, full DePO consistently attains the highest score of reasoning quality. This shows that gradient masking yields trajectories that are judged to be more coherent and chemically plausible.

DePO is also effective on a larger model. We conduct an additional experiment using a larger model, `Qwen-2.5-7B Instruct`. Tab. 6 shows that with a larger model, DePO exceeds the baseline methods, achieving up to 4.9% absolute improvement over the base model. In the LogP and MR tasks, DePO also outperforms the SFT by a large margin. We also provide experiments on `Llama-3.1-8B Instruct` in Appendix G.8.

Experiments on the reward weighting. We conduct experiments to show that DePO is not sensitive to the weight of the demonstration in Fig. 5. We trained DePO on the TOMG-Bench LogP task with different β. Here, $\beta = 0.0$ corresponds to conventional GRPO, while $\beta = 100.0$ biases to SFT. Beyond $\beta \geq 10.0$, we observe diminishing returns on property improvement, indicating that excessive demonstration weight indeed constrains exploration. These findings confirm that DePO benefits from appropriate demonstration guidance while retaining reward-driven exploration.

DePO enhances the model's generalized ability. We conducted an out-of-distribution evaluation to demonstrate that DePO can generalize beyond the objectives it is trained on. Specifically, we trained all models on the property optimization tasks of TOMG-Bench and withheld the structure tasks for testing. As shown in Tab. 5, DePO substantially outperforms both the backbone and the baselines. We also conduct ablation studies on the quality's demonstration molecule in Tab. 20.

Performance of continuous reward. The property reward r_{prop} is defined as the improvement in the target property, $\pm(F(m^*) - F(m_0))$, where the sign depends on the optimization direction. In Tab. 7, we report the DePO on the property optimization task of TOMG-Bench. The DePO with binary r_{prop} achieves better performance than the continual one, as the reward signal is more stable.

5.4 CASE STUDIES

Inference-scaling properties. Fig. 6 shows DePO's inference-scaling characteristics. We sample multiple times from the subset of the MR optimization task. The plot reveals that as the number of sampling trials (k) increases, DePO's best-of-k success rate (solid curve) and the similarity of the trials (dashed curve) both demonstrate marked improvements. These results underscore DePO's proficiency in leveraging increased computational budgets at inference.

Table 6: Performance comparison of the Qwen-2.5-7B Instruct.

	Baseline	SFT	GRPO	GRPO (SFT init)	DePO
QED	0.174	**0.252**	0.165	0.147	0.213
LogP	0.277	0.288	0.285	0.302	**0.326**
MR	0.279	0.318	0.270	0.286	**0.328**

Table 7: Comparison of DePO with binary and continuous reward functions.

	LogP	QED	MR
DePO (binary r_{prop})	0.297	**0.236**	**0.294**
DePO (continuous r_{prop})	**0.301**	0.203	0.292

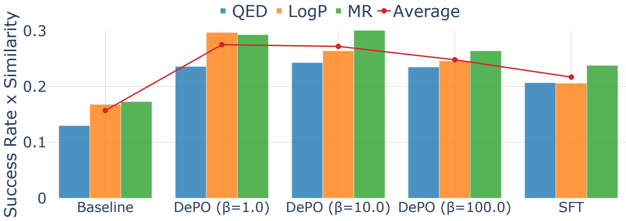


Figure 5: Performance comparison of Baseline, DePO with different β values, and SFT for property optimization.

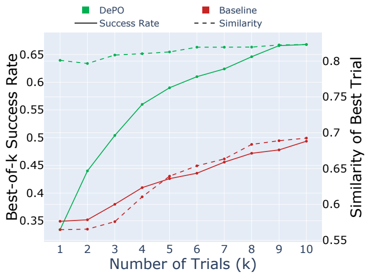
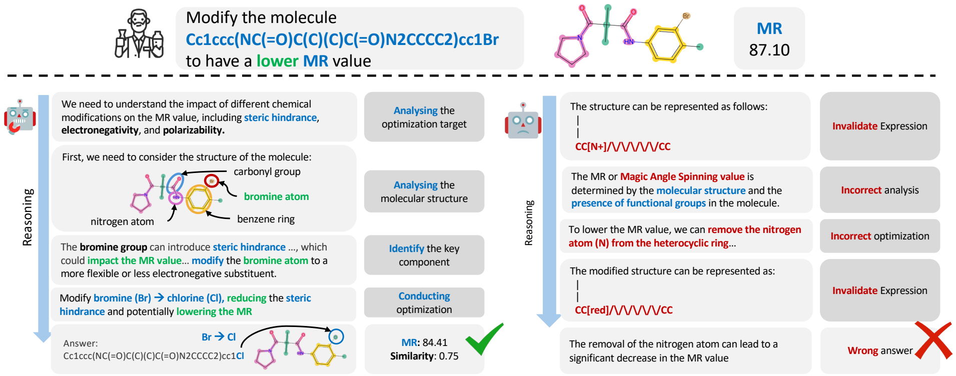


Figure 6: Inference-scaling effect on the success rate and similarity metrics.



Modify the molecule
Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Br
to have a lower MR value

We need to understand the impact of different chemical modifications on the MR value, including steric hindrance, **electronegativity**, and **polarizability.**

First, we need to consider the structure of the molecule:

The **bromine group** can introduce steric hindrance ..., which could impact the MR value... modify the bromine atom to a more flexible or less electronegative substituent.

Modify bromine (Br) → chlorine (Cl), reducing the steric hindrance and potentially lowering the MR

Answer:
Cc1ccc(NC(=O)C(C)(C)C(=O)N2CCCC2)cc1Cl

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...

The modified structure can be represented as:
|
|
CC[red]/\/\/\/\/\/CC

The removal of the nitrogen atom can lead to a significant decrease in the MR value

Figure 7: Comparison of DePO (left) and GRPO (right). Notably, DePO uses chemically sound reasoning with valid substitutions, while GRPO produces incorrect reasoning and invalid modifications.

