CovMOLAgent: Execution-Grounded Policy Learning for Multi-Objective Covalent Molecule Optimization

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) have shown considerable promise in chemistry, yet existing paradigms mainly focus on the design and single-objective optimization of chemical molecules. Multi-objective optimization of a given covalent molecule presents two challenges. First, tool availability does not equate to strategic tool use: agents struggle to recover from failed edits or decide when to stop, resulting in brittle search trajectories. Second, multi-objective search tends to improve molecular properties at the cost of destroying the key warhead that determines the covalent mechanism, while heterogeneous property scales make coordinated improvement difficult. To address these challenges, we propose Covalent Multi-Objective Lead Optimization Agent (CovMOLAgent), an agentic framework for multi-objective covalent lead optimization. Specifically, we first introduce Covalent Executable Trajectory Reconstruction and Alignment (CETRA), an execution-grounded trajectory reconstruction strategy. CETRA constructs failure-aware search trajectories based on deterministic tool executions, covering behaviors such as invalid trials and negative explorations, thereby grounding decisions in executable evidence rather than hypothesized molecular states. We further develop Warhead-Constrained Policy Optimization (WCPO), a source-relative multi-objective optimization mechanism. It incorporates direction-normalized multi-property rewards and a source-warhead penalty, among other components, to reconcile heterogeneous property scales and favor retention of source-warhead labels. Comprehensive evaluations on a held-out benchmark show that CovMOLAgent achieves the strongest aggregate and joint multi-property performance, with leading results on several individual-property metrics, while retaining all recognized source-warhead labels in 90.9% of benchmark instances.

1 Introduction

Chemistry-based models have been extensively studied for molecular tasks (Edwards et al., 2022; Zhao et al., 2023; Fang et al., 2024; Park et al., 2024; Liu et al., 2024; Ye et al., 2025; Nguyen & Grover, 2025; Hu et al., 2026). Classical approaches combine a molecular representation, a search operator, and property oracles (Olivecrona et al., 2017; Jensen, 2019; Gao et al., 2022), while recent LLM-based systems add medicinal-chemistry priors and tool interaction (Bran et al., 2023; Wang et al., 2025; Zhou et al., 2026; Li et al., 2026). Covalent inhibitors make this setting more restrictive because an electrophilic *warhead* mediates the intended covalent reaction after non-covalent positioning (Baillie, 2016; Boike et al., 2022; Du et al., 2021). Removing or replacing that source electrophile can change the mechanism even when conventional developability scores improve. Existing covalent resources and generators, including CovalentInDB (Du et al., 2021; 2025), CovaGEN (Zhang et al., 2026), and recent multi-objective covalent RL (Gil, 2026), mainly address de novo discovery. Generated molecules also struggle to balance multiple properties simultaneously; therefore, we study source-conditioned multi-objective optimization of covalent molecules: given a specific covalent molecule, find an effective local edit that improves QED, Lipinski compliance, synthetic accessibility (SA), and predicted toxicity while retaining the recognized source-warhead labels used by the evaluator.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

As illustrated in Figure 1, current multi-objective covalent molecular optimization suffers from the following problems. First, tool availability does not equate to strategic tool use: agents struggle to recover from failed edits and to decide when to stop, resulting in brittle search trajectories. For example, an untuned baseline agent fails on 75.21% of benchmark seeds, illustrating this brittleness. Second, multi-objective search often improves molecular properties at the cost of destroying the key warhead that determines the covalent mechanism, while heterogeneous property scales make coordinated improvement difficult. In structural diagnostics, the untuned baseline agent retains all recognized source warheads for only 26.69% of seeds; in the stage-wise development study, RL with the initial directly aggregated multi-property reward yields only 16.7% positive optimization. These failures motivate supervision that exposes executable local decision boundaries, as well as an optimization objective that explicitly protects the covalent mechanism.

To address the above problems, we propose the CovMOLAgent framework. As in Figure 2 and 3, first, we build a covalent molecule corpus by collecting molecules from CovalentInDB and constructing additional covalent molecules from ZINC by introducing diverse warheads (Zhang et al., 2026). Based on this, we construct the CETRA module, a trajectory supervision framework that consolidates 319,700 executed seed-local edits into 5,230 structured trajectories. Before language supervision, CETRA reconstructs trajectories containing real invalid trials, non-positive explorations, recovery, winner selection, and audited verification; a closed-source LLM verbalizes rationales around these immutable observations. We then refine the initialized policy through direction-normalized bounded property utilities and a source-warhead penalty, addressing the objective-mismatch problem. We construct CovaOpBench by collecting test molecules from CovalentInDB (Du et al., 2021) and further building a covalent molecular optimization benchmark based on ZINC (Irwin & Shoichet, 2005). On CovaOpBench, our framework achieves the highest overall improvement, together with the best empirical warhead-retention rate and the lowest failure rate. Furthermore, it delivers highly competitive performance on individual evaluation metrics compared with advanced closed-source large models, demonstrating its consistent superiority in multi-objective covalent optimization.

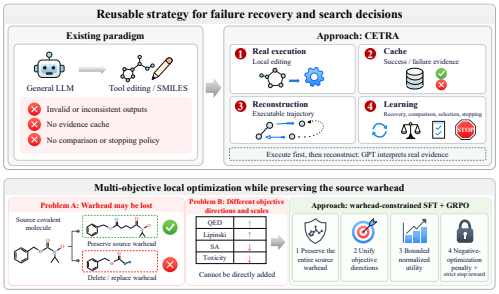

Figure 1: Motivation: (1) Tool availability does not ensure effective search behavior: agents may fail to recover from unsuccessful edits or to terminate appropriately. (2) Multi-objective optimization can improve molecular properties while losing the source warhead, and heterogeneous property scales complicate coordination.

We make three contributions. **First, CETRA: Execution-grounded, failure-aware trajectory supervision.** It reconstructs local editing operations into structured trajectories, explicitly encoding invalid trials, non-positive explorations, recovery, winner selection, and terminal verification, and grounds facts in deterministic tool executions, enabling the agent to learn to recover from failures and decide when to stop. **Second, WCPO: Warhead-aware multi-objective relative optimization.** Through a source-warhead penalty, direction-normalized bounded utilities, magnitude-aware negative penalties, and trajectory compliance validation, it favors source-label retention while improving multiple properties. **Third, strong empirical results under a common agent protocol.** On the constructed benchmark CovaOpBench, CovMOLAgent consistently outperforms the evaluated competing methods in aggregate improvement and structural fidelity. It achieves the highest improvement rate and the lowest failure rate, together with leading results on several individual optimization objectives.

2 Related Work

2.1 Multi-Objective Molecular Optimization and Covalent Design

Molecular optimization has been approached through reinforcement learning, graph-based search, evolutionary algorithms, diffusion models, and more recently LLM-guided search (Olivecrona et al., 2017; Jensen, 2019; Gao et al., 2022; Wang et al., 2025; Krüger et al., 2026; Gao et al., 2026). While

these methods increasingly address multiple, potentially competing molecular properties, covalent drug design introduces an additional structural requirement: optimization must preserve the reactive mechanism encoded by the source warhead. Existing covalent-design methods primarily focus on reactivity modeling or de novo generation, including warhead-guided diffusion and multi-objective reinforcement learning (Baillie, 2016; Du et al., 2021; 2025; Zhang et al., 2026; Gil, 2026). In contrast, we study source-conditioned covalent optimization, where retention of all recognized source-warhead labels is evaluated while simultaneously improving multiple molecular properties relative to the starting molecule.

2.2 Tool-Augmented LLM Agents for Molecular Optimization

Tool-augmented language models have emerged as a practical paradigm for grounding chemical reasoning in executable computation. ChemCrow integrates LLM reasoning with expert chemistry tools (Bran et al., 2023), ToolMol couples molecular editing with deterministic RDKit-backed execution (Zhou et al., 2026), and ChemCRAFT further explores post-training chemical agents with tool-integrated trajectories and chemistry-aware policy optimization (Li et al., 2026). These advances establish the utility of executable tools for reliable molecular reasoning, but leave open how supervision should encode the search process itself and how policy optimization should account for source-relative, heterogeneous objectives under covalent constraints. Our framework addresses these two aspects through execution-grounded, failure-aware trajectory construction and a warhead-preserving multi-objective relative optimization objective. A more detailed comparison and the original positioning table are provided in Appendix A.

3 Method

As in Figure 2, we present CovMOLAgent, a tool-augmented post-training framework for source-conditioned covalent molecule optimization. We first formalize the task, then introduce CETRA for execution-grounded trajectory supervision and WCPO for source-relative multi-objective policy optimization; implementation details are provided in the Appendix D and E.

3.1 Problem Formulation

We formulate covalent molecule optimization as source-conditioned local search over an executable chemistry environment. Let x_0 denote an immutable source molecule and let:

$$\tau = (t_1, a_1, o_1, \ldots, t_T, a_T, o_T) \tag{1}$$

denote a complete model–tool trajectory, where t_t is the thinking process of the model, a_t is the action selected at step t, o_t is the corresponding deterministic tool observation, and x_T is the molecule parsed from the terminal answer. The language-model policy π_θ generates both reasoning tokens and chemistry-tool calls conditioned on the complete interaction history.

Unlike de novo molecular generation, optimization is defined with respect to the supplied molecule x_0. We record two terminal evaluator predicates. Let $\mathrm{Valid}(x) \in \{0, 1\}$ indicate whether x can be successfully sanitized and canonicalized, and let $W(x)$ denote the set of covalent-warhead labels recognized by the evaluator. A terminal output is evaluated against:

$$\mathrm{Valid}(x_T) = 1, \qquad W(x_0) \subseteq W(x_T). \tag{2}$$

The second condition is explicitly source dependent: when multiple warhead labels are recognized in the source, every recognized label must remain in the terminal molecule at the warhead-label level. This is stronger than requiring the output to contain an arbitrary electrophilic motif.

