AEMO: Atom-Level Editing for Molecular Optimization via Relative Policy Learning in Large Language Models

Anonymous ACL submission

Abstract

Molecular optimization aims to improve target properties through controllable structural modifications while preserving the core scaffold of the initial molecule. However, most methods based on Large Language Models (LLMs) formulate optimization as end-to-end SMILES generation, introducing a fundamental structural mismatch. It forces models to approximate local structural edits via global sequence generation, diminishes control over fine-grained structural changes, and implicitly reduces optimization to black-box search. To address these challenges, we propose **Atom-level Editing for Molecular Optimization (AEMO)**, a new paradigm that formulates molecular optimization as a discrete decision process over an atom-level editing action space. By utilizing Group-Relative Policy Optimization (GRPO) to compare parallel edit trajectories based on atom-level decision units, AEMO reinforces efficient structural modifications and establishes an explicit causal link between local edits and global property changes. Experiments on multiple single-property optimization tasks show that AEMO achieves state-of-the-art (SOTA) performance while maintaining high structural similarity.

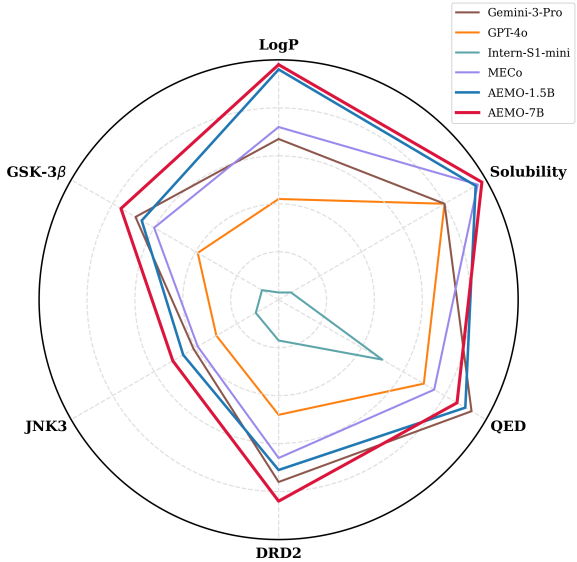

Figure 1: **Success rate comparison across single-objective tasks.** Compared to direct SMILES generation baselines, AEMO achieves broader task coverage, confirming the advantage of atom-level discrete decisions in overcoming structural mismatch.

1 Introduction

Drug discovery is computationally expensive and resource-intensive due to the large size of chemical space (Bleicher et al., 2003; Gómez-Bombarelli et al., 2018). Consequently, molecular optimization has become an important approach by avoiding exploration of chemical space from scratch and instead refining existing hit compounds (Hughes et al., 2011). Starting from a given initial molecule, it aims to enhance target properties through controlled and incremental structural edits while preserving chemical validity and structural similarity (Zhou et al., 2019; Jin et al., 2018b). In particular, it focuses on how local structural changes affect global properties, requiring fine-grained structural perception and reliable structural modification. Meanwhile, the large number of possible modifications leads to a vast search space, making efficient and controllable optimization challenging.

Recently, Large Language Models (LLMs) have been increasingly applied to molecular optimization for their capacity to capture structured knowledge and conditional dependencies from large-scale corpora (Fang et al., 2023; Yu et al., 2024). Prior work predominantly formulates molecular optimization as conditional SMILES generation (Bagal et al., 2021), often enhanced by instruction tuning. However, these methods are sensitive to data scale and coverage (Xu et al., 2026), frequently generating invalid or chemically implausible SMILES under Out-Of-Distribution (OOD) objectives or strict structural constraints. While some methods (Wang et al., 2024; Ran et al., 2025)

integrate LLMs into evolutionary algorithms by replacing genetic operators to improve search efficiency, they still require numerous iterations and oracle calls (Gao et al., 2022). Recent works decouple planning and execution to improve controllability (Zhu et al., 2025; Yang et al., 2025). However, their learning signals do not provide comparable relative feedback across edits within a shared editing space, making it difficult to capture implicit Structure–Activity Relationships (SAR).

At a fundamental level, these limitations arise from a structural mismatch between the requirements of molecular optimization and prevailing modeling paradigms. In practice, molecular optimization relies on precise local structural edits under fixed scaffolds. From a representation perspective, the inherently linearized nature of SMILES forces models to approximate local structural modifications through global sequence rearrangements (Flam-Shepherd et al., 2022). Consequently, the end-to-end generation lacks explicit intermediate decision states that reflect the evolving molecular structures during editing, which often leads to structural drift, obscures the causal link between edits and property changes, and weakens fine-grained structural control. From an optimization perspective, most training signals capture only overall property improvements, without differentiating the relative merits of local edits under a fixed scaffold. This coarse supervision effectively reduces molecular optimization to a black-box search process, making it difficult for models to learn SAR. These challenges call for a new paradigm that anchors optimization to molecular structure, treats local edits as explicit decision units, and learns relative merits at the editing level.

To this end, we propose **Atom-level Editing for Molecular Optimization (AEMO)**, a new molecular optimization paradigm based on an atom-level discrete action space and relative policy optimization. Rather than directly generating SMILES, we reinterpret molecules as graphs and introduce an atom-level action space (Li and Lai, 2025), enabling LLMs to operate on discrete atom-level actions as fundamental decision units. By decoupling reasoning and planning from structural execution, our method mitigates SMILES non-locality and achieves fine-grained edits. To address the inability of learning implicit SAR, we further adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to drive policy evolution. Through intra-group comparisons over diverse edit trajectories, the model learns to reinforce efficient edits where small structural changes yield large property gains (Stumpfe and Bajorath, 2012). As shown in Figure 1, with only 1.5B parameters, our method outperforms state-of-the-art (SOTA) models on multiple single-property optimization tasks, including Gemini-3-Pro (Gemini Team, 2025). This highlights that well-designed action spaces and efficient policy search enable small models to tackle challenging molecular optimization tasks.

Our contributions are summarized as follows:

- We propose AEMO, a new paradigm that formulates molecular optimization as a discrete decision process over an atom-level editing action space.
- We enable the model to explicitly attribute global property improvements to specific local atom-level edits by employing group-relative learning over parallel edit trajectories.
- Experiments demonstrate that AEMO significantly improves target properties while mitigating structural drift with 1.5B parameters. It outperforms SOTA models when scaled to 7B, demonstrating high parameter efficiency and strong scalability.

2 Related Works

Conventional methods for molecular design To efficiently explore the vast chemical space, early methods relied on rule-based search and heuristic exploration, such as ChemTS (Yang et al., 2017) and graph-based genetic algorithms (Jensen, 2019), which iteratively apply predefined transformations to generate candidate molecules. With the rise of deep learning, generative models based on SMILES or graph representations have become dominant, including junction-tree VAEs (Jin et al., 2018a), sequence-based models (Gómez-Bombarelli et al., 2018), and latent optimization frameworks such as LIMO (Eckmann et al., 2022). Reinforcement learning has also been adopted to guide molecular generation with property-based rewards (Olivecrona et al., 2017; Popova et al., 2018). Despite their success, these approaches often require complex pipelines and extensive oracle evaluations, leading to high computational cost and limited scalability.