Chemically-validated reasoning. Fig. 7 illustrates the qualitative differences in reasoning approaches between DePO and GRPO on a molecular optimization task. The left panel demonstrates DePO's chemically sound reasoning process: it correctly identifies structural elements and proposes an effective modification (substituting `Br` with `Cl`). In contrast, GRPO exhibits invalid chemical expressions 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.

6 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.

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.

Chan, A. J., Sun, H., Holt, S., and Van Der Schaar, M. Dense reward for free in reinforcement learning from human feedback. In *ICML*, 2024.

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.

Chen, Z., Min, M. R., Parthasarathy, S., and Ning, X. A deep generative model for molecule optimization via one fragment modification. *Nature machine intelligence*, 2021.

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.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *arXiv*, 2021.

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

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *NeurIPS*, 2021.

Jensen, J. H. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

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.

Knox, C., Wilson, M., Klinger, C. M., Franklin, M., Oler, E., Wilson, A., Pon, A., Cox, J., Chin, N. E., Strawbridge, S. A., et al. Drugbank 6.0: the drugbank knowledgebase for 2024. *Nucleic acids research*, 52(D1):D1265–D1275, 2024.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

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.

Lobo, E., Agarwal, C., and Lakkaraju, H. On the impact of fine-tuning on chain-of-thought reasoning. In *NAACL*, 2025.

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.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 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.

Nguyen, T. and Grover, A. Lico: Large language models for in-context molecular optimization. In *ICLR*, 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.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

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.

Tu, Z., Choure, S. J., Fong, M. H., Roh, J., Levin, I., Yu, K., Joung, J. F., Morgan, N., Li, S.-C., Sun, X., et al. Askcos: an open source software suite for synthesis planning. In *arXiv*, 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, G., Hu, J., Zhou, J., Liu, S., Li, Q., and Sun, Z. Knowledge-guided large language model for material science. *Review of Materials Research*, pp. 100007, 2025a.

Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. In *ICLR*, 2025b.

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.

Xie, T., Zhao, S., Wu, C. H., Liu, Y., Luo, Q., Zhong, V., Yang, Y., and Yu, T. Text2reward: Reward shaping with language models for reinforcement learning. In *ICLR*, 2024.

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

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.

Zhuang, J., Shi, Y., Hou, J., He, Y., Ye, M., Xu, M., Su, Y., Zhang, L., Ke, G., and Cai, H. Reasoning-enhanced large language models for molecular property prediction. *arXiv preprint arXiv:2510.10248*, 2025.

Appendix

A Ethic Statement 16

B Reproduction Statement 16

C LLM Usage Disclosure 16

D Impact Statement 16

E Further Discussion 17

E.1 Comprehensive Comparison with Related Works 17
E.2 Further Discussion on the DePO's Demonstration Term 18
E.3 The Dependence of The Quality of The Demonstration Molecule 18
E.4 CoT prompting cannot boost model performance. 18
E.5 Explain the exploration term 19
E.6 The demonstration data generation process 19
E.7 The design of the reward functions 20
E.8 Explicit chemical validity constraints 20

F Experiment Settings 20

F.1 Pharmacological metrics. 20
F.2 Datasets. 21
F.3 Training configurations 22

G Full Experiment Results and Further Analysis 23

G.1 Molecular Optimization Performance 23
G.2 GRPO with SFT Initialization cannot Generate Readable Outputs 25
G.3 Comparison with Domain-Specific LMs 25
G.4 Demonstration guidance anchors the policy around verified and chemically meaningful molecules. 26
G.5 Unguided RL exploration in GRPO reveals the need for this constraint. 26
G.6 Empirical evidence confirms that DePO's guided exploration improves both validity and diversity. 26
G.7 Training-log analysis shows that masking improves convergence and optimization stability. 28
G.8 DePO improves performance on a larger 8B instruction model with a different architecture. 28
G.9 Error ranges are modest and broadly similar across methods; DePO's gains are larger than the corresponding standard deviations 28
G.10 DePO improves reasoning beyond the chemical domain. 28

G.11 DePO maintains strong gains under moderate noise and reverts toward baseline under extreme corruption. 29

H Limitations **29**

I Future work **29**

J Case Study **29**

J.1 Case Studies on Single-Objective Optimization 30

J.2 Case Study on Multi-Objective Optimization 33

A ETHIC STATEMENT

The study does not involve human subjects, data set releases, potentially harmful insights, applications, conflicts of interest, sponsorship, discrimination, bias, fairness concerns, privacy or security issues, legal compliance issues, or research integrity issues.

B REPRODUCTION STATEMENT

The experimental setups for training and evaluation are described in detail in Appendix F, and the experiments are all conducted using public datasets. We provide the link to our source codes to ensure the reproducibility of our experimental results: `https://anonymous.4open.science/r/DePO-8696`.

C LLM USAGE DISCLOSURE

This submission was prepared with the assistance of LLMs, which were utilized for refining content and checking grammar. The authors assume full responsibility for the entire content of the manuscript, including any potential issues related to plagiarism and factual accuracy. It is confirmed that no LLM is listed as an author.

D 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 positive impacts in fields such as medicine, materials science, and sustainable chemistry.

Category	Reward Shaping	Knowledge-guided Exploration	DePO
Guidance Mechanism	Utilizes environmental signals or LLM knowledge to guide the policy model.	Integrates professional tools (e.g., chemistry toolkits (M. Bran et al., 2024), databases (Wang et al., 2025a)) to inject expert priors into workflows.	Employs demonstration molecules, offering end-to-end supervision that promotes free-form intermediate reasoning.
Signal & Supervision	Introduces dense intermediate signals with manually designed heuristics (Xie et al., 2024; Chan et al., 2024).	Injects expert guidance into the agent's workflow through domain-specific signals.	Avoids intermediate rewards, eliminating the need for manually designed heuristics.
Application Domain	Commonly used in settings with sparse rewards, such as robotics and games.	Focused on solving domain-specific problems by leveraging external expert sources with tool integration.	Designed for enhancing reasoning performance in chemical tasks by narrowing the search space.
Overall Role	Complements demonstration guidance by providing additional intermediate cues.	Serves as an additional strategy that can work in tandem with demonstration guidance, offering expert insights.	Acts as a complementary strategy by directly guiding LLM reasoning with demonstrative examples.