The optimization objectives comprise QED, Lipinski compliance, synthetic accessibility (SA), and predicted toxicity. Because these properties differ in both optimization direction and numerical scale, we first express each change relative to the source and orient it such that positive values consistently denote improvement. Let $\mathbb{1}[\cdot]$ denote the indicator function and $\mathrm{Lip}(x) \in \{0, 1\}$ the implemented Lipinski-compliance indicator. We define:

$$\begin{aligned} d_{\mathrm{QED}}(x; x_0) &= \mathrm{QED}(x) - \mathrm{QED}(x_0), & d_{\mathrm{Lip}}(x; x_0) &= \mathbb{1}[\mathrm{Lip}(x)] - \mathbb{1}[\mathrm{Lip}(x_0)], \\ d_{\mathrm{SA}}(x; x_0) &= \mathrm{SA}(x_0) - \mathrm{SA}(x), & d_{\mathrm{Tox}}(x; x_0) &= \mathrm{Tox}(x_0) - \mathrm{Tox}(x). \end{aligned} \tag{3}$$

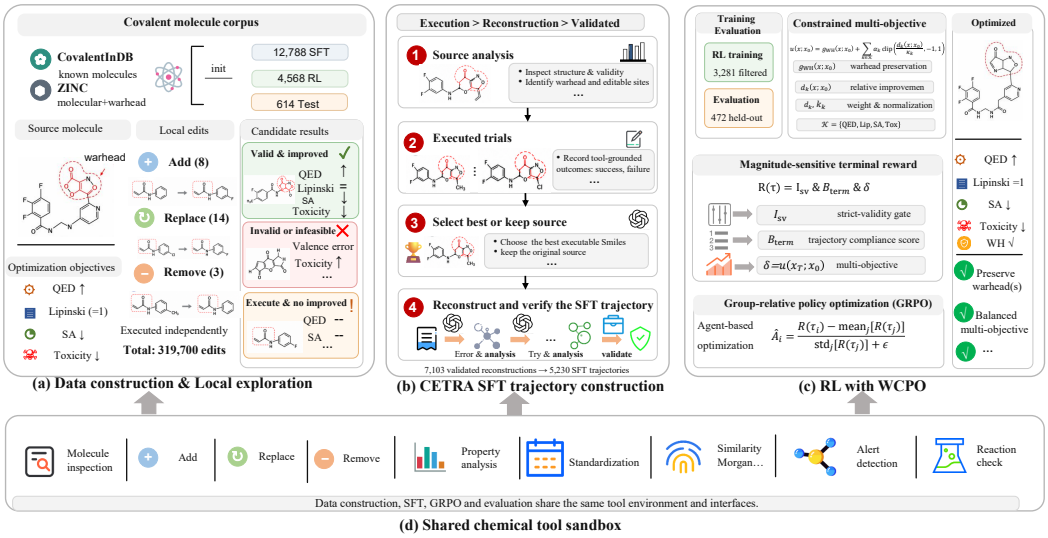

Figure 2: Overview of CovMOLAgent. Molecule-disjoint partitions are created before trajectory construction. CETRA executes the supported local-edit neighborhood and reconstructs failure-aware trajectories for SFT. WCPO then refines the policy on a disjoint RL split using bounded multi-property utility, terminal evaluator screening, and source-warhead retention diagnostics.

Accordingly, larger QED and improved Lipinski compliance yield positive changes, while reductions in SA and predicted toxicity are also mapped to positive values. We collect these quantities into the source-relative improvement vector:

$$\mathbf{d}(x; x_0) = \big(d_{\mathrm{QED}}, d_{\mathrm{Lip}}, d_{\mathrm{SA}}, d_{\mathrm{Tox}}\big). \tag{4}$$

Let $u(x; x_0)$ denote the scalar molecular utility constructed from these relative changes and defined in Section 3.3. The optimization problem can then be written as:

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x_0)} \left[u(x_T; x_0)\right] \\ \text{evaluated with:} \quad & \mathrm{Valid}(x_T) = 1, \qquad W(x_0) \subseteq W(x_T), \end{aligned} \tag{5}$$

where the expectation is taken over complete model–tool trajectories rather than unconstrained molecular strings; these predicates are scored by the terminal evaluator rather than implemented as an action mask. The full task definition, property orientation, and source construction procedure are provided in Appendix B, while the executable action space is detailed in Appendix C.

3.2 CETRA: Execution-Grounded SFT Trajectory Construction

Covalent molecule corpus. We first assemble the molecular corpus used throughout post-training and evaluation. The corpus combines curated covalent molecules collected from CovalentInDB (Du et al., 2021) with additional covalent molecules constructed from ZINC-derived molecular scaffolds (Irwin & Shoichet, 2005). Candidate molecules are sanitized, canonicalized, and deduplicated at the molecular level, resulting in 18,270 unique covalent molecules. Before trajectory construction, the corpus is partitioned into 12,788 SFT sources, 4,568 RL sources, and 614 held-out sources. CETRA operates exclusively on the SFT partition, ensuring molecule-level separation between supervised trajectory construction, subsequent policy optimization, and evaluation. Detailed data provenance, warhead grafting, preprocessing, and split construction are provided in Appendix B.

Executable local exploration. For each SFT source molecule x_0, CETRA instantiates a fixed set $\mathcal{A}(x_0)$ of one-step molecular edits covering functional-group addition, replacement, and removal. Each edit is independently executed from the original source molecule:

$$F(x_0, a) \to (x_a, o_a, e_a), \qquad a \in \mathcal{A}(x_0), \tag{6}$$

where x_a denotes the resulting molecule, o_a the complete tool observation, and e_a an explicit execution error when the edit fails. Independent execution preserves one-action attribution and prevents an unsuccessful intermediate state from propagating to later candidates.

Successful candidates are subsequently evaluated with the shared chemistry tools for molecular validity, source-relative property changes, structural similarity, warhead preservation, and toxicity-related signals. Let $H(x_a; x_0)$ denote the mining-time feasibility predicate, which jointly encodes successful execution, molecular validity, source-warhead preservation, structural locality, and toxicity-alert constraints. Using the source-relative molecular utility u introduced in Section 3.3, CETRA defines the positive executable neighborhood and its preferred candidate as:

$$\mathcal{P}(x_0) = \{a \in \mathcal{A}(x_0) : H(x_a; x_0) = 1,\ u(x_a; x_0) > 0\}, \qquad a^\star = \arg\max_{a \in \mathcal{P}(x_0)} u(x_a; x_0). \quad (7)$$

Thus, trajectory supervision is grounded in candidates that have been explicitly executed and evaluated, while candidate preference is aligned with the same source-relative optimization criterion used during RL. CETRA executes 319,700 local molecular edits over the SFT partition. The complete edit vocabulary, feasibility predicate, candidate records, and tie-breaking rules are provided in Appendices C and D.

GPT-4 trajectory construction. The key step in CETRA is to transform independently executed local outcomes into sequential optimization trajectories. For each source molecule, we first construct an immutable action–observation skeleton from the cached execution records. It contains source inspection, informative failed or non-improving trials, selection of the best positive candidate from the independently executed neighborhood, and terminal verification. We write this reconstruction abstractly as:

$$\bar{\tau}(x_0) = \tau_{\text{inspect}} \oplus \tau_{\text{trial}} \oplus \tau_{\text{select}} \oplus \tau_{\text{verify}}, \quad (8)$$

where τ_{trial} retains up to two informative unsuccessful or non-positive edits, and τ_{select} applies $a^\star$ when $\mathcal{P}(x_0) \neq \varnothing$. If no positive candidate is supported by the executed neighborhood, the source molecule is retained and explicitly verified instead.

GPT-4 (Achiam et al., 2023) is applied only after $\bar{\tau}(x_0)$ has been fixed. Given the ordered tool calls and their cached observations, GPT-4 constructs the reasoning that connects successive decisions:

$$\tau_{\text{SFT}} = G_{\text{GPT4}}\big(\bar{\tau}(x_0), \mathcal{E}(x_0)\big), \quad (9)$$

where $\mathcal{E}(x_0)$ denotes the complete executed evidence associated with the source. GPT-4 fills the EVIDENCE, INTERPRETATION, DECISION, and WHY fields for each retained step, while the executed actions, tool arguments, molecular states, numerical measurements, warhead labels, and terminal molecule remain immutable. In this way, GPT-4 constructs the reasoning over an already executed optimization process rather than determining the chemical outcomes themselves.

The synthesized trajectories are replayed against the cached execution records. Rationale–observation inconsistencies are repaired when they can be resolved deterministically; otherwise, the trajectory is discarded. This procedure produces 7,103 validated reconstructions. After balancing edited and source-retaining cases, 5,230 trajectories are retained for supervised fine-tuning. Detailed hard-case sampling, executable predicates, trajectory templates, rationale validation, balancing, and reconstruction examples are provided in Appendix D.

Given the resulting trajectory set $\mathcal{D}_{\text{SFT}}$, we optimize the assistant-token likelihood while treating tool observations as conditioning context:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{M} \sum_i \sum_t m_{i,t} \log p_\theta(y_{i,t} \mid h_{i,t}), \qquad M = \sum_i \sum_t m_{i,t}. \quad (10)$$

The complete SFT configuration is provided in Appendix E.

3.3 TRAINING AND EVALUATION CONSTRUCTION WITH WCPO

Training and evaluation construction. The RL and held-out partitions are processed independently from CETRA trajectory construction. Starting from the 4,568 RL sources, we apply the same molecular standardization and executable local-edit environment used for SFT data construction. We retain molecules whose supported edit neighborhood contains at least one valid candidate with

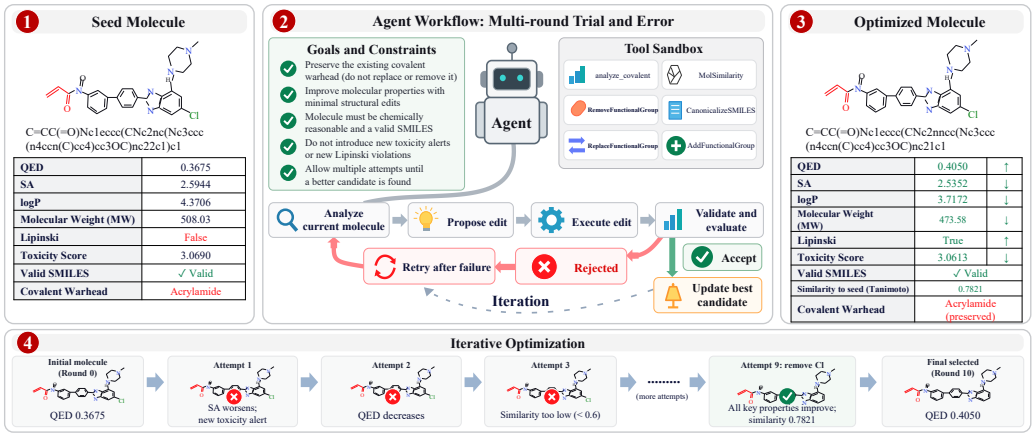

Figure 3: Implementation pipeline of CovMOLAgent. The framework uses executable chemistry tools for iterative molecular analysis, editing, verification, and multi-objective optimization.

positive source-relative utility, ensuring that every training instance admits a realizable improvement within the action space available to the policy. After filtering, 3,281 molecules remain for RL training. The held-out benchmark CovaOpBench consists of 472 positive-filtered, molecule-disjoint seeds constructed from the 614 held-out sources by the same task-construction procedure. Detailed filtering criteria, construction statistics, and disjointness checks are provided in Appendix B.

Source-warhead-aware multi-objective relative utility. WCPO evaluates a candidate molecule according to its improvement relative to the corresponding source rather than its absolute molecular-property values. Let $d_k(x; x_0)$ denote the direction-aligned source-relative change of objective k defined in Section 3.1, where positive values consistently indicate improvement.