LLMs for molecular optimization Recently, LLMs have emerged as a promising paradigm for

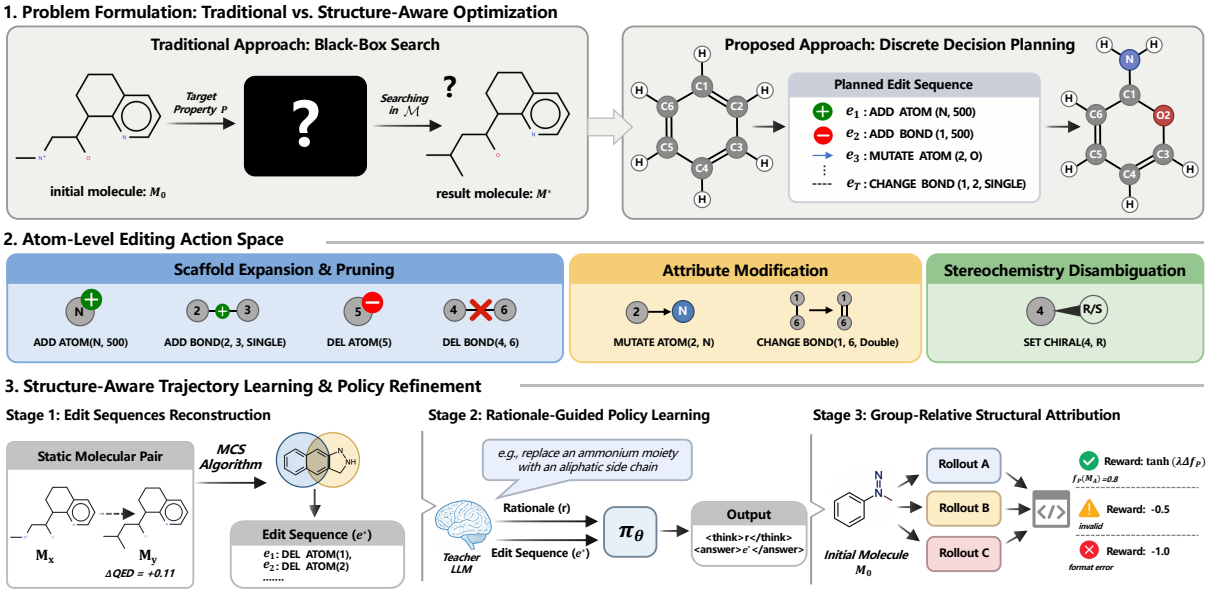

Figure 2: Overview of the AEMO framework. (1) **Problem Formulation**: Reformulates molecular optimization from black-box sequence generation to discrete atom-level decision planning on molecular graphs. (2) **Action Space**: Introduces atom-level edit operators anchored to the initial scaffold for fine-grained control. (3) **Trajectory Learning and Policy Optimization**: Reconstructs edit sequences via MCS for rationale-guided policy learning, then applies GRPO with tiered reward on validity and property improvement to promote efficient edit trajectories.

molecular optimization due to their strong representation learning and instruction-following abilities. Existing approaches typically formulate molecular optimization as conditional sequence generation. For example, ChatDrug (Liu et al., 2024) and AgentDrug (Le et al., 2024) iteratively refine molecules through prompt engineering and retrieval, while GeLLM^{3}o (Dey et al., 2025) and DrugAssist (Ye et al., 2025) construct instruction-tuning datasets to directly generate optimized SMILES. However, representing molecules as linear sequences forces LLMs to approximate local structural edits through global sequence rewriting, limiting fine-grained control under scaffold constraints. Although recent frameworks such as SynCraft (Li and Lai, 2025) and MECo (Zhu et al., 2025) decouple planning from execution, they still lack a shared editing space grounded in explicit structural operations. In contrast, our method operates within a shared action space of discrete atom-level edits and learns from parallel optimization trajectories, enabling explicit modeling of the causal relationship between structural modifications and property improvements.

3 Methods

To overcome the structural mismatch of traditional sequence generation, AEMO shifts toward explicit, structure-centric reasoning, as shown in Figure 2. We first reformulate molecular optimization as discrete decision planning (Section 3.1) over an atom-level action space (Section 3.2) that strictly anchors edits to the initial scaffold. Finally, Section 3.3 details our training pipeline.

3.1 Problem Formulation

Traditional Molecular Optimization and Its Limitations Traditional molecular optimization is commonly formulated as a black-box search problem driven by an oracle function. Given an initial molecule $M_0 \in \mathcal{M}$, represented by a SMILES string or a molecular graph, and a target property P defined by its corresponding oracle function:

$$f_P : \mathcal{M} \to \mathbb{R} \tag{1}$$

where $\mathcal{M}$ denotes the molecular space, the goal is to find a molecule M^* within the local neighborhood of M_0 that achieves a higher property value:

$$M^* = \underset{M \in \mathcal{N}(M_0)}{\arg\max}\ f_P(M) \tag{2}$$

where $\mathcal{N}(M_0)$ denotes the set of molecules reachable from M_0 via finite structural modifications. In LLMs-based approaches, this problem is often simplified as conditional sequence generation, where the model directly samples a complete SMILES string from the distribution $\pi_\theta(M \mid M_0, P)$.

However, the linearized nature of SMILES breaks the topological locality of molecular graphs.

As a result, generation decisions in sequence space do not reliably correspond to local structural edits. This structural mismatch causes molecular optimization to degenerate into a black-box search and prevents explicit modeling of the causal links between local edits and global property changes.

Discrete Decision Planning via Atom-level Edit Sequences To address these limitations, we reformulate molecular optimization as a discrete decision planning task over atom-level edit sequences. In this formulation, a molecule M is represented as a molecular graph:

$$G = (V, E) \tag{3}$$

where nodes $v \in V$ correspond to atoms and edges $\epsilon \in E$ correspond to chemical bonds. Each atom is assigned a unique identifier, enabling precise anchoring of all structural edits.

Given an initial molecule M_0 and a target property P, the policy π_θ does not directly generate the final molecule. Instead, it outputs a complete sequence of structural edits:

$$\mathbf{e} = \{e_1, e_2, \ldots, e_T\} \tag{4}$$

where each edit action e_t applies a local modification to the initial molecular graph. Molecular optimization is thus modeled as learning a conditional distribution over edit trajectories:

$$M_T = \mathcal{C}(M_0, \mathbf{e}) \quad \text{where} \quad \mathbf{e} \sim \pi_\theta(\cdot|M_0, P) \tag{5}$$

where $\mathcal{C}$ denotes the executable script that implements the specified edit sequence $\mathbf{e}$ to deterministically transforms M_0 into the final molecule M_T. This reformulation turns implicit black-box generation into explicit structure-centric planning, allowing the model to directly learn about how atom-level edits lead to global property improvements.