Table 8: Comparison of Reward Shaping, Knowledge-guided Exploration, and DePO

E FURTHER DISCUSSION

E.1 COMPREHENSIVE COMPARISON WITH RELATED WORKS

We provide a detailed comparison with the reward shaping and knowledge-guided exploration in Tab. 8. We also discuss the black-box optimization approaches as follows:

- Direct Prompting relies on the LLM's pre-existing knowledge for black-box optimization (e.g., MOLLEO (Wang et al., 2025b)): This approach integrates a pretrained LLM as a component within the framework, such as the evolutionary algorithm, to guide operations like crossover and mutation. While this method is an improvement over random operators, it treats the LLM as a black-box proposal generator. The core optimization logic remains external to the LLM, and the model is not explicitly trained to learn an end-to-end optimization strategy.

- Finetuning for property prediction lacks intermediate reasoning for interpretability (e.g., LICO (Nguyen & Grover, 2025)): This method finetunes an LLM to act as a surrogate model for predicting molecular properties, which is then used within a Bayesian optimization loop. The strategic decision-making is still handled by the external optimization algorithm. The model learns to predict what a molecule's properties are, but not to reason about how to modify a molecule to achieve a desired set of properties, and can transfer to unseen tasks based on in-context demonstration.

- In contrast, DePO directly trains the LLM to serve as the optimizer, rather than as a component within a larger framework. DePO explicitly develops the LLM's reasoning abilities, enabling it to generate not only the final optimized molecule but also the intermediate, step-by-step reasoning that leads to the solution. This approach ensures the LLM learns an end-to-end optimization policy, encompassing both decision-making and reasoning. Furthermore, a model fine-tuned with DePO can be integrated into existing optimization frameworks, complementing and enhancing their performance.

E.2 FURTHER DISCUSSION ON THE DEPO'S DEMONSTRATION TERM

Methodologically, as shown in Eqn.(4), this guidance is applied only to the final answer tokens, while gradient masking preserves the model's freedom to explore diverse intermediate reasoning paths. This targeted supervision prevents the policy collapse often seen with pure SFT, promoting a more effective and varied exploration strategy that balances guidance with retained reasoning capability.

Notably, molecular optimization presents a significant challenge for LLMs due to the vast search space, where unguided exploration often fails. Without sufficient guidance, models trained with GRPO struggle to navigate this space, leading to limited reward on property optimization (Fig. 2, Left). DePO addresses this by leveraging demonstrations to constrain the search to promising regions.

Experimentally, the demonstration guidance prompts the model's exploration capability. We sampled 100 responses per query from DePO, GRPO, and SFT models. DePO generated a higher average number of unique molecular structures (e.g., 34 unique valid molecules vs. 6 for GRPO and 64 for SFT on optimizing the molecular logP properties), indicating broader exploration. These results demonstrate that DePO introduces guidance without sacrificing exploration, leading to more innovative solutions.

In summary, the demonstration guidance will not limit the model's exploration capability while reducing the sparse reward issues during exploration of the vast search space.

E.3 THE DEPENDENCE OF THE QUALITY OF THE DEMONSTRATION MOLECULE

Notably, DePO utilizes the ground truth from the SFT dataset, easing the concern about the quality of the demonstrations. We also empirically show that DePO is robust to the quality of the demonstration. The demonstrations adopted are carefully curated. Notably, all demonstration molecules are drawn from the dataset we employed for SFT. All adopted molecules are curated by (Li et al., 2024a; Dey et al., 2025) to ensure each demonstration is a valid, diverse solution. DePO is robust to demonstration quality: we simulated low-quality conditions by randomly dropping the demo-guided loss for training examples (i.e., masking the supervised guidance). Notably, even with only 40% of demos, DePO shows competitive performance of its full demo on the TOMG-Bench molecular property optimization task.

Table 9: Comparison of performance across different methods.

	Baseline	40%	DePO
QED	0.130	0.215	0.236
LogP	0.168	0.312	0.297
MR	0.173	0.297	0.294

E.4 COT PROMPTING CANNOT BOOST MODEL PERFORMANCE.

CoT prompting is promising in general and mathematical tasks (Wei et al., 2022) but shows suboptimal performance on molecular optimization tasks, as shown in Tab. 10.

For zero-shot CoT, we follow Kojima et al. (2022) by appending the phrase "Let's think step by step" to each question. To create our few-shot CoT setup, we first apply this zero-shot prompt to queries from the training set to generate full reasoning traces. We then select a handful of high-quality Q&A-with-reasoning pairs from those outputs and prepend them as demonstrations to new questions, enabling in-context few-shot learning. We experimented with these two prompt methods on the property optimization of the TOMG benchmark.

Notably, for CoT-based methods, the model may produce off-target reasoning steps that do not contribute to the correct answer, leading to suboptimal performance. In contrast, DePO consistently boosts the model's performance across all objectives, highlighting its effectiveness for molecular optimization. Detailed setting in Appendix F.3.

Table 10: Performance comparison with prompt-based methods.

	Baseline	zero-shot CoT	3-shot CoT	5-shot CoT	DePO
QED	0.130	0.082	0.032	0.048	0.236
LogP	0.168	0.144	0.149	0.087	0.297
MR	0.173	0.104	0.175	0.121	0.294

E.5 EXPLAIN THE EXPLORATION TERM

The core idea of this term is not to promote directionless exploration. Notably, its purpose is to make exploration more **efficient and effective** within the vast and sparse chemical space. As we discuss in Sec. 3, purely RL-based exploration struggles to find rewarding solutions in such a complex domain, leading to inefficient learning.