To account for source-warhead preservation, we first define a source-warhead penalty:

$$g_{\mathrm{WH}}(x; x_0) = -\lambda_{\mathrm{WH}} \mathbb{1}\left[\mathrm{Valid}(x) = 0 \ \vee W(x_0) \not\subseteq W(x)\right]. \tag{11}$$

This term penalizes either an invalid molecular state or the loss of any warhead recognized in the source molecule. We then aggregate heterogeneous molecular objectives through direction-aligned and bounded components:

$$u(x; x_0) = g_{\mathrm{WH}}(x; x_0) + \sum_{k \in \mathcal{K}} \alpha_k \,\mathrm{clip}\left(\frac{d_k(x; x_0)}{\kappa_k}, -1, 1\right), \qquad \mathcal{K} = \{\mathrm{QED}, \mathrm{Lip}, \mathrm{SA}, \mathrm{Tox}\}. \tag{12}$$

Here, κ_k aligns heterogeneous property scales, while α_k specifies fixed objective weights. Direction alignment, scale normalization, and bounded aggregation place the different objectives on a comparable source-relative scale while incorporating source-warhead retention into the utility.

The trajectory reward is composed of three components: a terminal-validity screen I_{sv}, which credits well-formed terminal outputs, a format-compliance score that credits the required reasoning and tool-use structure, and the source-relative molecular utility $\delta = u(x_T; x_0)$, which is the only component that carries molecular information. Source-warhead retention is additionally assessed at the terminal step through:

$$I_{\mathrm{WH}}(\tau) = \mathbb{1}[W(x_0) \subseteq W(x_T)], \tag{13}$$

which records whether every recognized source label remains present. The source-warhead penalty in u discourages label loss, while the evaluator reports Full-WH and Edited-WH explicitly; thus the reported retention is an empirical outcome rather than a mechanism-level guarantee. The reward is monotone in δ and therefore preserves the ordering that GRPO normalizes within a rollout group. Appendix E gives the implementation and the distribution of δ over rollouts.

Policy optimization. Starting from the CETRA-supervised checkpoint, we use standard GRPO to optimize the trajectory-level WCPO reward.

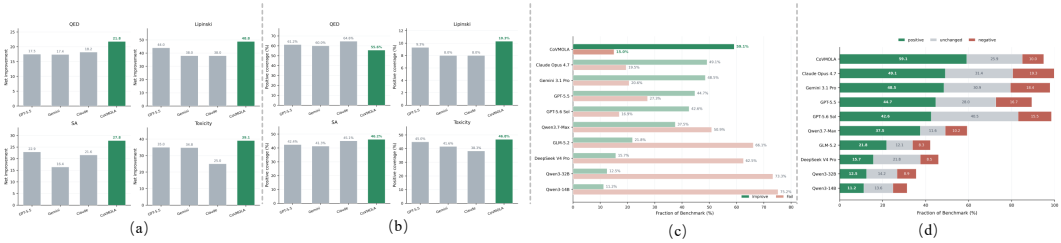

Figure 4: Visualization of method comparisons. Panels (a) and (b) show improvement magnitudes and positive rates across individual properties, respectively. Panels (c) and (d) summarize aggregate improvement and failure outcomes.

4 EXPERIMENTS

4.1 BENCHMARK AND METRICS

All methods are evaluated on CovaOpBench, which consists of 472 positive-filtered, molecule-disjoint seeds. In addition, we report the improvement scores and optimization success rates for QED, Lipinski, SA, and toxicity(Bickerton et al., 2012; Ertl & Schuffenhauer, 2009; Bento et al., 2020; Hao et al., 2026), as well as warhead retention rate, novelty rate, and other diagnostic metrics. Formal definitions of these metrics, the distinction between strict and post-hoc accounting, and the paired bootstrap procedure are provided in Appendix F. To verify that positive filtering identifies sources with executable optimization headroom, we evaluate the same checkpoint on all 614 held-out sources. The 142 excluded molecules have comparable terminal validity (97.2% versus 96.6%) and warhead preservation (89.4% versus 90.9%), but only 17.6% yield positive utility, compared with 59.1% among the retained 472 sources. Moreover, 97.2% of the excluded molecules already satisfy the Lipinski criteria, and the policy more frequently returns the source molecule when no supported improving edit exists. These results support the benchmark construction: positive filtering removes sources without an executable improving candidate while preserving molecule-disjoint evaluation and comparable structural validity (Appendix D.1).

4.2 BASELINES AND EXPERIMENTAL CONFIGURATION

The comparison includes frontier proprietary models—GPT-5.6 Sol (OpenAI, 2026), GPT-5.5 (Singh et al., 2025), Gemini 3.1 Pro (Google, 2026), and Claude Opus 4.7 (Anthropic, 2026)—as well as open-source models, including Qwen3-14B-Instruct, Qwen3-32B, DeepSeek-V4-Pro (Xu et al., 2026), GLM-5.2 (Zeng et al., 2026), and Qwen3.7-Max, together with domain-specific molecular optimization models, Chem-R and Ether0. All systems receive the same source molecule, optimization objective, state-machine instructions, and the same nine tool schemas and implementations. For each seed molecule, each system performs a single rollout with at most 20 assistant turns. The complete evaluation prompt is provided in Appendix G; the tool inventory, training hyperparameters, serving configuration details, and reproducibility controls are described in Appendices C, E, and F, respectively.

4.3 MAIN RESULTS

In Table 1 and Figures 4 and 5, CovMOLAgent achieves the strongest joint and aggregate optimization performance. It obtains the highest joint improvement rates for at least one, two, three, and four properties, as well as for the QED–SA–toxicity criterion. It also achieves the highest mean source-relative utility and overall improvement rate, together with the lowest failure rate. In particular, CovMOLAgent improves at least two properties on 54.03% of benchmark instances while satisfying the joint evaluation criterion, substantially outperforming all baselines.

Table 2 reports individual-property changes together with structural outcomes. CovMOLAgent achieves the largest net improvements in QED, Lipinski compliance, and toxicity, and the highest positive improvement rates for Lipinski compliance, synthetic accessibility, and toxicity. It also attains the highest Full-WH rate and exact-SMILES novelty. Under the same prompt, tool set, and turn

Table 1: Joint optimization results on CovaOpBench. $\mathrm{JIR}_{\geq k}$ denotes improvement in at least k properties, and $\mathrm{JIR}_{\mathrm{QST}}$ denotes joint improvement in QED, SA, and toxicity. Improve/Fail indicate positive-utility and unsuccessful instances, respectively.

Method	$\mathrm{JIR}_{\geq 4}$	$\mathrm{JIR}_{\geq 3}$	$\mathrm{JIR}_{\geq 2}$	$\mathrm{JIR}_{\geq 1}$	$\mathrm{JIR}_{\mathrm{QST}}$	$\overline{U}_{\mathrm{sc}}$	Improve (%)	Fail (%)
Closed-source models								
GPT-5.5	1.69	23.31	42.16	46.40	19.07	−0.14	44.70	27.33
Gemini 3.1 Pro	1.48	19.92	41.74	52.12	16.10	−0.12	48.52	20.55
Claude Opus 4.7	1.27	20.97	43.64	54.45	16.53	−0.12	49.15	19.49
GPT-5.6 Sol	1.48	23.09	40.04	43.64	19.28	−0.12	42.58	16.95
Qwen3.7-Max	1.69	20.34	35.38	38.77	14.62	−0.03	37.50	50.85
Open-source models								
Qwen3-14B-Instruct	0.00	4.87	10.59	13.14	4.03	−0.17	11.23	75.21
Qwen3-32B	0.21	5.08	11.65	15.04	4.24	−0.20	12.50	73.31
DeepSeek-V4-Pro	1.06	8.69	14.83	16.31	6.36	−0.19	15.68	62.50
GLM-5.2	0.64	11.65	20.34	22.03	9.53	−0.16	21.82	66.10
Domain-specific models								
Chem-R	0.00	0.21	1.48	2.97	0.21	–	0.00	100.00
BioMedGPT-LM-7B	0.00	0.00	0.00	0.00	0.00	–	0.00	100.00
Ether0	0.00	0.00	0.00	0.00	0.00	–	0.00	100.00
BioMistral-7B	0.00	0.00	0.00	0.00	0.00	–	0.00	100.00
Ours								
CovMOLAgent	**2.54**	**24.79**	**54.03**	**64.41**	**20.97**	**0.14**	**59.11**	**15.04**

Table 2: Property changes and structural outcomes on CovaOpBench. Each property reports the net direction-normalized change and positive improvement rate. Full-WH denotes retention of all recognized source warheads, and Novel denotes exact canonical-isomeric-SMILES novelty.

Method	QED	Lipinski	SA (↓)	Toxicity (↓)	Full-WH (%)	Novel (%)
Closed-source models						
GPT-5.5	17.55 / 61.23	44.00 / 9.32	22.78 / 42.37	34.95 / 44.92	80.93	61.23
Gemini 3.1 Pro	17.52 / 60.38	38.00 / 8.05	16.53 / 41.31	34.70 / 41.53	83.26	67.16
Claude Opus 4.7	18.20 / **64.62**	38.00 / 8.05	21.54 / 45.13	24.99 / 38.35	85.81	68.01
GPT-5.6 Sol	16.83 / 58.05	34.00 / 7.20	**28.12** / 43.01	35.15 / 39.62	85.38	57.42
Qwen3.7-Max	16.73 / 46.40	48.00 / 10.17	23.50 / 32.84	25.22 / 31.99	50.42	47.46
Open-weight models						
Qwen3-14B-Instruct	4.12 / 15.68	7.00 / 1.48	5.20 / 11.23	7.79 / 10.59	26.69	17.37
Qwen3-32B	6.18 / 18.22	12.00 / 2.75	11.98 / 14.62	6.98 / 12.08	29.45	21.61
DeepSeek-V4-Pro	6.92 / 23.09	19.00 / 4.03	8.31 / 16.74	14.24 / 17.37	38.14	24.15
GLM-5.2	9.54 / 29.24	18.00 / 3.81	11.82 / 21.19	19.22 / 20.76	34.11	30.08
Domain-specific models						
Chem-R	-0.04 / 2.97	0.00 / 0.00	-3.23 / 2.12	-1.35 / 2.54	10.38	5.72
Ether0	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00	0.00
BioMedGPT-7B	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00	0.00
BioMistral-7B	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00	0.00
Ours						
CovMOLAgent	**21.79** / 55.51	**49.00 / 10.38**	27.92 / **46.19**	**38.93 / 46.82**	**90.89**	**68.43**

budget, CovMOLAgent also outperforms GPT-5.6 Sol, Gemini 3.1 Pro, and Claude Opus 4.7 in joint improvement, mean source-relative utility, and Full-WH retention, while achieving a lower failure rate. These results highlight the benefit of task-specific trajectory supervision and source-relative optimization for improving molecular properties while preserving the covalent warhead. Together with its superior aggregate outcomes, these results demonstrate that CovMOLAgent provides a strong balance among multi-property optimization, execution reliability, and covalent structural fidelity.