3.2 Atom-Level Editing Action Space

Inspired by prior work (Li and Lai, 2025) on molecular editing, we define atom-level operators as basic decision units for molecular optimization. The policy π_θ outputs a structured sequence of edit actions. Each action $e_t \in \mathcal{A}$ is deterministically mapped to a local graph transformation, rather than a global rewrite of SMILES. We partition the action space $\mathcal{A}$ into three core operator types:

- **Scaffold expansion and pruning**
  - `ADD ATOM`: Add a new atom and assign it an index starting from a large offset (e.g., ≥ 500) to distinguish generated atoms from the original scaffold.
  - `ADD BOND`: Create a bond between two atoms with an explicit bond order.
  - `DEL ATOM` / `DEL BOND`: Remove a specified atom or bond.
- **Attribute modification**
  - `MUTATE ATOM`: Change the element type of an existing atom.
  - `CHANGE BOND`: Modify the bond order, direction, or aromaticity.
- **Stereochemistry disambiguation**
  - `SET CHIRAL`: Specify atom chirality (R/S) or bond stereochemistry (E/Z).

Unlike end-to-end generation, this discrete action space serves as a strong structural inductive bias. By assigning unique identifiers to atom nodes, all edit operations are explicitly anchored to specific atoms and bonds. This design ensures that the editing process consistently remains centered on the original molecular scaffold. A detailed description of all valid commands and corresponding arguments is provided in Appendix A.6.

3.3 Structure-Aware Trajectory Learning and Relative Policy Refinement

To learn structure-aware molecular editing policies, we propose a two-stage training framework: (1) Supervised Fine-Tuning (SFT): We first cold-start π_θ with "Think-then-Act" supervision using atom-level edit sequences reconstructed from static molecular pairs. (2) GRPO: We then refine π_θ under scaffold constraints by comparing merits across various edit trajectories, attributing property gains to local edits. This turns molecular optimization from a black-box search into an interpretable editing decision process.

Edit Sequences Reconstruction To construct high-quality training trajectories, we reconstruct dynamic editing paths from static molecular pairs using graph algorithms. We first select molecular pairs (M_x, M_y) with significant property differences while maintaining high structural similarity. To extract the optimal edit sequence $\mathbf{e}^*$ from M_x to M_y, we apply the Maximum Common Substructure (MCS) algorithm (Raymond and Willett, 2002)

to decompose the topological differences into an atom-level edit sequence under the action space defined in Section 3.2. This procedure converts implicit property gaps into explicit, chemically valid step-by-step edit trajectories.

Rationale-Guided Policy Learning Since behavior cloning on edit sequence $\mathbf{e}^*$ alone provides limited semantic guidance, we employ a Teacher LLM to generate post-hoc rationales r for each optimization instance $(M_x, M_y, \mathbf{e}^*, \Delta P)$. To internalize the rationale, we formulate the SFT via a "Think-then-Act" paradigm, causally factorizing the policy distribution:

$$\pi_\theta(\tau|M_x, P) = \pi_\theta(r|M_x, P) \cdot \pi_\theta(\mathbf{e}^*|M_x, P, r) \tag{6}$$

where the rationale r acts as a high-dimensional semantic constraint, forcing the model to explicitly map chemical logic to atom-level edits, thereby compressing the sparse molecular search space into a chemically plausible subspace.

Group-Relative Structural Attribution of Property Changes To address the challenge of learning implicit SAR, we adopt GRPO to learn efficient edits under a fixed scaffold. Through group-based sampling, it explicitly models the causal link between local edits and target property changes. Formally, given an initial molecule M_0 and a target property P, the policy π_θ samples a group of trajectories in parallel, denoted as $\{\tau_i\}_{i=1}^G = \{\langle r_i, \mathbf{e}_i\rangle\}_{i=1}^G$. Since all trajectories share the same initial molecule, intra-group advantages provide a differential signal that isolates the marginal effect of atom-level edits. Each edit sequence $\mathbf{e}_i$ is deterministically executed to obtain optimized molecule M_i, defined as $M_i = \mathcal{C}(M_0, \mathbf{e}_i)$. We then design a tiered reward function:

$$R(\tau_i) = \begin{cases} -1.0, & \text{if } \tau_i \text{ has a format error,} \\ -0.5, & \text{if } M_i \text{ is invalid,} \\ \max(0, R_i^{\text{prop}}), & \text{otherwise.} \end{cases} \tag{7}$$

where the trajectory τ_i is required to follow a predefined format. The property reward R_i^{prop} is computed from the improvement of the target property:

$$\begin{aligned} R_i^{\text{prop}} &= \tanh(\lambda \cdot \Delta P_i)\,, \\ \Delta P_i &= f_P(M_i) - f_P(M_0) \end{aligned} \tag{8}$$

where $f_P(\cdot)$ denotes the oracle function for property P. Here, $\tanh(\cdot)$ normalizes property improvements to $[0, 1]$ for stable training, and the hyperparameter λ controls the scaling magnitude of the reward signal. Since unconstrained atom-level edits may disconnect the molecular graph into multiple fragments, such structures are regarded as invalid and assigned a penalty of -0.5. Detailed information is in Appendix A.5.

4 Experiments

4.1 Experimental Setting

Task Settings. We evaluate our method on six single-property molecular optimization tasks from ChemCoTBench (Li et al., 2025). These tasks cover three physicochemical properties (LogP (Lipinski et al., 1997), Solubility (Delaney, 2004), QED (Bickerton et al., 2012)) and three target-specific bioactivities (DRD2 (Fan et al., 2020), JNK3 (Gupta et al., 1996), GSK-3β (Woodgett, 1990)). Under the AEMO framework, we applied a unified wrapper to the original prompt to constrain the output format and facilitate reliable extraction of editing actions. The full wrapper template is provided in Appendix A.7.

Datasets. We construct our training dataset from MuMOInstruct (Dey et al., 2025) and ChemCoTDatasets (Li et al., 2025). For "Think-then-Act" learning, we construct 10K edit trajectories for each of the pLogP, QED, and DRD2 tasks using molecular pairs from MuMOInstruct. During the SFT stage, we train only on these three tasks, leaving the remaining three tasks as SFT Out-Of-Distribution (SFT-OOD) tasks. We use Gemini-3-Flash (Google DeepMind, 2025) as the Teacher LLM. Detailed distillation prompt is provided in Appendix A.8. For the GRPO stage, we train on all tasks using 2,557 uniformly sampled molecular pairs from ChemCoTDatasets. A more detailed description of the datasets is provided in Appendix A.5.

Evaluation Metrics. Following the benchmark protocol, we employ two complementary metrics: (1) Success Rate (SR%): percentage of molecules achieving a property increase; (2) Δ: average absolute improvement across target properties.

Baselines. We evaluate AEMO against ten baseline methods across three categories: (1) General-purpose reasoning LLMs: Gemini-3-Pro, Gemini-2.5-Pro (Comanici et al., 2025), and DeepSeek-R1 (Guo et al., 2025); (2) General-purpose non-reasoning LLMs: GPT-4o (OpenAI, 2024), Gemini-3-Flash, and Gemini-2.5-Flash(Comanici

Table 1: Performance Comparison on ChemCoTBench for molecular optimization tasks. Each task reports success rate (SR%), average absolute improvement (Δ). The higher the better. The best results for each task are **bolded**, and the second-best are underlined. Benchmarked results are quoted from ChemCoTBench (Li et al., 2025).