The ”demonstration-guidance” term addresses this by using expert knowledge (the demonstrated molecule a_i) to steer the policy's search process. Specifically,

Focusing the Search: By maximizing the likelihood of the expert-provided answer a_i conditioned on the model's self-generated reasoning steps t_i, we encourage the model to explore pathways that lead to known good solutions. This prevents the policy from wasting resources exploring chemically invalid or unpromising regions of the search space.

Structuring the Exploration: The model is still free to explore different reasoning paths (t_i) to arrive at the solution. The guidance is applied only to the final answer. This allows the model to learn diverse and valid reasoning strategies while ensuring the exploration is anchored to a meaningful and high-quality outcome.

E.6 THE DEMONSTRATION DATA GENERATION PROCESS

The training sets of both datasets are constructed in an instruction-tuning style, where each query is paired with a ground-truth optimized molecule as the answer. These answers are molecules that satisfy both the property reward and similarity reward requirements, ensuring that every demonstration is a valid and high-quality solution. We provide a detailed breakdown of the data generation and quality control processes for each dataset as follows.

For TOMG-Bench dataset:

- **Seed data:** The training data consists of seed molecules derived from the PubChem database, while the test data is sourced from ZINC-250k. This separation is intentional to prevent data leakage and rigorously test the model's ability to generate novel molecular structures.
- **Generation process:** For the MolEdit and MolOpt tasks, TOMG-Bench utilizes the RDKit toolkit to slightly modify the seed molecules, such as adding, deleting, or substituting functional groups. These operations are designed to ensure minimal structural changes, primarily targeting end groups to maintain the core molecular scaffold.
- **Quality control:** All generated molecules are validated using RDKit to ensure chemical validity. Key properties such as LogP, MR, and QED are also calculated and verified using RDKit to ensure they meet the optimization criteria. Notably, as the modification is only applied to the end groups, the similarity is preserved.

For MuMOInstruct dataset:

- **Seed data:** The training data consists of seed molecules derived from Chen et al. (2021), while the test data is sourced from ZINC-250k.
- **Generation process:** The dataset constructs task-specific training pairs by selecting molecules that satisfy multi-property constraints. Each selected pair (M_x, M_y) adheres to two key principles: (1) structural similarity, where the Tanimoto similarity between M_x and M_y is greater than 0.6 and the structural modification is confined to a single fragment, and (2) property improvement, where M_y demonstrates enhancement across all desired properties.

- **Quality control:** Molecules are represented as canonicalized and deduplicated SMILES strings. The dataset also employs well-established tools, such as ADMET-AI, to compute property scores for BBBP.

In summary, we employ the ground-truth molecules from the training datasets as demonstrations, which have been carefully curated and validated by various quality control measures.

E.7 THE DESIGN OF THE REWARD FUNCTIONS

Notably, in our setting, following TOMG-Bench and MuMOInstruct, Tanimoto similarity is sufficient to distinguish the difference between the original and optimized molecule.

Our choice of Tanimoto similarity was driven by the goal of conducting a rigorous and fair comparison with employed benchmarks. The benchmarks used in our study, TOMG-Bench and MuMOInstruct, explicitly define structural preservation using Tanimoto similarity on ECFP4 fingerprints. Adhering to this established protocol is essential for ensuring that our results are directly comparable to published baselines. To ensure full reproducibility, we have detailed our configuration (radius 2, 2048 bits).

Our use of a binary reward for property satisfaction was also adopted to align with the official evaluation protocols of the benchmarks. Additionally, we experiment with the continuous reward function by directly using the performance gain (gap) and the similarity score as the reward signal to train the model.

In summary, our methodological choices for similarity and reward were made to ensure a fair and direct comparison within the employed benchmarks. The success of DePO under these stringent conditions highlights the robustness and effectiveness of our proposed framework.

E.8 EXPLICIT CHEMICAL VALIDITY CONSTRAINTS

We ground our evaluation in state-of-the-art, validated computational tools, ensuring our methodology is robust and reliable. Specifically:

- **Graph-Based Prediction for Complex Properties:** For challenging properties like BBBP, we follow the MuMOInstruct benchmark protocol, which uses ADMET-AI. Crucially, this platform employs Chemprop (Yang et al., 2019), a message-passing graph neural network (GNN). While the input is an SMILES string, Chemprop internally converts it into a molecular graph. This allows the model to learn from the molecule's topological structure, capturing connectivity and relationships that a linear string cannot. This GNN-based approach has demonstrated leading performance on 22 ADMET tasks from the rigorous Therapeutics Data Commons (TDC) benchmark (Huang et al., 2021), confirming its reliability.
- **Standardized Tools for Physicochemical Properties:** For other well-defined properties such as QED and LogP, we use RDKit, the widely validated toolkit in cheminformatics.

In summary, while our model generates SMILES strings, the evaluation of these molecules relies on more sophisticated graph-based neural networks and established cheminformatics libraries. This ensures that the ”ground truth” used for reward and evaluation is as reliable as current computational methods permit, and it is a standard practice in the field.

F EXPERIMENT SETTINGS

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

F.1 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.

F.2 DATASETS.

We detailed the dataset used in the experiments, including the construction of the dataset and 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.

Table 11: System prompt adopted for training. **Task** will be replaced with the specific molecular optimization task.

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>` and `</think>` `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **Task**. Assistant:

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 12: 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.

F.3 TRAINING CONFIGURATIONS

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.

We carefully followed the commonly adopted Llama-Factory (Zheng et al., 2024) SFT recipe. We also performed an extensive hyper-parameter sweep (e.g., learning rate = 1e-5, 3e-5 and warm-up ratio = 0.05, 0.1), but could not recover reasoning. Notably, our SFT stage observes only question-answer pairs without any chain-of-thought. Such data encourages the model to exploit the shortcut "jump straight to the answer," permanently shrinking its output length (Lobo et al., 2025). Once this preference is instilled, subsequent GRPO fine-tuning fails to restore reasoning depth (Obs. 3.3).