4.4 COVALENT DOCKING CASE STUDY ON 5V9U

We validate structural compatibility on CovaOpBench using a representative seed mapped to PDB 5V9U. The optimized molecule retains the acrylamide warhead and differs by only a single defluorination, with an MCS covering 29/30 heavy atoms (96.67%). This edit improves QED (0.6405→0.6651) and SA (2.8693→2.6514), yielding a $+0.0710$ utility gain. Across five docking runs, the optimized molecule consistently outperforms the seed, with a shared-scaffold RMSD of 1.503 Å, supporting preservation of the original binding mode. Together with improved label-level warhead retention, this case provides additional evidence for the structural feasibility of CovMOLAgent's optimized molecules. Full docking details are provided in Appendix H.

4.5 ABLATION AND STAGE-WISE ANALYSIS

Table 3 compares training stages and reward configurations. Base denotes the backbone without task-specific post-training. RL-Init uses an unnormalized, directly aggregated multi-property reward. RL-Norm adds direction alignment and scale normalization to that reward. WCPO only further incorporates the source-warhead penalty and trains directly from the base model without CETRA initialization. CETRA only applies supervised fine-tuning on the reconstructed trajectories without subsequent RL refinement. The full framework combines CETRA initialization with WCPO refinement.

The reward-design comparisons show increasing improvement rates from RL-Init to RL-Norm and WCPO only (16.7%, 21.0%, and 27.3%, respectively). Relative to RL-Norm, WCPO only also increases Full-WH from 36.3% to 78.3%, supporting the importance of explicitly accounting for source-warhead preservation in the optimization reward. CETRA only achieves a 46.2% improvement rate and an 86.6% WHR. Subsequent WCPO refinement raises these rates to 59.1% and 90.9%, respectively, while reducing the failure rate from 44.7% to 15.0%. These comparisons support combining execution-grounded trajectory supervision with constrained policy refinement.

Table 3: Training-stage and reward-design comparisons on CovaOpBench. Improve denotes positive-utility outcomes, Fail denotes unsuccessful outcomes, and Full-WH denotes retention of all recognized source warheads.

Variant	Improve. ↑	Failure ↓	Full-WH ↑
Base	11.2	75.2	26.7
RL-Init	16.7	69.5	34.7
RL-Norm	21.0	62.5	36.3
WCPO only	27.3	58.1	78.3
CETRA only	46.2	44.7	86.6
CETRA + WCPO	**59.1**	**15.0**	**90.9**

As in Table 13, across three independent seeds, CovMOLAgent achieves stable performance ($54.45\% \pm 0.56\%$ $\text{JIR}_{\geq 2}$), confirming reproducible multi-objective gains. Figure 1 shows that natural recovery after failed edits is weak, while Table 3 demonstrates the stage-wise benefits of full CETRA supervision. Appendix I.5 further shows that CovMOLAgent outperforms random and greedy search and approaches the one-step oracle with fewer executions by adapting subsequent edits and stopping decisions from tool feedback. Finally, the Pareto analysis places CovMOLAgent on the non-dominated frontier for 74.36% of benchmark sources, jointly supporting CETRA's closed-loop, failure-aware use of executable evidence across multiple objectives.

5 CONCLUSION

CovMOLAgent enables multi-objective covalent molecule optimization with source-warhead preservation. CETRA provides execution-grounded supervision from local-edit outcomes, while WCPO coordinates multi-property optimization under structural constraints. Experiments show consistent gains in optimization, execution reliability, and warhead retention, supporting execution-grounded, constraint-aware post-training for covalent molecular optimization.

AI USE STATEMENT

Generative AI was used in two capacities. First, GPT-4 was used as part of CETRA to synthesize rationale fields around immutable action–observation records obtained from deterministic chemistry-tool executions. GPT-4 did not determine molecular actions, chemical states, measured property values, warhead labels, or terminal targets; these were fixed before rationale generation, and generated rationales were validated against the cached execution records. Second, generative AI tools were used to assist with language polishing. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content and results.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Introducing claude opus 4.7. 2026. URL `https://www.anthropic.com/news/claude-opus-4-7`.

Thomas A Baillie. Targeted covalent inhibitors for drug design. *Angewandte Chemie International Edition*, 55(43):13408–13421, 2016.

A Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J Bellis, Marleen De Veij, and Andrew R Leach. An open source chemical structure curation pipeline using rdkit. *Journal of cheminformatics*, 12(1):51, 2020.

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

Lydia Boike, Nathaniel J Henning, and Daniel K Nomura. Advances in covalent drug discovery. *Nature Reviews Drug Discovery*, 21(12):881–898, 2022.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

Hongyan Du, Junbo Gao, Gaoqi Weng, Junjie Ding, Xin Chai, Jinping Pang, Yu Kang, Dan Li, Dongsheng Cao, and Tingjun Hou. Covalentindb: a comprehensive database facilitating the discovery of covalent inhibitors. *Nucleic Acids Research*, 49(D1):D1122–D1129, 2021.

Hongyan Du, Xujun Zhang, Zhenxing Wu, Odin Zhang, Shukai Gu, Mingyang Wang, Feng Zhu, Dan Li, Tingjun Hou, and Peichen Pan. Covalentindb 2.0: an updated comprehensive database for structure-based and ligand-based covalent inhibitor design and screening. *Nucleic Acids Research*, 53(D1):D1322–D1327, 2025.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 375–413, 2022.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 48221–48251, 2024.

Rui Gao, Youngseung Jeon, Swastik Roy, Morteza Ziyadi, and Xiang'Anthony' Chen. C-moral: Controllable multi-objective molecular optimization with reinforcement alignment for llms. *arXiv preprint arXiv:2604.23061*, 2026.

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

Renee Gil. Multi-objective reinforcement learning for generating covalent inhibitor candidates. *arXiv preprint arXiv:2604.20019*, 2026.

Google. Gemini 3.1 pro: A smarter model for your most complex tasks. 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

Li Hao, He Cao, Bin Feng, Daniel Shao, Robert Tang, Zhiyuan Yan, Yonghong Tian, Li Yuan, and Yu Li. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *Advances in Neural Information Processing Systems*, 38, 2026.

Chengxin Hu, Hao Li, Yihe Yuan, Zezheng Song, Chenyang Zhao, and Haixin Wang. Omni-mol: Multitask molecular model for any-to-any modalities. *Advances in Neural Information Processing Systems*, 38:66945–66988, 2026.

John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

Jan H Jensen. 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.

Fabian P Krüger, Andrea Hunklinger, Adrian Wolny, Tim J Adler, Igor Tetko, and Santiago David Villalba. Seismo: Increasing sample efficiency in molecular optimization with a trajectory-aware llm agent. *arXiv preprint arXiv:2602.00663*, 2026.

Hao Li, He Cao, Shenyao Peng, Zijing Liu, Bin Feng, Yu Wang, Zhiyuan Yan, Yonghong Tian, Yu Li, and Li Yuan. Agentic reinforcement learning empowers next-generation chemical language models for molecular design and synthesis. *arXiv preprint arXiv:2601.17687*, 2026.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Conversational drug editing using retrieval and domain feedback. In *International Conference on Learning Representations*, volume 2024, pp. 48325–48361, 2024.

Tung Nguyen and Aditya Grover. Lico: Large language models for in-context molecular optimization. In *International Conference on Learning Representations*, volume 2025, pp. 53576–53597, 2025.

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

OpenAI. Gpt-5.6 system card. 2026. URL https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf.

Jinyoung Park, Minseong Bae, Dohwan Ko, and Hyunwoo J Kim. Llamo: Large language model-based molecular graph assistant. *Advances in Neural Information Processing Systems*, 37:131972–132000, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Haorui Wang, Marta Skreta, Cher-Tian Ser, Wenhao Gao, Lingkai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 51694–51727, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Wenbo Zhang, Tianxiao Liu, Xiaoying Dong, Saisai Sun, Xiaojun Yao, Pengyong Li, and Lin Gao. De novo covalent drug generation with enhanced drug-likeness and safety. *Communications Biology*, 9(1):446, 2026.

Haiteng Zhao, Shengchao Liu, Ma Chang, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in neural information processing systems*, 36:5850–5887, 2023.

Andrew Y Zhou, Sharvaree Vadgama, Sumanth Varambally, Peter Eckmann, Michael K Gilson, and Rose Yu. Toolmol: Evolutionary agentic framework for multi-objective drug discovery. *arXiv preprint arXiv:2605.12784*, 2026.

A Extended Related Work

A.1 Detailed Related-Work Positioning

The concise related-work discussion in the main paper focuses on the three closest methodological regimes. Table 4 provides a detailed presentation of the relevant work.

B Data Construction Details

This appendix provides the full data provenance, molecular preprocessing, source construction, molecule-disjoint partitioning, positive filtering, and construction statistics summarized in Sections 3.2 and 3.3.

B.1 Task definition

Let x_0 be a valid source molecule and let a tool trajectory $\tau = (a_1, o_1, \ldots, a_T, o_T)$ return a final molecule x_T. Let $W(x)$ be the set of warhead labels recognized in molecule x by the evaluator. The terminal evaluator predicates are:

$$\mathrm{Valid}(x_T) = 1, \qquad W(x_0) \subseteq W(x_T). \tag{14}$$

The set inclusion records whether every recognized source label remains, rather than accepting an output that contains at least one electrophile. In the current implementation, W is label-level SMARTS matching rather than atom-mapped identity.

The agent optimizes four source-relative properties. We define:

$$\begin{aligned} d_{\mathrm{QED}} &= \mathrm{QED}(x) - \mathrm{QED}(x_0), \qquad d_{\mathrm{Lip}} = \mathbb{1}[\mathrm{Lip}(x)] - \mathbb{1}[\mathrm{Lip}(x_0)], \\ d_{\mathrm{SA}} &= \mathrm{SA}(x_0) - \mathrm{SA}(x), \qquad d_{\mathrm{Tox}} = \mathrm{Tox}(x_0) - \mathrm{Tox}(x). \end{aligned} \tag{15}$$

Thus, positive values consistently denote improvement. QED is the quantitative estimate of drug-likeness; Lipinski is a binary rule-of-five indicator; SA is the Ertl–Schuffenhauer synthetic-accessibility score; and toxicity is the current local predictor with an alert-count fallback. The toxicity prediction tool uses Covagen's tool, which uses LD50 in the form of - log (mol/kg). The smaller the value, the lower the toxicity. These quantities are optimization proxies rather than experimental claims about efficacy or safety.

B.2 Seed sources and warhead grafting

As shown in Table 5, we collect curated covalent molecules from CovalentInDB and construct additional covalent molecules from the scaffold resources used by our preprocessing pipeline, including ZINC-derived molecular scaffolds. Four electrophilic motifs—acrylamide, cyanamide, nitrile, and methyl acrylate—are grafted onto compatible scaffolds using RDKit graph edits, followed by molecular sanitization and canonicalization. Among 20,518 attempted grafting operations, 10,000 valid molecules are retained, with 2,500 examples for each warhead class. Merging the constructed molecules with the curated covalent collection and deduplicating by canonical isomeric SMILES produces 18,270 unique source molecules.