Methods	LogP		Solubility		QED		DRD2		JNK3		GSK-3β	
	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ
W/o thinking LLMs												
GPT-4o	42	-0.20	80	0.82	70	0.05	48	0.05	30	-0.05	39	-0.04
Gemini-3-Flash	90	0.85	81	0.79	87	0.17	66	0.25	42	0.03	66	0.07
Gemini-2.5-Flash	75	0.35	54	0.19	79	0.10	63	0.15	34	0.03	38	0.00
W/ thinking LLMs												
Gemini-3-Pro	67	0.40	80	0.56	**93**	**0.26**	76	**0.40**	41	**0.05**	69	**0.14**
Gemini-2.5-Pro	81	-0.28	92	1.91	84	0.21	74	0.35	35	-0.04	68	0.04
DeepSeek-R1	74	0.36	97	1.48	72	0.05	62	0.10	29	-0.06	41	-0.02
Task-Specific LLMs												
Intern-S1-mini	3	0.00	6	0.01	50	0.07	17	0.00	11	0.00	8	0.00
GeLLM^{3}O	18	0.22	9	0.06	29	0.06	20	0.06	5	0.00	15	0.01
MECo	72	-	96	-	75	-	66	-	39	-	60	-
MolOptAgent-7B	92	1.02	81	1.44	30	0.01	29	0.00	20	-0.03	42	0.05
AEMO												
Qwen2.5-1.5B-Instruct	96	1.50	95	1.75	90	0.21	71	0.20	46	-0.01	66	0.03
Qwen2.5-3B-Instruct	96	1.63	97	2.31	85	0.23	**84**	0.25	42	-0.01	68	0.03
Qwen2.5-7B-Instruct	**98**	**2.26**	**98**	**3.02**	86	0.21	**84**	0.25	**51**	0.01	**76**	0.05
Qwen3-1.7B	90	1.19	88	0.91	83	0.18	57	0.05	30	-0.03	40	0.01
Llama3.2-1B-Instruct	92	1.24	91	1.76	82	0.22	63	0.25	31	-0.03	69	0.05

et al., 2025); (3) Task-specific LLMs: Intern-S1-mini (Bai et al., 2025), GeLLM^{3}O, MECo, and MolOptAgent (Yang et al., 2025), which are specifically trained for molecular tasks. Notably, AEMO uses Qwen2.5-Instruct (Qwen et al., 2025) family, spanning 1.5B–7B parameters, while most LLM baselines rely on models with more than 7B parameters. A comprehensive list of all training hyperparameters is provided in Appendix A.2.

4.2 Main Results

Table 1 presents the main results on single-objective molecular optimization tasks.

AEMO Surpasses Existing Baselines. Although strong reasoning models such as Gemini-3-Pro achieve a high success rate of 93% on simpler heuristic tasks like QED, general-purpose LLMs suffer a significant performance collapse on other complex tasks requiring fine-grained scaffold modifications. In contrast, AEMO ensures precise control over microscopic edits to achieve a SOTA success rate of 85% on the challenging DRD2 task. Furthermore, existing domain-specific models such as MolOptAgent-7B lack stability across different tasks, achieving a mere 29% success rate on DRD2. AEMO consistently outperforms these baselines on all tasks, providing a robust and highly effective optimization paradigm.

AEMO Demonstrates Exceptional Generalization on SFT-OOD Tasks. In our two-stage training setup, the SFT phase covers only QED, LogP, and DRD2. This implies that Solubility, JNK3, and GSK-3β are completely unseen optimization objectives for the model. SMILES-based sequence generation models often face grammatical hallucination or generate invalid molecules when encountering these SFT-OOD tasks. However, using only relative preference signals during the full-properties GRPO phase, AEMO achieves remarkable results on these SFT-OOD tasks. For instance, Qwen2.5-7B-Instruct achieves a peak SR% of 98% and Δ of 3.02 on the unseen Solubility task.

AEMO Exhibits High Parameter Efficiency and Scalability. Despite having only 1.5B parameters, AEMO consistently surpasses both large closed-source models and specialized 7B baselines such as MECo. Furthermore, AEMO exhibits remarkable scalability and robustness, achieving SOTA performance when scaled to 7B parameters. This consistent success across various LLM backbones, including Llama3.2-1B and Qwen3-1.7B,

Table 2: Validity rate (VR%), average structural Similarity (Sim) on ChemCoTBench for molecular optimization tasks. The higher the better. The best results for each task are **bolded**, and the second-best are underlined. * Results are quoted from MECo (Zhu et al., 2025).

Methods	LogP		Solubility		QED		DRD2		JNK3		GSK-3β	
	VR%	Sim	VR%	Sim	VR%	Sim	VR%	Sim	VR%	Sim	VR%	Sim
General-Purpose LLMs												
GPT-4o*	22	0.11	41	0.21	31	0.11	27	0.12	20	0.08	19	0.08
Gemini-2.5-Pro*	78	0.37	79	0.40	83	0.33	85	0.36	74	0.32	74	0.38
DeepSeek-R1*	62	0.31	70	0.39	68	0.31	73	0.36	43	0.19	45	0.22
Task-Specific LLMs												
Intern-S1-mini	3	**0.73**	6	0.63	95	**0.80**	63	**0.80**	47	**0.83**	19	**0.76**
GeLLM^{3}O	25	0.66	17	0.67	34	0.65	25	0.60	23	0.59	29	0.63
MECo*	93	0.54	96	0.63	87	0.49	89	0.55	92	0.55	91	0.56
MolOptAgent-7B	92	0.70	84	0.58	64	0.55	48	0.51	57	0.48	48	0.50
AEMO												
Qwen2.5-1.5B-Instruct	96	0.66	96	0.64	**99**	0.64	**100**	0.57	**100**	0.56	98	0.56
Qwen2.5-3B-Instruct	**98**	0.60	97	0.58	94	0.61	97	0.58	99	0.60	99	0.58
Qwen2.5-7B-Instruct	**98**	0.55	**98**	0.47	97	0.60	99	0.58	99	0.57	**100**	0.59
Qwen3-1.7B	**98**	0.68	94	**0.71**	97	0.70	97	0.73	94	0.71	69	0.72
Llama3.2-1B-Instruct	96	0.64	92	0.61	94	0.65	97	0.61	94	0.65	**100**	0.66

further confirms AEMO as a versatile and efficient paradigm for empowering small language models in single-property molecular optimization.

4.3 Further Analysis

Beyond success rates, chemical Validity Rate (VR) and average structural Similarity (Sim) are critical metrics for assessing practicality. Specifically, Sim is quantified as the Tanimoto similarity between the ECFP4 fingerprints of the initial and optimized molecules. Table 2 reports the performance of AEMO compared to baselines.

AEMO Exhibits Robust Chemical Validity. AEMO maintains a near-perfect VR% exceeding 96% across all tasks, even at the 1.5B parameter scale. Conversely, SMILES-based generation baselines such as GPT-4o often fail to master long-range syntactic dependencies, resulting in a large volume of invalid strings with VR% dropping as low as 17%. These results confirm that AEMO's atom-level action space fundamentally circumvents the inherent syntax errors of SMILES, ensuring highly reliable molecular modifications.

AEMO Balances Property Optimization with Structural Retention. In molecular optimization, an ideal model must strike a balance between enhancing target properties and preserving the original molecular scaffold. As shown in Table 2, although Intern-S1-mini maintains a high Sim ranging from 0.73 to 0.83, its severely degraded success rate of under 20% suggests that the model fails to learn meaningful chemical optimizations within the linear representation space. Conversely, AEMO achieves superior SR% while maintaining a highly competitive Sim of approximately 0.60. This indicates AEMO executes controlled, local modifications instead of generating unrelated molecules to cheat for higher property boosts, offering greater practical utility for real-world drug discovery.