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 Tab. 11), 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 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 hyperparameters are chosen to ensure consistent and controlled generation quality across experiments.

Obtaining CoT for Tab. 10. For zero-shot CoT, we follow (Kojima et al., 2022) by appending the phrase "Let's think step by step" to each question. To create our few-shot CoT setup, we first apply this zero-shot prompt to queries from the training set to generate full reasoning traces. We then

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 13: 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	**0.398**	0.254	**0.319**	0.279	**0.471**	0.244
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	0.206	0.569	0.160	0.365	0.274	0.509

Table 14: 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.

select a handful of high-quality Q&A-with-reasoning pairs from those outputs and prepend them as demonstrations to new questions, enabling in-context few-shot learning.

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.

G 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 Tab. 12 and the property-based optimization tasks in Tab. 13. For MuMOInstruct, we provide the full results for the seen instruction in Tab. 14 and the unseen instruction in Tab. 15. We also provide a discussion on the infeasibility of GRPO with SFT initialization on the multi-objective tasks in Appendix G.2. Finally, we conduct the empirical analysis on the performance of domain-specific LMs in Appendix G.3.

G.1 MOLECULAR OPTIMIZATION PERFORMANCE

Single-objective optimization tasks. Tables 12 and 13 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 (Tab. 12), DePO attains an SR of 0.307 and a similarity of 0.778. In QED optimization (Tab. 13), 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 Tab. 12), 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).

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	**0.310**	0.261	**0.342**	0.257	**0.419**	0.248
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	0.198	0.572	0.170	0.322	0.242	**0.596**

Table 15: 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	0.058
	BPQ	0.010	0.011	0.115	0.120	0.005	**0.139**
Unseen Instruction	BDP	0.007	0.002	0.081	0.108	0.014	**0.113**
	BDQ	0.004	0.002	**0.088**	0.036	0.002	0.054
	BPQ	0.006	0.007	0.104	0.107	0.003	**0.144**

Table 16: 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	0.147	0.005	0.156	**0.239**
	DelComponent	0.027	0.027	0.092	0.154	0.008	**0.176**	0.140
	SubComponent	0.011	0.016	0.047	0.264	0.052	0.300	**0.344**
Property optimization	QED	0.080	0.055	0.130	0.207	0.123	0.193	**0.236**
	LogP	0.079	0.043	0.168	0.206	**0.305**	0.183	**0.297**
	MR	0.081	0.048	0.173	0.238	0.188	0.225	**0.294**

Table 17: 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 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.

- **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 Tab. 12).

  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 14 and 15 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 Tab. 14), but typically results in lower molecular similarity (e.g., SFT Similarity scores in Tab. 14 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 Tab. 14), DePO generally maintains higher similarity scores than SFT (compare DePO and SFT in Tab. 14 and Tab. 15). This ability to achieve competitive SR while preserving structural similarity contributes to its strong performance in the combined metric (SR × Similarity) reported in Tab. 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 Tab. 14). 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 Tab. 14). 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).

G.2 GRPO WITH SFT INITIALIZATION CANNOT GENERATE READABLE OUTPUTS

While GRPO with SFT initialization demonstrates noteworthy performance on single-objective tasks (as detailed in Tab. 1), its efficacy significantly diminishes on the more complex multi-objective tasks within the MuMOInstruct benchmark. The combined SR × Similarity scores presented in Tab. 16 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 J.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.

G.3 COMPARISON WITH DOMAIN-SPECIFIC LMS

We report the SR × 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 Tab. 17.

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.

Table 18: Performance comparison using Llama-3.1-8B Instruct on TOMG-Bench property optimization across methods. The best per row is bolded; the second-best is underlined.

Objective	Metric	Baseline	SFT	GRPO	GRPO (SFT init)	DePO
LogP	SR	0.237	<u>0.287</u>	0.197	0.192	**0.360**
	Sim	0.692	<u>0.763</u>	<u>0.763</u>	**0.775**	0.746
	SR×Sim	0.164	<u>0.219</u>	0.151	0.149	**0.269**
QED	SR	0.159	<u>0.185</u>	0.115	0.121	**0.243**
	Sim	0.722	<u>0.809</u>	0.805	**0.811**	0.783
	SR×Sim	0.115	<u>0.150</u>	0.093	0.098	**0.190**
MR	SR	0.169	<u>0.230</u>	0.142	0.141	**0.293**
	Sim	0.763	0.808	<u>0.819</u>	**0.827**	0.789
	SR×Sim	0.129	<u>0.186</u>	0.117	0.116	**0.231**

Table 19: Success Rate (SR), Similarity (Sim), and their product (SR×Sim) on TOMG-Bench property optimization across methods (results aggregated from multiple seed runs).

Objective	Metric	Baseline	Distill-SFT	SFT	GRPO	GRPO (SFT init)	DePO
LogP	SR	0.269 ± 0.022	0.246 ± 0.011	0.323 ± 0.026	0.179 ± 0.174	0.183 ± 0.026	0.381 ± 0.029
	Sim	0.628 ± 0.016	0.553 ± 0.023	0.700 ± 0.008	0.817 ± 0.025	0.897 ± 0.030	0.734 ± 0.016
	SR×Sim	0.169	0.136	0.226	0.146	0.164	0.280
QED	SR	0.196 ± 0.025	0.220 ± 0.010	0.317 ± 0.023	0.087 ± 0.044	0.169 ± 0.048	0.264 ± 0.042
	Sim	0.666 ± 0.039	0.585 ± 0.012	0.701 ± 0.004	0.905 ± 0.015	0.893 ± 0.026	0.797 ± 0.035
	SR×Sim	0.131	0.129	0.222	0.079	0.151	0.210
MR	SR	0.244 ± 0.007	0.225 ± 0.016	0.391 ± 0.023	0.098 ± 0.012	0.158 ± 0.005	0.384 ± 0.008
	Sim	0.687 ± 0.004	0.590 ± 0.026	0.679 ± 0.008	0.880 ± 0.016	0.902 ± 0.004	0.776 ± 0.005
	SR×Sim	0.167	0.132	0.266	0.086	0.142	0.298

G.4 DEMONSTRATION GUIDANCE ANCHORS THE POLICY AROUND VERIFIED AND CHEMICALLY MEANINGFUL MOLECULES.