The construction records also contain a natural-compound scaffold pool in addition to the virtual-screening scaffold collection.

B.3 Molecule-disjoint splits and positive filtering

Partitioning is performed before executable trajectory construction. Let $c(x)$ denote RDKit canonical isomeric SMILES and $c(\mathcal{D}) = \{c(x) : x \in \mathcal{D}\}$. We enforce pairwise disjointness among the SFT, RL, and held-out partitions:

$$c(\mathcal{D}_{\mathrm{SFT}}) \cap c(\mathcal{D}_{\mathrm{RL}}) = \varnothing, \qquad c(\mathcal{D}_{\mathrm{SFT}}) \cap c(\mathcal{D}_{\mathrm{hold}}) = \varnothing, \qquad c(\mathcal{D}_{\mathrm{RL}}) \cap c(\mathcal{D}_{\mathrm{hold}}) = \varnothing. \tag{16}$$

The initial partitions contain 12,788 SFT sources, 4,568 RL sources, and 614 held-out sources. CETRA is applied only to the SFT partition. For RL and evaluation, we additionally retain sources

for which the executable local-edit space contains at least one valid positive candidate:

$$\mathcal{D}^+ = \{x_0 \in \mathcal{D} : \mathcal{P}(x_0) \neq \varnothing\}. \tag{17}$$

Here $\mathcal{P}(x_0)$ denotes the positive executable candidate set defined in Appendix D. Positive filtering leaves 3,281 RL sources. For the held-out partition, task construction first yields 614 evaluable instances, of which 472 satisfy the positive-candidate criterion and constitute the final benchmark. Canonical-SMILES disjointness is checked again after filtering.

Positive filtering makes the final benchmark a capability-oriented evaluation: each retained source is known to admit at least one improving action under the supported edit vocabulary. This avoids conflating optimization ability with the absence of a feasible move in the restricted action space, while also limiting the benchmark's interpretation to molecules that are optimizable under that space.

Table 4: Positioning relative to representative neighboring methods. We distinguish de novo generation from fixed-seed editing. "Agent" denotes a post-trained policy that selects executable actions; an LLM used only as an inference-time operator is not counted as a learned agent. The final column summarizes how each method treats covalent structure. Detailed training signals are discussed in the surrounding text.

Method	Task regime	Optimization / agent	Covalent treatment
Gil	De novo covalent	SMILES LSTM + policy-gradient RL	Electrophile quality and target-related scores
CovaGEN	Target-conditioned de novo	Latent diffusion + RL	Classifier-guided warhead formation
ToolMol	De novo population search	MOGA + inference-time LLM operator	No source-warhead constraint
ChemCRAFT	General chemical tasks	Tool sandbox + SFT/GRPO agent	Generic structural / functional fidelity
C-MORAL	Molecular-property optimization	Molecular LLM + relative RL	Scaffold similarity, not covalent identity
Ours	Fixed-seed covalent editing	Trained nine-tool local-search policy (CETRA + WCPO)	Preserve every recognized source warhead

C Tool Environment and Executable Search Details

The main paper only states that the same nine chemistry tools are shared across construction, training, and evaluation. The complete tool inventory and the exact executable local-edit evidence construction follow.

C.1 Shared tool environment

The environment exposes nine functions. `analyze_covalent` returns validity, QED, SA, logP, molecular weight, Lipinski status, warhead labels, toxicity alerts, and a toxicity score. `FunctionalGroups` enumerates editable groups. The three graph editors are `AddFunctionalGroup`, `ReplaceFunctionalGroup`, and `RemoveFunctionalGroup`; each returns a sanitized candidate or an explicit error. `CanonicalizeSMILES` and `MolSimilarity` validate representation and locality. Finally, `predict_residue` and `predict_reaction` provide rule-based residue–warhead compatibility checks. Table 6 gives the complete inventory.

The same implementations and schemas are used during offline execution, SFT trajectory construction, RL rollouts, and baseline evaluation. This consistency matters: the model is never trained on an action that is unavailable at test time, and every numerical statement in an SFT trajectory can be replayed from a cached tool result.

Table 5: Data construction statistics. Counts are molecule-level unless marked as edits, trajectories, or constructed tasks

.

Stage	Count
Unique covalent source pool	18,270
SFT sources	12,788
Initial RL sources	4,568
Held-out sources	614
Executed SFT candidate edits	319,700
Positive candidate edits	21,989
SFT sources with ≥ 1 positive edit	7,959
Validated CETRA reconstructions	7,103
Final SFT trajectories	5,230
Filtered RL training sources	3,281
Constructed held-out tasks	614
Final benchmark sources	472

Table 6: Tools available to every agent. The same implementations are used for data construction, post-training, and evaluation.

Tool	Role	Principal output
`analyze_covalent`	Property and covalent analysis	validity, QED, SA, logP, molecular weight, Lipinski status, detected warheads, toxicity alerts, toxicity score
`FunctionalGroups (FG)`	Structure inspection	recognized editable functional groups and locations
`AddFG`	Molecular edit	sanitized candidate SMILES or explicit failure
`ReplaceFG`	Molecular edit	sanitized candidate SMILES or explicit failure
`RemoveFG`	Molecular edit	sanitized candidate SMILES or explicit failure
`predict_residue`	Mechanism prior	predicted reactive residue class, confidence, and reaction family
`predict_reaction`	Mechanism validation	reaction-valid flag and covalent-addition proxy
`CanonicalizeSMILES`	Representation check	canonical valid SMILES or failure
`MolSimilarity`	Locality check	Morgan-fingerprint Tanimoto similarity

D Detailed CETRA Construction

This appendix provides the execution and reconstruction details omitted from Section 3.2, including the local-edit vocabulary, cached candidate records, feasibility criteria, hard-case selection, GPT-4 trajectory construction, rationale validation, and keep-source balancing.

D.1 Effect of Positive Filtering on the Benchmark

The benchmark keeps only seeds whose tool-tested one-step neighborhood contains at least one strictly improving, warhead-preserving candidate. Because this is a selection on the seeds rather than on the method, we evaluate the same checkpoint on all 614 evaluable held-out seeds and compare the 472 retained seeds against the 142 the filter removed. The retained set is an exact subset of the 614, so the removed group is a set difference rather than a re-implementation of the filter. Both groups are scored under the same reward, prompt, tool set, decoding configuration, and label-level SMARTS warhead criterion used for Full-WH elsewhere in this paper.

The filter does not retain the easier molecules. Table 7 compares the source molecules themselves. The removed seeds are systematically *smaller* and *simpler*, and are in several respects already *better* developed: they carry higher QED (0.608 vs 0.417), lower molecular weight (320.2 vs 404.6), fewer heavy atoms (22.3 vs 28.4), lower logP (0.63 vs 1.38), fewer rotatable bonds (4.45 vs 6.73), and easier synthetic accessibility (3.37 vs 3.58). Most strikingly, 97.2% of them already satisfy Lipinski's rule of five against 69.9% of the retained seeds. The filter therefore removes molecules that are

Table 7: Source-molecule properties for the 472 retained benchmark seeds and the 142 seeds removed by positive filtering, all recomputed with RDKit under one criterion. The removed seeds are smaller, simpler, and already more compliant; the 472 benchmark is the harder and less drug-like part of the held-out space. SA and logP are lower-is-better.

Property	Retained (472)	Removed (142)	Difference
QED	0.417	0.608	+0.191
SA (lower is easier)	3.575	3.366	−0.210
Molecular weight	404.6	320.2	−84.4
Heavy atoms	28.4	22.3	−6.2
logP	1.38	0.63	−0.75
TPSA	114.6	101.7	−12.8
Rotatable bonds	6.73	4.45	−2.28
Lipinski pass rate	69.9%	97.2%	+27.3 pp

Table 8: Terminal outcomes on the 472 retained seeds and the 142 seeds removed by positive filtering. The retained column is the system reported in Section 4; both groups use the same checkpoint, reward, prompt, and decoding configuration. Source return counts terminal molecules identical to the source.

Outcome	Retained (472)	Removed (142)
Valid terminal output	456/472 (96.6%)	138/142 (97.2%)
Positive utility	279/472 (59.1%)	25/142 (17.6%)
Warhead preserved	429/472 (90.9%)	127/142 (89.4%)
Returns source molecule	122/448 (27.2%)	93/138 (67.4%)

already close to the top of the developability scales—precisely the regime in which no single local edit can improve one property without damaging another. The retained set is the larger, less drug-like, harder-to-satisfy part of the held-out space. The number of recognized warheads is essentially unchanged, so the filter is not a warhead-difficulty selector either.

Behaviour when no improving move exists. Table 8 reports the terminal outcomes of our system on both groups. Three findings matter. First, output reliability is unchanged: the strict-valid terminal rate is 97.2% on the removed seeds against 96.6% on the retained seeds, so the policy does not degrade when the oracle says the seed is unimprovable. Second, warhead retention is likewise preserved (89.4% vs 90.9%), so the stronger result on the benchmark is not obtained by compromising molecular fidelity more often. Third, and most directly relevant, the behaviour that distinguishes a usable system is visible only in this group: on seeds with no improving move the policy returns the source molecule 67.4% of the time, against 27.2% on improvable seeds. Abstention is thus the dominant behaviour exactly where no improving move exists, which is the behaviour a deployed optimizer needs and which a positive-filtered benchmark cannot test on its own.

The system is also not merely idle on these seeds: it finds improvements on 25 of the 142 removed seeds (17.6%), and all 25 are genuine structural changes rather than source returns. A one-step oracle over a fixed edit vocabulary is not an upper bound on a multi-turn search policy, and these are cases where the policy recovered an improvement the oracle did not enumerate.

Scope. Measuring the unfiltered group bounds the effect of the filter in both directions. It confirms that the reported improvement rate is measured where an improving move exists, and it shows that this part of the held-out space is the more complex and less drug-like one in the recorded comparison. This observation does not make the benchmark representative of unrestricted covalent space; it only describes the relationship between the retained and excluded subsets. It also reveals behaviour the filtered numbers cannot: the policy abstains rather than degrades when no improving move is available, and it still recovers improvements on a minority of seeds a one-step oracle rejects. The positive filter is thus best read as a construction criterion that every training instance admits a realizable improvement—which GRPO's within-group advantage normalization requires—rather than as a device that raises the reported numbers.

D.2 EXECUTABLE LOCAL-EDIT EVIDENCE

For every SFT source, we instantiate 25 independent local edits: eight additions, fourteen replacements, and three removals. The vocabulary covers small substituents and common medicinal-chemistry transformations, including fluoro, hydroxyl, methyl, amide, carboxyl, nitrile, pyridine, piperidine, halo removal, and benzene-to-pyridine replacement. Each edit is applied directly to the original source rather than to the product of a preceding rejected edit. This preserves one-action attribution and prevents a later successful action from obscuring the origin of an earlier failure.