The Impact of Scaling on Structural Retention. Scaling AEMO's backbone models significantly improves SR% and Δ, but at the cost of decreased Sim. This trend suggests that larger models, given their greater expressive capacity, are highly susceptible to reward hacking during optimization. Specifically, they execute aggressive edits to maximize rewards, causing excessive deviation from the initial scaffold. Sacrificing structural fidelity contradicts real-world lead optimization principles. This exposes a classic reinforcement learning problem in molecular optimization: the tension between maximizing property-improvement rewards and adhering to molecular structural constraints leads to reward hacking. Therefore, future work will explicitly integrate structural similarity into the reward formulation to mitigate reward hacking and strictly preserve scaffold fidelity during exploration.

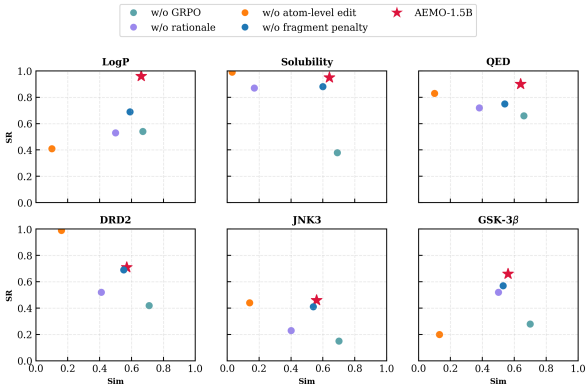

Figure 3: Ablation study of AEMO. The x-axis and y-axis represent average structural Similarity (Sim) and Success Rate (SR), respectively.

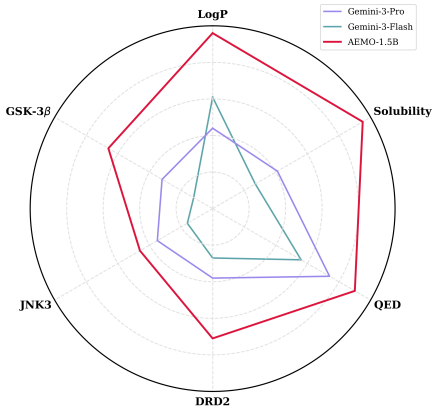

Figure 4: SR% of SOTA general-purpose LLMs, using the prompt wrapper described in the Appendix A.7

4.4 Ablation Studies

We conducted ablation studies to validate the contributions of key components in AEMO, including GRPO, the rationale of the edit sequence, atom-level edits, and the fragment penalty in the reward function (7). Results are shown in Figure 3 with numerical values detailed in Appendix A.1.

Experimental results demonstrate that removing GRPO severely degrades performance on all tasks. Specifically, the SR% drops sharply across all tasks. This highlights that imitation learning alone is insufficient for navigating high-dimensional search spaces. Instead, distinguishing the relative merits of local edits via intra-group comparisons serves as the crucial engine for discovering successful and efficient optimization trajectories. Furthermore, omitting the rationale induces a drastic decline in Sim across most tasks. This confirms that an explicit, structure-centric reasoning process effectively establishes a causal link between local edits and property variations. Consequently, it prevents the model from destroying the molecular scaffold merely to maximize property values.

Removing atom-level edits and reverting to direct SMILES generation triggers a catastrophic collapse in Sim. Consequently, the model degenerates into a generic SMILES generator incapable of retaining the initial scaffold. This empirically validates our core hypothesis. Discrete graph editing actions are essential for anchoring optimization to the local chemical neighborhood. They effectively resolve the structural mismatch inherently caused by linear SMILES representations. Furthermore, omitting the fragment penalty leads to a decline in SR%. This demonstrates that ignoring molecular graph connectivity severely weakens structural constraints, ultimately destroying both the chemical validity and controllability of the editing.

For a fair comparison, we apply the prompt wrapper in Appendix A.7 to SOTA general-purpose LLMs, enforcing the same atom-level edit sequences as AEMO. Results are shown in Figure 4. Under this identical setup, both Gemini-3-Pro and Gemini-3-Flash achieve substantially lower success rates than AEMO based on Qwen2.5-1.5B-Instruct across all tasks. This suggests that surface-level prompting is insufficient: forcing models to output reasoning and edit sequences does not confer genuine structural understanding. Without relative policy learning in a shared editing space, they fail to link local edits to global properties and remain limited to inefficient black-box optimization.

5 Conclusion

In this work, we propose AEMO, a new paradigm for molecular optimization based on atom-level action space and relative policy optimization. By reformulating molecular optimization as a discrete decision process over atom-level edit sequences and driven by relative policy optimization, we resolve the structural mismatch caused by SMILES linearization and explicitly attribute global property improvements to local structural edits. Our empirical evaluation on ChemCoTBench shows that with only 1.5B parameters, AEMO achieves SOTA performance, effectively transforming implicit black-box search into an explicit structure-centric reasoning process. These results confirm that with carefully designed action spaces and efficient policy optimization, small models can meaningfully solve complex scientific optimization problems. The model and datasets will be publicly released to facilitate reproducibility.

Limitations

Although AEMO resolves the SMILES structural mismatch via discrete atom-level editing, it faces two primary limitations. First, strictly confining the action space to fine-grained atom-level operations necessitates overly long edit sequences for tasks requiring large-scale scaffold hopping. This exacerbates long-horizon planning difficulties for LLMs and restricts exploration efficiency. Second, AEMO's reliance on static oracles makes it susceptible to reward hacking, especially as model capacity scales. Larger policies tend to over-optimize these scalar proxies by aggressively altering the scaffold, thereby compromising structural fidelity. Future work will introduce an "atom-fragment" hybrid action space to accelerate macroscopic edits and explicitly integrate structural constraints into a multi-objective reward framework, mitigating shortcut learning to further bridge the gap to practical drug design.

References

Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. 2021. Molgpt: molecular generation using a transformer-decoder model. *Journal of chemical information and modeling*, 62(9):2064–2076.

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, and 1 others. 2025. Intern-s1: A scientific multimodal foundation model. *arXiv preprint arXiv:2508.15763*.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. 2012. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98.

Konrad H Bleicher, Hans-Joachim Böhm, Klaus Müller, and Alexander I Alanine. 2003. Hit and lead generation: beyond high-throughput screening. *Nature reviews Drug discovery*, 2(5):369–378.

Ziqi Chen, Martin Renqiang Min, Srinivasan Parthasarathy, and Xia Ning. 2021. A deep generative model for molecule optimization via one fragment modification. *Nature Machine Intelligence*, 3(12):1040–1049.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

John S Delaney. 2004. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005.

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

Peter Eckmann, Kunyang Sun, Bo Zhao, Mudong Feng, Michael K Gilson, and Rose Yu. 2022. Limo: Latent inceptionism for targeted molecule generation. *Proceedings of machine learning research*, 162:5777.

Luyu Fan, Liang Tan, Zhangcheng Chen, Jianzhong Qi, Fen Nie, Zhipu Luo, Jianjun Cheng, and Sheng Wang. 2020. Haloperidol bound d2 dopamine receptor structure inspired the discovery of subtype selective ligands. *Nature communications*, 11(1):1074.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2023. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*.

Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. 2022. Language models can learn complex molecular distributions. *Nature Communications*, 13(1):3293.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. 2022. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35:21342–21357.

Google Gemini Team. 2025. Gemini 3 pro: State-of-the-art reasoning and multimodal understanding. Technical report, Google DeepMind.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276.

Google DeepMind. 2025. Gemini 3 flash: Frontier intelligence built for speed. Accessed: 2026-02-22.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shashi Gupta, Tamera Barrett, Alan J Whitmarsh, Julie Cavanagh, Hayla Karen Sluss, Benoit Dérijard, and Roger J Davis. 1996. Selective interaction of jnk protein kinase isoforms with transcription factors. *The EMBO journal*, 15(11):2760–2770.

Jiazhen He, huifang you, Emil Sandström, eva nittinger, Esben Jannik Bjerrum, Christian Tyrchan, Werngard Czechtizky, and Ola Engkvist. 2020. Molecular optimization by capturing chemist's intuition using deep neural networks. *ChemRxiv*, 2020(1029).

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. 2021. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*.

James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. 2011. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249.

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

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018a. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. 2018b. Learning multimodal graph-to-graph translation for molecular optimization. *arXiv preprint arXiv:1812.01070*.

Greg Landrum and 1 others. 2013. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281.

Khiem Le, Ting Hua, and Nitesh V Chawla. 2024. Agentdrug: Utilizing large language models in an agentic workflow for zero-shot molecular optimization. *arXiv preprint arXiv:2410.13147*.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Li Yuan, Yonghong Tian, and Yu Li. 2025. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *arXiv preprint arXiv:2505.21318*.

Junren Li and Luhua Lai. 2025. Syncraft: Guiding large language models to predict edit sequences for molecular synthesizability optimization. *arXiv preprint arXiv:2512.20333*.

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

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2024. Conversational drug editing using retrieval and domain feedback. In *The twelfth international conference on learning representations*.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. 2017. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48.

OpenAI. 2024. Gpt-4o system card. Official Technical Report. Accessed: 2026-02-22.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. 2018. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nian Ran, Yue Wang, Xiaoyuan Zhang, Zhongzheng Li, Qingsong Ran, Wenhao Li, and Richard Allmendinger. 2025. Exllm: Experience-enhanced llm optimization for molecular design and beyond. *arXiv preprint arXiv:2502.12845*.

John W Raymond and Peter Willett. 2002. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of computer-aided molecular design*, 16(7):521–533.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. *URL https://arxiv. org/abs/2402.03300*, 2(3):5.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

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

Jiangming Sun, Nina Jeliazkova, Vladimir Chupakhin, Jose-Felipe Golib-Dzib, Ola Engkvist, Lars Carlsson, Jörg Wegner, Hugo Ceulemans, Ivan Georgiev, Vedrin Jeliazkov, and 1 others. 2017. Excape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of cheminformatics*, 9(1):17.

Haorui Wang, Marta Skreta, Cher-Tian Ser, Wenhao Gao, Lingkai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, and 1 others. 2024. Efficient evolutionary search over chemical space with large language models. *arXiv preprint arXiv:2406.16976*.

James Robert Woodgett. 1990. Molecular cloning and expression of glycogen synthase kinase-3/factor a. *The EMBO journal*, 9(8):2431–2438.

Dong Xu, Qihua Pan, Sisi Yuan, Jianqiang Li, Zexuan Zhu, and Junkai Ji. 2026. Unveiling scaling behaviors in molecular language models: Effects of model size, data, and representation. *arXiv preprint arXiv:2601.22757*.

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

Zhuo Yang, Yeyun Chen, Jiaqing Xie, Ben Gao, Shuaike Shen, Wanhao Liu, Liujia Yang, Beilun Wang, Tianfan Fu, and Yuqiang Li. 2025. Molact: An agentic rl framework for molecular editing and property optimization. *arXiv preprint arXiv:2512.20135*.

Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. 2025. Drugassist: A large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1):bbae693.

Botao Yu, Frazier N Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. 2019. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):10752.

Wenyu Zhu, Chengzhu Li, Xiaohe Tian, Yifan Wang, Yinjun Jia, Jianhui Wang, Bowen Gao, Ya-Qin Zhang, Wei-Ying Ma, and Yanyan Lan. 2025. Coder as editor: Code-driven interpretable molecular optimization. *arXiv preprint arXiv:2510.14455*.

A Experiment Details

A.1 Ablation Results

We detail the full results of the ablation studies in table 3.

A.2 Training Configurations

We detail the corresponding hyperparameters for SFT and GRPO in Tables 4 and 5. The SFT stage is implemented using the LLaMA-Factory (Zheng et al., 2024) framework, while the GRPO stage is conducted utilizing the verl (Sheng et al., 2024) framework.

A.3 Pharmacological Metrics

We employ the following pharmacological metrics for the single-property molecular optimization tasks in ChemCoTBench:

- LogP (Octanol-water Partition Coefficient) (Lipinski et al., 1997): LogP quantifies the lipophilicity of a compound, reflecting its partitioning tendency between non-polar and polar environments. A higher LogP value typically indicates stronger lipophilicity and transmembrane diffusion capacity. However, it may be accompanied by decreased aqueous solubility and an increased risk of non-specific binding.
- QED (Quantitative Estimate of Drug-likeness) (Bickerton et al., 2012): QED systematically evaluates drug-likeness by comprehensively quantifying eight key physicochemical properties: molecular weight, octanol-water partition coefficient, hydrogen bond donors and acceptors, polar surface area, rotatable bonds, aromatic rings, and structural alerts. A higher QED indicates closer alignment with the statistical distribution of approved small-molecule drugs in multidimensional physicochemical space, generally implying superior pharmacokinetic properties and development potential.
- Solubility (Delaney, 2004): Solubility measures the maximum dissolved concentration of a compound at thermodynamic equilibrium in an aqueous phase. It directly determines the drug's dissolution rate and free concentration in gastrointestinal fluids. Higher solubility facilitates increased gastrointestinal dissolution and systemic exposure.
- DRD2 (Dopamine Receptor D2) (Fan et al., 2020): DRD2 measures the binding affinity of a molecule to the dopamine receptor D2 subtype.

Table 3: Ablation study of AEMO. Each task reports success rate (SR%), average structural similarity (Sim). The higher the better. The best results for each task are **bolded**, and the second-best are underlined.

Methods	LogP		Solubility		QED		DRD2		JNK3		GSK-3β	
	SR%	Sim	SR%	Sim	SR%	Sim	SR%	Sim	SR%	Sim	SR%	Sim
AEMO	**96**	0.66	95	0.64	**90**	0.64	71	0.57	46	0.56	**66**	0.56
w/o GRPO	54	**0.67**	38	**0.69**	66	**0.66**	42	**0.71**	15	**0.70**	28	**0.70**
w/o rationale	53	0.50	87	0.17	72	0.38	52	0.41	23	0.40	52	0.50
w/o atom-level edit	41	0.10	**99**	0.03	83	0.10	**99**	0.16	44	0.14	20	0.13
w/o fragment penalty	69	0.59	88	0.60	75	0.54	69	0.55	41	0.54	57	0.53

Table 4: SFT hyperparameters used in our experiments.