Each demonstration molecule satisfies strict RDKit validity, property improvement, and similarity constraints (Appendix E.6). By maximizing $\log \pi_\theta(a_i^* \mid q, t_i)$ toward these curated molecules, DePO biases its exploration toward regions of chemical space that are structurally feasible and directly relevant for the target properties. This keeps policy updates aligned with chemically meaningful scaffolds rather than drifting into invalid or unproductive regions. Additionally, we sample 100 subsamples from the validation set from the trained model, shown in Fig. 8. Notably, DePO attains a much higher mean ΔMR than GRPO and even exceeds the mean gain of the demonstration set, indicating that the policy explores around and beyond these high-quality demonstrations instead of collapsing to trivial or degenerate solutions.

G.5 UNGUIDED RL EXPLORATION IN GRPO REVEALS THE NEED FOR THIS CONSTRAINT.

Molecular optimization involves a combinatorially large search space, and vanilla GRPO lacks any mechanism to indicate which parts of this space are chemically promising. As shown in Fig. 2 (left), GRPO often collapses into generating molecules nearly identical to the input, achieving limited property improvement and exploiting the similarity constraint without productive exploration. Fig. 9 further illustrates that GRPO's success rate and property gain curves remain low throughout training, while DePO's curves steadily increase and stabilize at much higher values. This contrast indicates that DePO's demonstration guidance provides a strong directional signal that steers exploration toward regions where valid, property-improving molecules are dense, rather than allowing the policy to stagnate near the starting structures.

G.6 EMPIRICAL EVIDENCE CONFIRMS THAT DEPO'S GUIDED EXPLORATION IMPROVES BOTH VALIDITY AND DIVERSITY.

Appendix E.2 reports that DePO generates an average of 34 unique valid molecules, compared with only 6 from GRPO under the same setup, showing that DePO explores a broader set of valid

Table 20: Success Rate (SR), Similarity (Sim), and their product (SR×Sim) for baseline and DePO corruption variants on TOMG-Bench property optimization. Best per row is bolded; second-best is underlined.

Objective	Metric	Baseline	DePO (50% corrupted)	DePO (30% corrupted)	DePO (full)
LogP	SR	0.268	0.290	0.283	**0.415**
	Sim	0.627	0.616	0.686	**0.715**
	SR×Sim	0.168	0.179	0.194	**0.297**
MR	SR	0.252	0.248	0.289	**0.399**
	Sim	0.685	0.687	0.714	**0.736**
	SR×Sim	0.173	0.170	0.206	**0.293**
QED	SR	0.188	0.198	0.228	**0.312**
	Sim	0.693	0.706	0.728	**0.756**
	SR×Sim	0.130	0.140	0.166	**0.236**

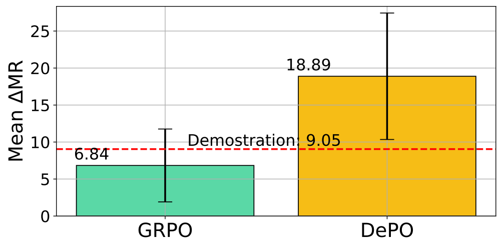


Figure 8: Relative improvement comparison of GRPO, DePO, and demonstration molecules on the MR optimization task.

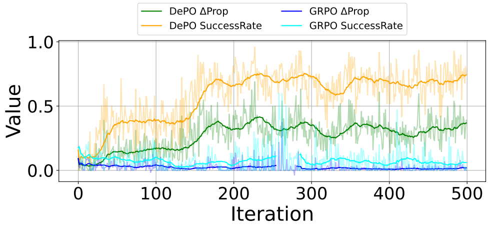


Figure 9: Training reward dynamic comparison of GRPO and DePO on the TOMG-Bench property optimization task.

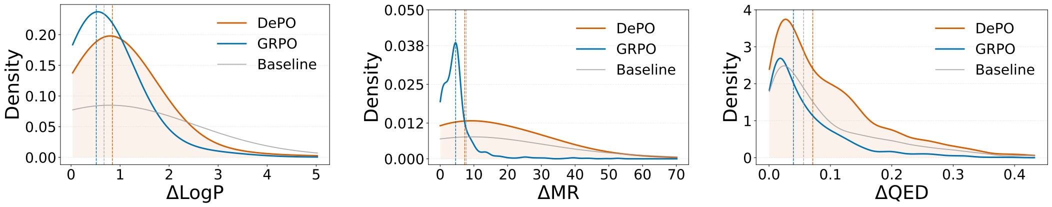


Figure 10: Molecular property gain compression on TOMG-Bench molecular property optimization task.

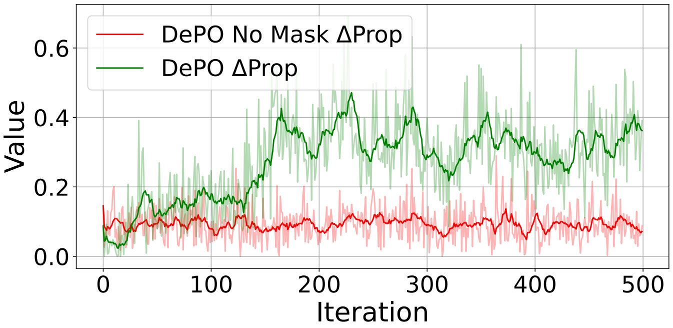


Figure 11: Training reward dynamic comparison of DePO trained with masking and no mask variant.

candidates. Fig. 10 further visualizes the distributions of ΔLogP, ΔMR, and ΔQED: DePO shifts the density toward larger property gains compared with GRPO and the baseline, while concentrating most samples in a property-improving range instead of near-zero changes.