Let $\mathcal{A}(x_0)$ denote the supported actions instantiated for source x_0. The executor returns:

$$F(x_0, a) \to (x_a, o_a, e_a), \tag{18}$$

where x_a is the resulting molecule, o_a is the complete tool observation, and e_a records an explicit error state. If execution fails, $x_a = \perp$. Each successful candidate is subsequently re-analyzed and compared with its source, and we cache:

$$v_a = (a, x_a, o_a, e_a, u_a, s_a, W_a, A_a), \tag{19}$$

where $u_a = u(x_a; x_0)$ is the WCPO molecular utility, s_a is Morgan-fingerprint Tanimoto similarity, $W_a = W(x_a)$ is the detected warhead set, and A_a is the toxicity-alert count. The complete local evidence set is:

$$\mathcal{E}(x_0) = \{v_a : a \in \mathcal{A}(x_0)\}. \tag{20}$$

Mining uses the conservative feasibility criterion:

$$H(v_a) = \mathbb{1}[e_a = \varnothing]\mathbb{1}[\mathrm{Valid}(x_a)]\mathbb{1}[W(x_0) \subseteq W_a]\mathbb{1}[s_a \geq \tau]\mathbb{1}[A_a \leq A(x_0)], \qquad \tau = 0.6. \tag{21}$$

The executable positive set and its supervised winner are:

$$\mathcal{P}(x_0) = \{v_a : H(v_a) = 1,\ u_a > 0\}, \qquad v^*(x_0) = \arg\max_{v_a \in \mathcal{P}(x_0)} u_a. \tag{22}$$

If $\mathcal{P}(x_0)$ is empty, the source molecule becomes the terminal target; otherwise, the molecule stored in v^* is selected. Across 319,700 executed edits, 21,989 satisfy the positive criterion, and 7,959 SFT sources contain at least one positive action.

The source-similarity threshold and toxicity-alert condition are used only for offline candidate mining. SA and Lipinski changes remain part of the multi-objective utility rather than becoming additional feasibility filters.

D.3 FAILURE-AWARE TRAJECTORY RECONSTRUCTION

CETRA reconstructs a compact sequential trajectory from the independently executed evidence set. Let $\mathcal{N}(x_0)$ contain execution failures, infeasible candidates, and feasible candidates with non-positive utility. From this set, we retain at most two informative hard cases before the terminal decision. The resulting skeleton:

$$\pi(x_0) = \pi_{\mathrm{inspect}}(x_0) \oplus \mathrm{Hard}_{\leq 2}(\mathcal{N}(x_0)) \oplus \pi_{\mathrm{terminal}}(x_0), \tag{23}$$

where π_{inspect} contains analysis of the source molecule and its editable functional groups. The terminal branch is:

$$\pi_{\mathrm{terminal}}(x_0) = \begin{cases} (a^*, x^*, o^*) \oplus \pi_{\mathrm{verify}}(x^*), & \mathcal{P}(x_0) \neq \varnothing, \\ \pi_{\mathrm{verify}}(x_0) \oplus \mathrm{return}(x_0), & \mathcal{P}(x_0) = \varnothing. \end{cases} \tag{24}$$

Verification includes re-analysis of molecular properties, canonicalization, source similarity, and covalent-compatibility checks. A keep-source trajectory therefore arises from an executed unsuccessful search whenever such evidence is available, rather than from an unconditional preference for copying the input.

D.4 GPT-4 RATIONALE SYNTHESIS AND VALIDATION

After the action–observation skeleton is fixed, it is provided to GPT-4 together with the corresponding cached execution records. GPT-4 constructs four rationale fields around each retained observation:

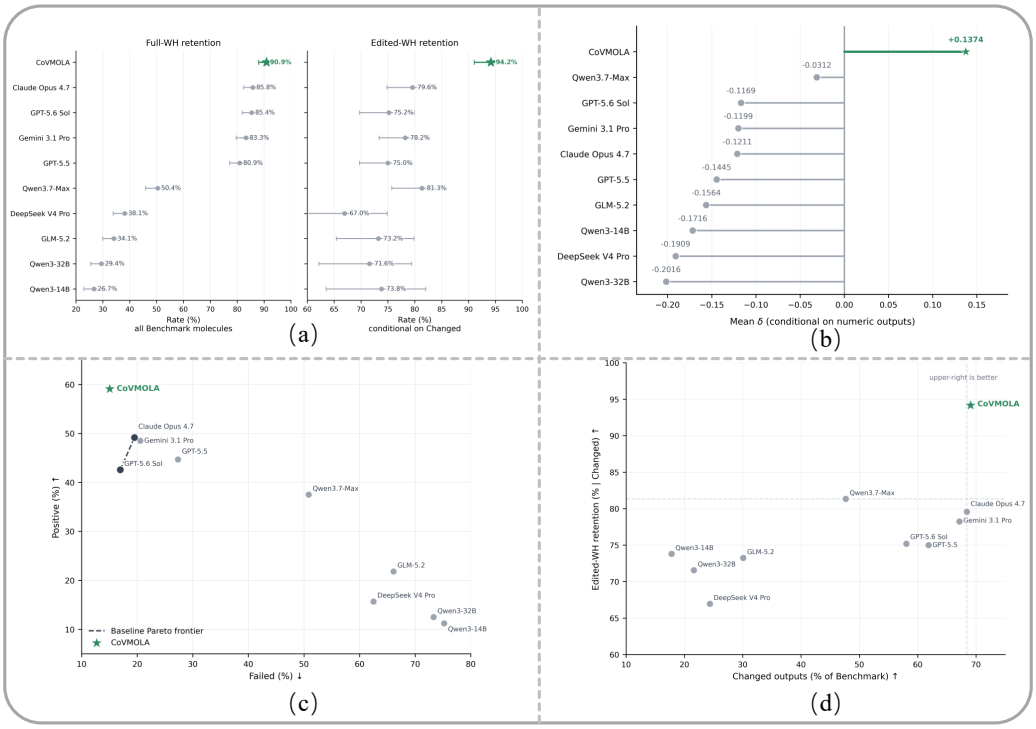

Figure 5: Performance comparison of CovMOLAgent and baseline methods across optimization, structural fidelity, and execution outcomes.

EVIDENCE, which summarizes the directly observed tool result; INTERPRETATION, which relates that result to the optimization objectives and constraints; DECISION, which determines whether the candidate should be retained or rejected; and WHY, which connects the decision to the next search action.

GPT-4 is not permitted to modify the executable state of the trajectory. In particular, the molecular SMILES, selected actions, tool arguments, property values, similarity scores, detected warheads, error messages, and terminal target are frozen before language generation. The model therefore constructs the explanatory and decision-making context around an already executed search process rather than generating chemical outcomes itself.

The generated rationale is subsequently validated against the corresponding tool observations. Required fields and tool references are checked automatically. Rationale text that conflicts with the cached execution result is repaired deterministically when the intended interpretation is recoverable; examples whose molecular state cannot be reconciled are removed. A subsequent paraphrasing pass varies the linguistic realization while leaving all actions, observations, measurements, SMILES, and terminal answers unchanged.

This procedure produces 7,103 valid trajectories, of which 4,446 terminate in an edited molecule and 2,657 retain the source. Because the unbalanced set would make source copying disproportionately frequent during SFT, we constrain the retained-source fraction. If N_e and N_k denote the edited and keep-source counts, respectively, we require:

$$N_k \leq \left\lfloor \frac{0.15}{0.85} N_e \right\rfloor . \tag{25}$$

For $N_e = 4,446$, this retains 784 keep-source trajectories and produces the final set of 5,230 SFT examples.

Algorithm 1 CETRA reconstruction. GPT-4 is applied only after execution and terminal selection; chemical states and measured outcomes are fixed before rationale synthesis.

Input: source x_0, actions $\mathcal{A}(x_0)$, executor F.
Inspect x_0 and cache source properties and warheads.
Execute every $a \in \mathcal{A}(x_0)$ independently and record either the resulting candidate or the explicit failure.
Evaluate feasibility and source-relative utility; construct the positive and non-positive candidate sets.
Select up to two informative failed or non-positive trials.
if a feasible edit exists **then**
    Append the highest-ranked feasible edit.
else
    Retain the source.
end if
Append canonicalization, similarity, and covalent-compatibility verification.
Use GPT-4 to construct rationale fields around the immutable execution records and validate their consistency.
Output: execution-grounded SFT trajectory $\tau(x_0)$.

E Training Objectives and Hyperparameters

E.1 Training configuration

We full-parameter fine-tune Qwen3-14B-Instruct on the 5,230 CETRA trajectories for three scheduled epochs with maximum sequence length 12,288, learning rate 10^{-5}, cosine decay, 3% warmup, bfloat16 precision, and DeepSpeed ZeRO-3. The RL run is initialized from checkpoint 1700, corresponding to epoch 2.603 of this schedule. GRPO uses the 3,281 positive-filtered RL seeds, learning rate 10^{-6}, 10% warmup, training batch size 64, PPO mini-batch size 32, per-GPU micro-batch size 1, and eight rollouts per prompt. Sampling uses temperature 0.6, top-$p = 0.9$, and top-$k = 20$. Each rollout is limited to 20 assistant turns, 8,192 response tokens, and a 25,000-token model context. Table 10 records the full configuration.

Range of the source-warhead penalty. Solving for a warhead violation inside u would require a positive property contribution exceeding $\lambda_{\mathrm{WH}} = 2$. Across 456 recorded rollouts, the largest simultaneously attained values of the four weighted property terms sum to 1.53, against a nominal cap of 3.5. QED carries the largest coefficient, $\alpha_{\mathrm{QED}} = 2$, and its realized contribution never exceeds 6.1% of that ceiling, since a single seed-local edit cannot raise the absolute QED score by an amount close to $\kappa_{\mathrm{QED}} = 1$. The observed range indicates that $\lambda_{\mathrm{WH}} = 2$ gives a strong preference for retaining source-warhead labels. It is still a reward preference rather than an atom-mapped or mechanism-level guarantee; the benchmark therefore reports Full-WH and Edited-WH as empirical diagnostics. The coefficients are $\lambda_{\mathrm{WH}} = 2$, $\alpha_{\mathrm{QED}} = 2$, $\alpha_{\mathrm{Lip}} = 0.3$, $\alpha_{\mathrm{SA}} = 0.2$, and $\alpha_{\mathrm{Tox}} = 1$.