Hyperparameter	Value
LoRA rank	64
LoRA alpha	128
LoRA target	all linear layers
Batch size	2
Epochs	3
Gradient accumulation	4
Learning rate	5e-5
Warmup ratio	0.1
LR scheduler	cosine

Table 5: GRPO hyperparameters used in our experiments.

Hyperparameter	Value
Batch size	32
Epochs	3
Learning rate	1e-6
Group Size	8
KL coeffcient	0.01
Generation temperature	1.0
Reward λ	1.0

A higher DRD2 affinity indicates stronger ligand-receptor binding. This effectively modulates central dopaminergic signaling and is directly relevant to targeted therapies for neurological disorders such as Parkinson's disease.

- JNK3 (c-Jun N-terminal Kinase 3) (Gupta et al., 1996): JNK3 evaluates the specific inhibitory potency of a molecule against c-Jun N-terminal kinase 3. Enhancing the inhibitory activity against this target aims to block neurodegenerative pathological cascades, providing critical neuroprotective interventions for Alzheimer's disease and related conditions.

- GSK-3β (Glycogen Synthase Kinase 3 Beta) (Woodgett, 1990): GSK-3β quantifies the inhibitory affinity of a compound against glycogen synthase kinase-3β. Stronger GSK-3β inhibitory activity indicates the molecule can effectively regulate related signaling pathways. This potentially mitigates abnormal Tau phosphorylation and neuroinflammation, offering therapeutic potential for Alzheimer's disease.

A.4 Oracle Functions

Molecular optimization tasks require oracle functions to evaluate the quality of generated molecules. We utilize TDC (Huang et al., 2021) and ESOL (Delaney, 2004), which together provide the most comprehensive collection of molecular oracles currently available:

- LogP: The oracle is calculated via RDKit (Landrum et al., 2013).

- QED: The oracle is calculated via RDKit, using a set of chemical rules about drug-likeness defined by (Bickerton et al., 2012).

- Solubility: The oracle is a linear regression equation that expresses the log of the aqueous solubility in terms of simple molecular descriptors calculated from the structure. It is derived by fitting experimental solubilities to descriptor values using multiple linear regression.

- DRD2: The oracle is constructed by (Olivecrona et al., 2017), using a support vector machine classifier with a Gaussian kernel with ECFP6 fingerprint on ExCAPE-DB dataset (Sun et al., 2017).

- JNK3 & GSK-3β: The oracle is a random forest classifer using ECFP6 fingerprints using ExCAPE-DB dataset.

A.5 Dataset Details

MuMOInstruct is the first instruction-tuning dataset specifically designed for complex, multi-property molecular optimization. Its molecular pairs (M_x, M_y) originate from a foundational library of 255K pairs provided by (Chen et al., 2021). Each pair differs at only a single cleavage site. This

means M_x can be transformed into M_y by modifying exactly one fragment. The dataset covers five key pharmaceutical properties: plogP, QED, BBBP, Mutag, and DRD2.

The molecular optimization tasks in ChemCoT-Dataset derive from 198,559 property-annotated pairs provided by Deep-Mol-Opt (He et al., 2020). This dataset includes preliminary filtering during its construction. It excludes samples with marginal property improvements ($\Delta < 0.3$) and those with complex polycyclic structures. Its task objectives cover six core properties: QED, Solubility, LogP, DRD2, JNK3, and GSK-3β.

Using two public datasets, we perform task-specific processing for the SFT and GRPO stages. For SFT, we sample 20K molecular pairs from each MuMOInstruct subtask (plogP, QED, DRD2) and extract atom-level edit sequences via MCS. To avoid excessively long trajectories, we apply a 1-second timeout filter and retain 10K high-quality pairs with complete edit sequences per subtask for distillation. For GRPO, we randomly sample across all ChemCoTDataset subtasks, yielding 2,557 molecular pairs. These samples include only task instructions and final target molecules, without intermediate reasoning steps. We format the trajectory as $\tau = \langle$<think>r</think><answer>$\mathbf{e}^*$</answer>$\rangle$ and optimize π_θ using autoregressive cross-entropy loss.

A.6 Commands and Arguments of Action Space

Following the formulation in SynCraft, the discrete atom-level editing decisions generated by the LLM are restricted to a sequence of commands operating on specific atom indices. The valid commands and their corresponding arguments are defined as follows:

- **`["ADD ATOM", new id, symbol]`**: Introduces a new atom. To avoid index conflict, new IDs start from 500.
- **`["ADD BOND", id1, id2, bond type]`**: Creates a bond between atoms. Bond types include `"SINGLE"`, `"DOUBLE"`, `"TRIPLE"`, and `"AROMATIC"`.
- **`["DEL ATOM", map num]`**: Deletes the atom specified by the map number and its incident bonds.
- **`["DEL BOND", map num1, map num2]`**: Removes the bond between two existing atoms.
- **`["MUTATE ATOM", map num, new symbol]`**: Changes the atomic element of the target atom (e.g., C to N).
- **`["CHANGE BOND", map num1, map num2, new bond type]`**: Modifies the order of an existing bond.
- **`["SET CHIRAL", id, tag]`**: Explicitly sets stereocenters (CW/CCW).
- **`["STOP"]`**: A mandatory token indicating the end of the edit sequence.

A.7 Prompt Wrapper

Prompt Template

You are an expert medicinal chemist specializing in molecular editing and optimization. Your task is to optimize the given molecule using the specified Molecular Editing Operations to increase {target_property}, while keeping structural changes as minimal as possible. The modified molecule should maintain a structural similarity of at least 0.4 with the original molecule.

Molecular Editing Operations:

- `["DEL_ATOM", map_num]`: Deletes the atom specified by the map number and its incident bonds.
- `["MUTATE_ATOM", map_num, new_symbol]`: Changes the atomic element of the target atom (e.g., C to N).
- `["ADD_ATOM", new_id, symbol]`: Introduces a new atom. To avoid index conflict, new IDs start from 500.
- `["DEL_BOND", map_num1, map_num2]`: Removes the bond between two existing atoms.
- `["ADD_BOND", id1, id2, bond_type]`: Creates a bond between atoms. Bond types include `SINGLE`, `DOUBLE`, `TRIPLE`, `AROMATIC`.
- `["CHANGE_BOND",map_num1,map_num2, new_bond_type]`: Modifies the order of an existing bond.
- `["SET_CHIRAL", id, tag]`: Explicitly sets stereocenters (CW/CCW).
- `["STOP"]`: A mandatory token indicating

the end of the edit sequence.

Initial molecule SMILES:
{source_smiles_with_maps}

Think about how to improve the initial molecule step by step, you must use ONLY the provided Molecular Editing Operations, ensure new atom IDs start from 500, and the sequence must ends with ["STOP"].

The <answer> section must contain only a single valid JSON list of operations without any markdown code blocks or additional text:

<think>Your reasoning about how to optimize the molecule...</think> <answer>[["OPERATION_1"],["OPERATION_2"],["STOP"]]</answer>

A.8 Distillation Prompt

Prompt Template

You are an expert in Molecular Edit and Molecular Design. Your task is analyze the reasoning process of molecular optimization based on a specific editing sequence.