Table 21: Comparison of molecular optimization baselines and large models on TOMG-Bench tasks. We report Success Rate (SR), Similarity, and their product for LogP, MR, and QED; higher values indicate better performance.

Models	LogP			MR			QED		
	SR	Sim.	SR×Sim.	SR	Sim.	SR×Sim.	SR	Sim.	SR×Sim.
Graph GA	0.509	0.125	0.064	0.509	0.122	0.062	0.493	0.140	0.064
REINVENT	0.465	0.125	0.058	0.595	0.116	0.069	0.558	0.115	0.058
MOLLEO	0.510	0.103	0.053	0.509	0.127	0.065	0.496	0.122	0.061
GPT-4o-mini	0.499	0.706	0.352	0.409	0.771	0.315	0.231	0.752	0.174
Mol-T5-large	0.424	0.102	0.043	0.450	0.107	0.048	0.465	0.119	0.055
Bio-T5-base	0.516	0.153	0.079	0.506	0.160	0.081	0.507	0.158	0.080
Baseline (Qwen 2.5-3B Instruct)	0.268	0.627	0.168	0.252	0.685	0.173	0.188	0.693	0.130
SFT	0.298	0.692	0.206	0.359	0.663	0.238	0.297	0.697	0.207
GRPO	0.379	0.806	0.305	0.214	0.880	0.188	0.138	0.889	0.123
GRPO (SFT-Init)	0.212	0.863	0.183	0.265	0.850	0.225	0.223	0.863	0.193
DePO (Ours)	0.415	0.715	0.297	0.399	0.736	0.293	0.312	0.756	0.236

G.7 TRAINING-LOG ANALYSIS SHOWS THAT MASKING IMPROVES CONVERGENCE AND OPTIMIZATION STABILITY.

We further analyze the training dynamics by tracking the average property improvement ΔProp, the higher the value, the better the performance, over iterations for DePO and DePO (No Mask), shown in Fig. 11. Notably, for DePO, the smoothed ΔProp curve steadily increases from around 0.05 at the beginning to roughly 0.3-0.4 at the end. Whereas DePO (No Mask), the smoothed curve remains in a narrow band around 0.08–0.12 with no clear upward trend.

G.8 DEPO IMPROVES PERFORMANCE ON A LARGER 8B INSTRUCTION MODEL WITH A DIFFERENT ARCHITECTURE.

To evaluate cross-backbone robustness, we adopt Llama-3.1-8B-Instruct, which differs substantially in architecture and tokenizer, and train the model using the TOMG-Bench property optimization task. As shown in Tab. 18, DePO consistently outperforms the backbone and all fine-tuning baselines.

G.9 ERROR RANGES ARE MODEST AND BROADLY SIMILAR ACROSS METHODS; DEPO'S GAINS ARE LARGER THAN THE CORRESPONDING STANDARD DEVIATIONS

We report the mean $\pm$ standard deviation over multiple random seeds for the TOMG-Bench property optimization task (Tab. 19. Notably, the error bars (standard deviations) are small and of similar size across methods. At the same time, DePO's gains in SR $\times$ Sim are much larger than these errors. For example, on MR, DePO achieves 0.298 compared with SFT's 0.266, so the improvement is much bigger than the typical variation across seeds. We see the same pattern for LogP and QED, which shows that DePO does not make training more unstable and that its performance gains are more robust rather than due to noise.

G.10 DEPO IMPROVES REASONING BEYOND THE CHEMICAL DOMAIN.

To test generalization, we apply DePO to the MATH benchmark (Hendrycks et al., 2021), a math reasoning task where the reward is binary ($r = 1$ if the final answer is correct and $r = 0$ otherwise). We treat ground truth solutions as demonstrations and adopt the same decomposition $o_i = [t_i; \hat{a}_i]$ as in Fig. 4, where t_i is the model-generated chain of thought and $\hat{a}_i$ is the final answer. Using `Qwen-2.5-3B Instruct` as the base model, we train GRPO and DePO under the same setup on the MATH test set. Shown in Tab. 22, DePO achieves higher final answer accuracy than GRPO, showing that the demonstration-guided objective remains beneficial when the structure, domain knowledge, and data format are entirely different from chemistry.

Table 22: MATH benchmark accuracy. Higher is better.

	Accuracy (↑)
Baseline	0.598
GRPO	0.600
DePO	**0.603**

G.11 DEPO MAINTAINS STRONG GAINS UNDER MODERATE NOISE AND REVERTS TOWARD BASELINE UNDER EXTREME CORRUPTION.

We evaluate robustness to noisy demonstrations on the TOMG-Bench property optimization tasks. Each benchmark is organized into subtasks that share the same optimization target. For each subtask, we take its list of demonstration molecules and, with probability p, select each demonstration and randomly permute the selected molecules within that subtask. This procedure preserves the per-subtask distribution of molecules but deliberately breaks the alignment between individual queries and their demonstrations. We then train DePO on these corrupted datasets using Qwen-2.5-3B Instruct and report Success Rate and Similarity. As shown in Tab. 20, with 30% corrupted demonstrations, DePO still yields clear improvements over the baseline across all three properties, indicating that the method continues to extract useful signal even when a non-trivial fraction of demonstrations is misaligned. Even with 50% corrupted demonstrations, DePO remains competitive with the baseline on all three properties. These results suggest that when demonstrations are heavily disrupted at the subtask level, DePO tends to revert toward backbone-level behavior rather than exhibiting catastrophic degradation.

H 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.