E.2 WCPO Utility and Terminal-Reward Implementation

Fixed multi-objective utility. For the molecular utility in Eq. 12, we use a single fixed configuration across all RL training instances. The source-warhead penalty coefficient is:

$$\lambda_{\mathrm{WH}} = 2.$$

The objective weights are

$$\alpha_{\mathrm{QED}} = 2, \qquad \alpha_{\mathrm{Lip}} = 0.3, \qquad \alpha_{\mathrm{SA}} = 0.2, \qquad \alpha_{\mathrm{Tox}} = 1,$$

and the scale factors are

$$\kappa_{\mathrm{QED}} = \kappa_{\mathrm{Lip}} = \kappa_{\mathrm{Tox}} = 1, \qquad \kappa_{\mathrm{SA}} = 2.$$

Accordingly, the implemented utility is:

$$\begin{aligned} u(x; x_0) = g_{\mathrm{WH}}(x; x_0) &+ 2\ \mathrm{clip}\left(d_{\mathrm{QED}}, -1, 1\right) + 0.3\ \mathrm{clip}\left(d_{\mathrm{Lip}}, -1, 1\right) \\ &+ 0.2\ \mathrm{clip}\left(\frac{d_{\mathrm{SA}}}{2}, -1, 1\right) + \mathrm{clip}\left(d_{\mathrm{Tox}}, -1, 1\right). \end{aligned} \tag{26}$$

Table 9: Empirical reachability of the positive utility range over 456 recorded rollouts. For each utility component, the observed maximum is compared with its nominal positive cap. The final row sums the component-wise maxima; this is an optimistic upper estimate because the maxima need not occur in the same molecule.

Utility term	Nominal cap	Observed max.	Cap reached
$2\,\mathrm{clip}(d_{\mathrm{QED}}, -1, 1)$	2.0	0.1222	6.1%
$0.3\,\mathrm{clip}(d_{\mathrm{Lip}}, -1, 1)$	0.3	0.3000	100%
$0.2\,\mathrm{clip}(d_{\mathrm{SA}}/2, -1, 1)$	0.2	0.1067	53.4%
$\mathrm{clip}(d_{\mathrm{Tox}}, -1, 1)$	1.0	1.0000	100%
Sum	**3.5**	**1.5289**	**43.7%**

These values are treated as fixed utility-calibration constants throughout RL training. Their role is to place direction-aligned property changes with heterogeneous numerical ranges on stable and comparable contribution scales. Each component is additionally bounded to limit the influence of extreme changes in any single property.

Empirical reachability of the utility range. The nominal positive cap of the utility in Eq. 26 is

$$\alpha_{\mathrm{QED}} + \alpha_{\mathrm{Lip}} + \alpha_{\mathrm{SA}} + \alpha_{\mathrm{Tox}} = 2 + 0.3 + 0.2 + 1 = 3.5,$$

which exceeds the source-warhead penalty $\lambda_{\mathrm{WH}} = 2$ only in a purely formal sense. To determine how much of this range is actually reachable, we recorded the per-property utility terms over 456 rollouts and examined the maximum value attained by each term independently. As summarized in Table 9, even under this deliberately optimistic accounting—which would require a single molecule to simultaneously attain the observed maximum of every property term—the four component maxima sum to only 1.5289, corresponding to 43.7% of the nominal positive cap.

QED is the binding term: its largest observed positive change is only $+0.0611$, which yields a maximum weighted contribution of $2 \times 0.0611 = 0.1222$ and reaches just 6.1% of its nominal cap. Moreover, QED improves in only 28 of the 456 recorded rollouts, indicating that a seed-local edit cannot approach the $\kappa_{\mathrm{QED}} = 1$ scale required to saturate this component. The two components that individually reach their nominal caps, Lipinski and toxicity, also rarely improve together, co-occurring positively in only 1 of 456 rollouts.

More importantly, the component-wise sum in Table 9 is not itself attained by any observed molecule. The best single rollout reaches only $u = 1.1733$, corresponding to 33.5% of the nominal positive cap of 3.5, with the toxicity component accounting for approximately 85% of that utility. Thus, the source-warhead penalty $\lambda_{\mathrm{WH}} = 2$ exceeds the largest observed positive molecular utility, providing a strong empirical preference for source-warhead retention.

Fixed calibration. All numerical values above are used as fixed calibration constants throughout RL training rather than as separate optimization objectives. The calibration follows three general principles: terminal validity and protocol compliance provide only limited auxiliary signals; molecular improvement is primarily determined by the source-relative utility; and all utility and reward contributions are bounded to stabilize policy optimization.

Terminal reward composition. The trajectory reward combines three components:

- a terminal-validity screen I_{sv}, which records whether the rollout ends with exactly one terminal answer that canonicalizes to a valid molecule;
- a format-compliance score, which credits the required reasoning structure, tool-call protocol, and terminal-answer syntax;
- the source-relative molecular utility $\delta = u(x_T; x_0)$.

The protocol terms are independent of the returned molecule and take bounded values, providing only limited auxiliary signal. The molecular signal enters solely through δ, which is bounded: nominally $\delta \in [-5.5, 3.5]$, and on CovaOpBench it remains within $[-2.1, 1.95]$. The reward is a bounded, monotone function of δ, so the ordering of terminal outcomes is the same under the reward as under

Table 10: Training configuration supported by the current experiment records.

Component	Setting	Value
Base policy	Architecture	Qwen3-14B-Instruct
SFT	Data / scheduled epochs	5,230 trajectories / 3 epochs
SFT	RL initialization checkpoint	step 1700, epoch 2.603
SFT	Maximum sequence length	12,288
SFT	Learning rate / schedule	10^{-5} / cosine
SFT	Warmup / precision / sharding	3% / bfloat16 / DeepSpeed ZeRO-3
GRPO	Training seeds	3,281
GRPO	Learning rate / warmup	10^{-6} / 10%
GRPO	Batch / PPO mini-batch / micro-batch	64 / 32 / 1 per GPU
GRPO	Rollouts per prompt	8
GRPO	Assistant-turn / response / context limits	20 / 8,192 / 25,000
GRPO	Sampling	temperature 0.6, top-$p = 0.9$, top-$k = 20$
GRPO	Clipping / KL coefficient / PPO epochs	$\varepsilon = 0.2$ / $\beta = 0.001$ / 1
Evaluation	Selected checkpoint / rollouts	step 150 / one rollout per seed

u. Source-warhead retention is recorded at the terminal step through the indicator:

$$I_{\mathrm{WH}}(\tau) = \mathbb{1}[W(x_0) \subseteq W(x_T)]\,,$$

which is reported alongside the reward; the source-warhead term g_{WH} inside u continues to shape the utility landscape, while Full-WH and Edited-WH quantify the resulting empirical retention.

Accordingly, WCPO follows the computation:

$$\mathbf{d}(x; x_0) \longrightarrow u(x; x_0) \longrightarrow R(\tau),$$

where $\mathbf{d}(x; x_0)$ captures source-relative property changes, $u(x; x_0)$ combines multi-objective improvement with source-warhead preservation, and $R(\tau)$ converts the resulting molecular utility into a bounded trajectory-level reinforcement learning signal.

The terminal evaluator applies a fixed sequence of checks. First, it verifies that the rollout contains exactly one parseable and chemically valid terminal molecule. It then checks whether all recognized source-warhead labels are retained. Next, the evaluator records format and protocol compliance and computes the source-relative molecular utility $\delta = u(x_T; x_0)$ from the four direction-normalized molecular objectives. For terminal outputs that pass both the validity and source-warhead checks, the format score is combined with a transformation of δ to produce the final terminal reward, which is lower-bounded at zero. Because a severely deteriorated yet warhead-preserving molecule may also receive zero reward, the evaluator prevents outputs with source-warhead loss from receiving positive reward but does not impose a strict ordering when both outputs are mapped to zero. We therefore report source-warhead retention independently through Full-WH and Edited-WH. All evaluator constants are fixed before RL training and remain unchanged across experiments; they are not tuned for individual molecules or baselines.

F Detailed Evaluation Protocol

The main experiment section reports only the shared benchmark, metrics, baselines, and high-level configuration. This appendix retains the original research questions, strict evaluation accounting, paired uncertainty procedure, serving burden, and reproducibility controls.

F.1 Research questions

We organize the evaluation around four questions. **RQ1:** Does post-training a 14B tool policy improve fixed-seed covalent optimization relative to untuned open models and frontier API agents under the same interface? **RQ2:** Are gains distributed across QED, Lipinski, SA, and toxicity, rather

than produced by one objective? **RQ3:** Does the method improve source-warhead fidelity while still making non-trivial and novel edits? **RQ4:** What evidence do the available training-stage runs provide, and which causal ablations remain unresolved?

F.2 Evaluation accounting

Let $N = 472$ and let $\delta_i = u(x_i; x_{0,i})$ be the source-relative utility for example i. The strict scored subset and its size are:

$$\mathcal{S} = \{i : \delta_i \in \mathbb{R}\}, \qquad N_{\mathrm{sc}} = |\mathcal{S}|, \tag{27}$$

where membership requires successful parsing by the strict online evaluator. We report:

$$\overline{U}_{\mathrm{sc}} = \frac{1}{N_{\mathrm{sc}}} \sum_{i \in \mathcal{S}} \delta_i, \tag{28}$$

$$\mathrm{ScoredRate} = \frac{N_{\mathrm{sc}}}{N}, \tag{29}$$

$$\mathrm{ImproveRate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\delta_i > 10^{-12}], \tag{30}$$

$$\mathrm{FailRate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[i \notin \mathcal{S} \ \vee \ \delta_i < -10^{-12}], \tag{31}$$

$$\mathrm{NonImproveRate} = 1 - \mathrm{ImproveRate}. \tag{32}$$

The scored-output mean is conditional on successful parsing; missing utilities are not imputed as zero. We therefore present it only beside scored coverage and fixed-N outcome rates. "Failed" combines unscorable and numerically negative outcomes, while the unscorable subset is also shown separately. Non-improvement additionally contains unchanged outputs.

For property $k \in \{\mathrm{QED}, \mathrm{Lip}, \mathrm{SA}, \mathrm{Tox}\}$, let $\mathcal{V}_k$ contain examples with a computable source–final pair. We report the net direction-normalized change and fixed-denominator positive rate:

$$\Delta_k = \sum_{i \in \mathcal{V}_k} d_{i,k}, \tag{33}$$

$$PosRate_k = \frac{1}{N} \sum_{i \in \mathcal{V}_k} \mathbb{1}[d_{i,k} > 10^{-12}]. \tag{34}$$

Raw units differ across properties, so Δ_k is compared across systems only within the same property. Structural metrics are: (i) complete source-warhead retention over all 472 seeds; (ii) retention conditional on a structural edit; (iii) fixed-N structural-change coverage; and (iv) exact canonical-isomeric-SMILES novelty relative to the union of RL-training and benchmark source molecules.

For property and structure diagnostics, we additionally apply the documented post-hoc final-SMILES extractor and recompute molecular quantities whenever a valid molecule can be recovered. A response can therefore be unscorable online yet contribute a parseable source–final pair to a property table. We keep strict and post-hoc accounting separate.

F.3 Paired uncertainty

For proposed system A and baseline B, let $\mathcal{C}_{AB}$ contain seeds with numeric online utility from both systems and $M_{AB} = |\mathcal{C}_{AB}|$. Bootstrap replicate b resamples paired complete cases:

$$I_j^{(b)} \overset{\mathrm{iid}}{\sim} \mathrm{Unif}(\mathcal{C}_{AB}), \tag{35}$$

$$D^{(b)} = \frac{1}{M_{AB}} \sum_{j=1}^{M_{AB}} \left(\delta^A_{I_j^{(b)}} - \delta^B_{I_j^{(b)}} \right). \tag{36}$$

We use 50,000 replicates and random seed 20260807; 95% intervals are the 2.5th and 97.5th percentiles. Pair counts are always reported because complete-case analyses condition on different subsets. These intervals quantify variation across benchmark molecules, not repeated-generation uncertainty.