Input Data:

- Optimization Target: increase {target_property}
- Source Molecule: {source_smiles}
- Target Molecule: {target_smiles}

Editing Operations:
{editing_sequence}

Operational Reference:

- ["DEL_ATOM", map_num]: Deletes the atom specified by the map number and its incident bonds.
- ["MUTATE_ATOM", map_num, new_symbol]: Changes the atomic element of the target atom (e.g., C to N).
- ["ADD_ATOM", new_id, symbol]: Introduces a new atom. To avoid index conflict, new IDs start from 500.
- ["DEL_BOND", map_num1, map_num2]: Removes the bond between two existing atoms.
- ["ADD_BOND", id1, id2, bond_type]: Creates a bond between atoms. Bond types include SINGLE, DOUBLE, TRIPLE, AROMATIC.
- ["CHANGE_BOND",map_num1,map_num2, new_bond_type]: Modifies the order of an existing bond.
- ["SET_CHIRAL", id, tag]: Explicitly sets stereocenters (CW/CCW).
- ["STOP"]: A mandatory token indicating the end of the edit sequence.

Analysis Requirements: Based on the molecular editing operation sequence, output the reasoning process for molecular optimization, including a description of the editing of specific functional groups and the rationale, and assuming prior knowledge of the target molecule and the specific operation sequence, gradually deduce the specific operation sequence.

Strict Constraints:

- If you cannot guarantee that your analysis is absolutely correct, simply output 'I don't know'.
- ONLY output the reasoning process in one paragraph.
- Do not imply that you knew the molecular editing sequence in advance.
- NO introductory phrases/headings, meta-talk, or irrelevant statements.
- Maximum 200 tokens.
- Do not ask me any questions in the end.

B Visualization of Atom-level Editing

To illustrate how atom-level edit sequences achieve molecular structure modifications, we provide examples of complete optimization trajectories in this section. Detailed visual demonstrations for QED, plogP, and DRD2 optimization are presented in Figure 5, 6, and 7, respectively.

C Case Studies

To highlight the difference between direct SMILES generation and our AEMO framework, we analyze molecular optimization tasks and present representative examples in Figure 8. Direct generation

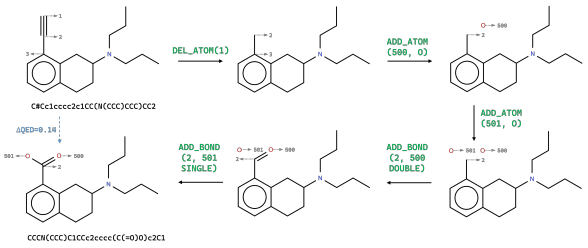

Figure 5: An example of the step-by-step atom-level edit trajectory for QED optimization

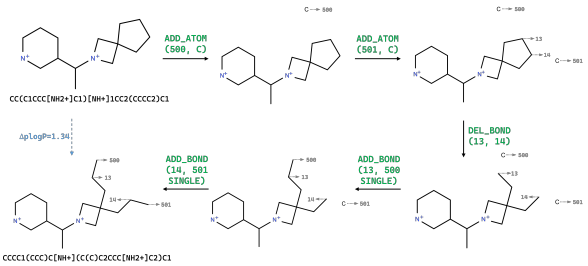

Figure 6: An example of the step-by-step atom-level edit trajectory for plogP optimization

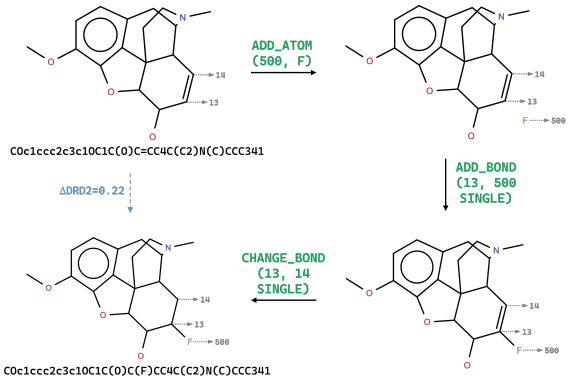

Figure 7: An example of the step-by-step atom-level edit trajectory for DRD2 optimization

methods frequently fail to preserve the scaffold of the initial molecule, resulting in structural drift and ultimately failed optimization. In contrast, AEMO applies interpretable edits that maintain high similarity to the initial molecule while improving the oracle score.

Figure 9 illustrates a representative trajectory of optimizing the DRD2 property. As shown, AEMO's rationale accurately analyzes the source molecular structure and formulates a targeted modification strategy based on the target property. It then sequentially describes the editing process and explicitly explains the underlying link between local structural modifications and global property changes. Notably, the structural differences between the source and optimized molecules are precisely analyzed, and the description of the editing

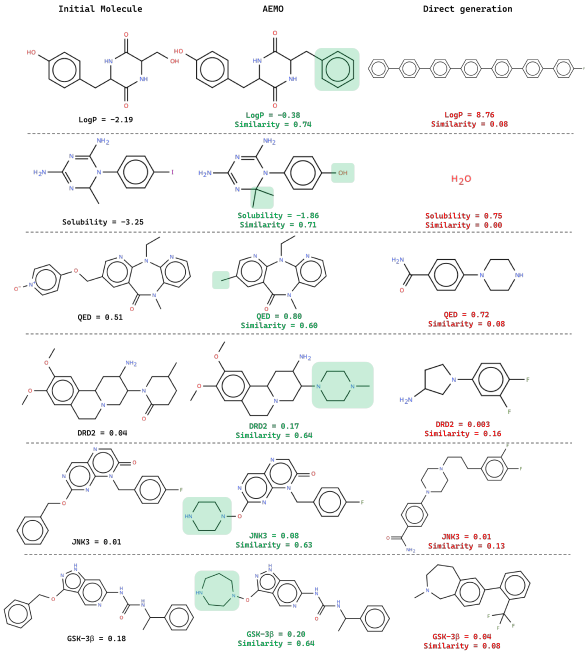

Figure 8: Representative case studies of molecular optimization comparing AEMO and direct SMILES generation.

process strictly corresponds to the final edit sequence. This high degree of alignment between the natural language rationale and the final edit sequence demonstrates the correctness, logical consistency, and interpretability of AEMO.

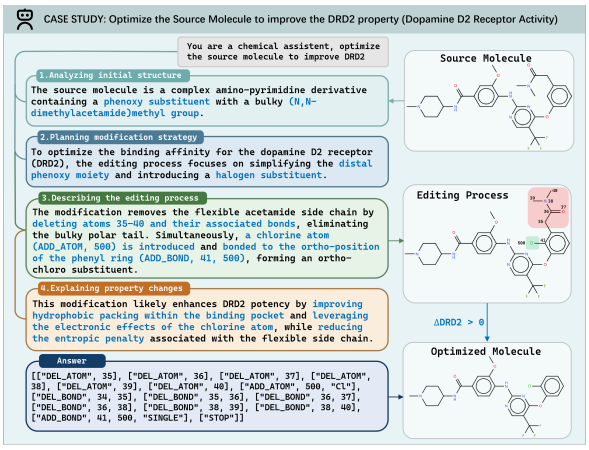

Figure 9: A case study of the alignment between AEMO's optimization rationale and the edit sequence for DRD2.