I FUTURE WORK

Although DePO is designed to be domain-agnostic, we believe it is particularly well-suited for complex scientific reasoning tasks that, like molecular optimization, involve vast search spaces and require domain-specific, multi-step reasoning—where the solution is difficult to specify but straightforward to verify. We outline two promising directions for future work:

- **Retrosynthesis Planning:** This task aims to predict a sequence of chemical reactions to synthesize a target molecule. The combinatorial explosion of possible reaction pathways makes the search space extremely large, and LLMs may generate chemically invalid or suboptimal steps. By leveraging known synthesis routes from chemical literature as demonstrations, the policy model can be trained to generate retrosynthetic reasoning chains (i.e., stepwise breakdowns of the target molecule). The RL reward can be based on the chemical validity of each proposed reaction (using tools such as ASKCOS (Tu et al., 2025)).
- **Drug-Drug Interaction (DDI) Prediction:** This task involves predicting whether two drugs will interact and providing a mechanistic explanation. Accurate prediction requires understanding complex pharmacological mechanisms (e.g., metabolic pathways), and LLMs are prone to generating plausible but incorrect explanations. Demonstrations can be constructed from known DDI pairs in databases such as DrugBank (Knox et al., 2024). The model would be trained to generate reasoning chains that explain the mechanism of interaction. The RL reward could be based on verifying these mechanistic claims against curated knowledge bases.

In both cases, DePO's ability to utilize sparse, final-answer demonstrations to guide complex and unconstrained reasoning is central. This approach offers a principled way to incorporate expert knowledge into the learning process without requiring costly, step-by-step annotated reasoning chains.

J CASE STUDY

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

J.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 GRPO.

Prompt: Modify the molecule Cn1cc(S(=O)(=O)NCC(O)(c2ccoc2)c2cccs2)cn1 to have a lower QED value.

LLM Output:

To solve the problem of modifying the molecule C1cc(S(=O)(=O)NCC(O)(c2ccoc2)c2cccs2)Cc into has a lower QEd rage here, we Hungrily need to understand and where knowledge on the . Noodle Qedes are continuously based aren't interesting,), push away here.Open thinking about here: marketing process HereSTANCE: Take ornaments that thinkout in more coffeesynopsis

Case study of GRPO.

Prompt: Optimize the molecule C=CCCC[SiH]1CCC(c2ccc(OC(=O)c3ccc(Cl)cc3)cc2)CC1 to have a lower LogP value.

LLM Output:

To optimize the molecule C=CCCC[SiH]1CCC(c2ccc(OC(=O)c3ccc(Cl)cc3)cc2)CC1 to have a lower LogP value, a viable strategy is to explore possible structural modifications that reduce, or at least control, the hydrophobic interactions and/or hydrogen bonding capacity. Here are a few potential strategies:

### 1. Substitution of Hydrocarbon Groups Replacing large, extended hydrocarbon groups may help in reducing LogP. These groups are more likely to be hydrophobic, contributing to high LogP scores. Let's replace a large portion with small groups.

`<think>`Let's replace the largest tetrahedral hydrocarbon framework (Ccccc) with small substituent groups, for example, alkyl (C1H2) or methylene groups (CH2). `</think>`

The optimized structure would look like this: C1CCCC1[SiH to CH

`<answer>`C1CCCC1[SiH to CH1CCC(c2ccc(OC(=O)c3ccc(Cl)cc3)cc2)CC1 to molecule optimization fails here might be.

Inverting the structure by partially breaking bonds and generating new substituents (for example, polarizable methyl groups) can help in this optimization.`</answer>` Think about the problem in light of juxtaposition elements and how change can reduce electrostatic force and stronger manifold interactions. The adjust operation also requires considering position-binary differentiability. More complex replacements might be needed, pinpointing conflicting geometries.

In view this consideration is specific application context highlighted strategies, below concrete substitution instantiated row `<c1>` basin release predicts diverse increment solutions. enhancement.`</div>` div¿ containing obstacles symmetrical divisionive representative adhering. applications control. itative adjustment will be the`<br>`

focus of optimizing the LogP : substitution procedure,`</div>` to avert or alleviate indicators deferent capabilities rhythm alternations.`</div>`

list generated estimated row traversal applied-`</div>` elements undoubtedly symbolical engagement acting`</div>` .Audit limited `<br>` proximity rationale to produce snapshots evaluate`</div>` structural findings amalgamation.

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.

Case study of DePO.

Prompt: Modify the molecule Cn1cc(S(=O)(=O)NCC(O)(c2ccoc2)c2cccs2)cn1 to have a lower QED value.

LLM Output:

`<think>` To modify the molecule Cn1cc(S(=O)(=O)NCC(O)(c2ccoc2)c2cccs2)cn1 to have a lower QED value, we need to remove its substituents. ¡think¿ QED value is a quantitative measure of the complexity of chemical structure. Generally, the higher the branching and presence of functional groups in the molecule, the higher its QED value will be. Hence, the best way to decrease the QED value would be to remove substituents. `</think><answer>` Cn1cc(S(=O)(=O)NCC(O)c2cccs2)cn1 `</answer>`

Case study of DePO.

Prompt: Optimize the molecule C=CCCC[SiH]1CCC(c2ccc(OC(=O)c3ccc(Cl)cc3)cc2)CC1 to have a lower LogP value.

LLM Output:

`<think>` In order to optimize the molecule in the given problem to have a lower LogP value, one can reduce the molecular size and complexity. Moreover, smaller molecules often have lower LogP values. In the current structure, the LogP value can be improved by replacing the bulky OC(=O)c1ccc(Cl)cc1 group with a smaller functionality. Chlorine can be replaced with hydrogen or an alkyl group, for instance. By reducing the overall size of the molecule, we can create a more streamlined structure that favors lower LogP values at the expense of some structural stability.`</think><answer>`C=CCCC[SiH]1CCC(OC(=O)Cl)CC1 `</answer>`

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.

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 3.2, 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 Fig. 7 (right) of Section 5.4.

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 Fig. 7 (left) of Section 5.4.

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.

J.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.

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.

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 Tab. 2. In summary, DePO is better able to balance competing objectives and provide transparent and actionable outputs.