F.4 SERVING BURDEN

In the API harness, one assistant turn corresponds to one remote model request. Preserved logs contain 5,284 GPT-5.5 requests, 3,484 Gemini requests, and 4,138 Claude requests for the 472-seed evaluation, excluding chemistry-tool calls. CovMOLAgent performs local policy inference and makes no remote model request. We do not convert these counts to monetary cost because prices, caching, hidden reasoning tokens, and serving configurations are not controlled.

F.5 REPRODUCIBILITY CONTROLS

Canonical isomeric SMILES define molecule identity. The final checkpoint produces one saved rollout for every benchmark seed. Every table specifies whether its denominator is N, the strict scored subset, the post-hoc parseable subset, or the edited subset. We report exact counts together with percentages wherever the denominator changes.

G EVALUATION PROMPT

All systems are evaluated with the same molecule-specific instruction and the same tool interface. The source SMILES is substituted for `{SOURCE_SMILES}`. The full evaluation prompt template corresponding to the recorded state machine is reproduced below; tool input/output schemas are documented separately in Appendix C.

```
You are a covalent lead-optimization agent. Starting from the supplied
source molecule, search only through the available chemistry tools and
return one final SMILES.

SOURCE_SMILES: {SOURCE_SMILES}

Goal
- Improve the molecule jointly with respect to QED, Lipinski compliance,
  synthetic accessibility (SA), and predicted toxicity.
- Preserve every warhead label detected in the source molecule.
- Make a conservative local edit rather than redesigning the molecule.

Required procedure
1. Your first tool call must be analyze_covalent on SOURCE_SMILES.
2. Record the source warhead labels and treat all of them as mandatory.
3. Inspect editable groups before proposing an edit.
4. Use only the supported AddFunctionalGroup, ReplaceFunctionalGroup,
   and RemoveFunctionalGroup operations for structural edits.
5. Treat a tool error, invalid molecule, or unchanged "successful" edit
   as a failed attempt. Do not continue editing from a rejected product.
6. Reject a candidate if:
   - any source warhead label is missing;
   - Morgan/Tanimoto similarity to the source is below 0.60;
   - it introduces a new Lipinski violation; or
   - it introduces a new toxicity alert.
7. After every successful structural edit, verify the candidate with:
   analyze_covalent, CanonicalizeSMILES, and MolSimilarity.
8. Use predict_residue and predict_reaction when checking whether the
   preserved warhead remains compatible with the intended covalent
   reaction family.
9. Maintain the best verified candidate seen so far. A later trial must
   not overwrite it unless the verified molecular objective improves.
10. If a trial is invalid or non-improving, recover to the best verified
    candidate and choose a different supported action.
11. Stop after two consecutive edit failures, or when no supported edit
    is expected to improve the best verified candidate.
```

Table 11: Covalent docking validation on the 5V9U benchmark case. Lower tethered AutoDock scores are better within this fixed paired setup. The score is a relative ranking quantity, not an absolute covalent binding free energy.

Measure	Seed	Optimized	Change
QED	0.6405	0.6651	+0.0246
SA (↓)	2.8693	2.6514	−0.2179
Minimum tethered score	-9.88	-10.62	-0.74
Mean tethered score	-9.524	-10.418	-0.894
Median tethered score	-9.65	-10.61	-0.96
MCS coverage of seed	29/30 heavy atoms		96.67%
Seed–optimized MCS RMSD	1.503 Å		–
Seed redocking RMSD	2.209 Å		–

```
12. If no acceptable improvement is found, return the source molecule.

Reasoning format
After each tool observation, explicitly provide:
Evidence: the relevant facts from the immediately preceding observation.
Interpretation: what those facts imply for validity, utility, locality,
                toxicity, and warhead preservation.
Decision: the next tool action, candidate verification, recovery, or stop.
Why: why this decision is preferable to the available alternatives.

Final answer format
- End with exactly one terminal block and no additional SMILES after it:
  <answer>FINAL_SMILES</answer>
- FINAL_SMILES must be the canonical SMILES of the best verified candidate.
```

The prompt deliberately specifies a state machine rather than an answer heuristic: all systems must inspect the source, edit through deterministic graph tools, verify accepted candidates, maintain a best-so-far state, recover after failed trials, and terminate in a machine-parseable format. The inserted source SMILES and tool observations are treated as data fields; they cannot modify the protocol or add instructions. This mirrors the executable protocol learned by CETRA while keeping the evaluation interface identical across trained and untrained baselines.

H Covalent Docking Validation Details

This appendix documents the structural validation summarized in Section 4.4. The experiment is deliberately orthogonal to the optimization reward: docking is not used during training, candidate selection, or benchmark scoring. It is used only to ask whether one successful benchmark edit remains structurally plausible in the crystallographic covalent environment from which the seed can be exactly identified. Figure 6 provides a detailed visualization of this case.

H.1 Case identity, selection, and molecular edit

The selected molecule is benchmark index 173 (line 174 in the recorded JSONL) and is part of the final 472-seed test benchmark but not the training set. Its canonical isomeric source SMILES exactly matches the crystallographic ligand in PDB 5V9U. The target is GTPase KRas, the reactive residue is Chain A Cys12, the crystallographic ligand is Chain A residue 204 (ligand code 91S), the database reaction type is Michael addition, and the crystal structure resolution is 1.38 Å. Case selection was fixed before docking and required a positive benchmark utility, valid and structurally changed seed/final molecules, source-warhead preservation, an available PDB structure, and exact source-to-crystal-ligand matching.

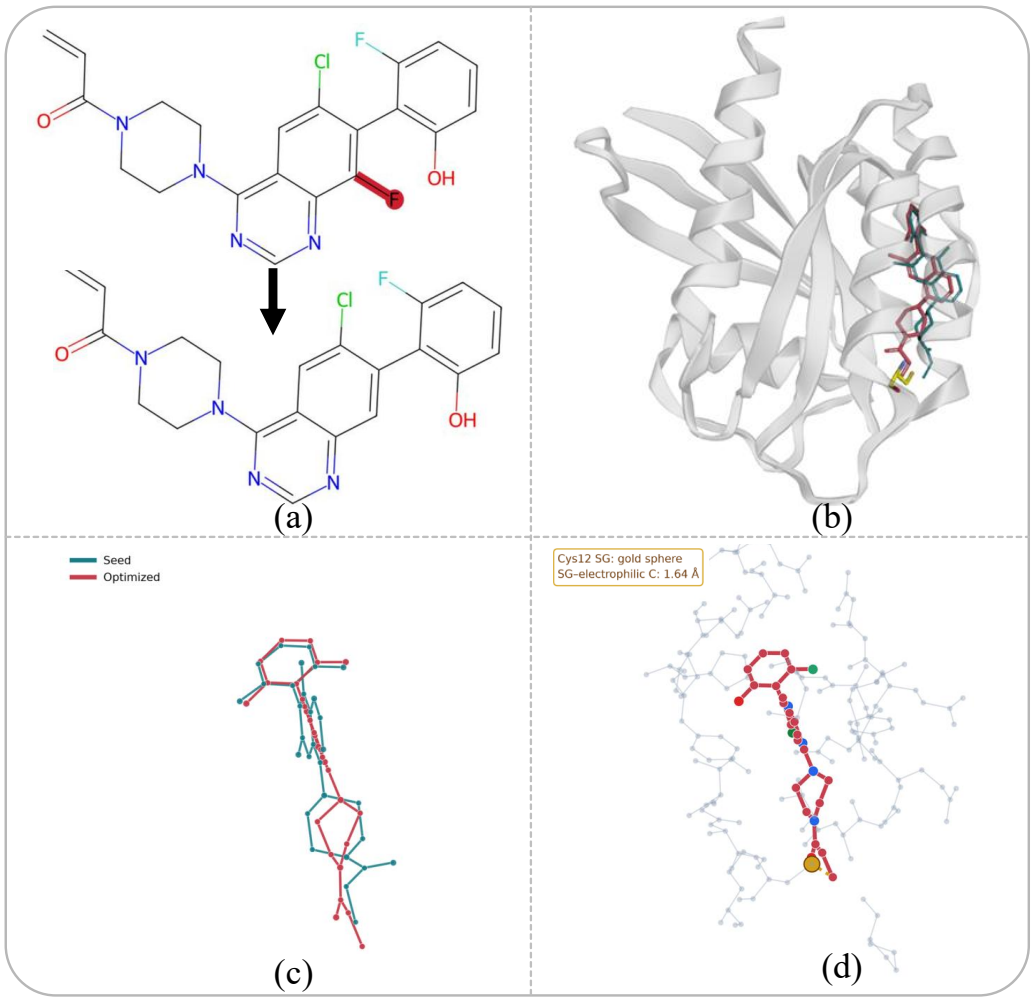

Figure 6: Covalent docking case study. Panel (a) compares the source and optimized molecules, panel (b) shows the three-dimensional target structure, and panels (c) and (d) show the corresponding two- and three-dimensional molecular views.

The source and optimized SMILES are

seed: `C=CC(=O)N1CCN(c2ncnc3c(F)c(-c4c(O)cccc4F)c(Cl)cc23)CC1`,
final: `C=CC(=O)N1CCN(c2ncnc3cc(-c4c(O)cccc4F)c(Cl)cc23)CC1`.

The only graph edit removes one fluorine from the fused heteroaromatic core. The acrylamide `C=CC(=O)N` warhead, the principal fused scaffold, the second aryl fluorine, chlorine substituent, hydroxy-substituted aryl group, and ring-linked amine region are retained. The seed has 30 heavy atoms, the final molecule has 29, and the maximum common substructure (MCS) contains 29 heavy atoms, corresponding to 96.67% coverage of the source and 100% of the optimized molecule. The benchmark molecular similarity is 0.7143.

Re-evaluation with the same benchmark analyzer gives QED $0.6405 \rightarrow 0.6651$, SA $2.8693 \rightarrow 2.6514$, unchanged Lipinski pass status, and unchanged toxicity estimate 2.734099. The corresponding reward contributions are $+0.0492$ (QED), 0 (Lipinski), $+0.0218$ (SA), and 0 (toxicity), for total source-relative utility $+0.0710$. Both molecules are recognized as containing the acrylamide, carbonyl, and Michael-acceptor labels.

Table 12: Five paired LGA searches for the 5V9U case. Lower is better within the fixed tethered setup.

LGA run	Seed	Optimized	Optimized − seed
1	-9.88	-10.09	-0.21
2	-8.97	-10.62	-1.65
3	-9.34	-10.61	-1.27
4	-9.65	-10.61	-0.96
5	-9.78	-10.16	-0.38
Minimum	-9.88	-10.62	-0.74
Mean	-9.524	-10.418	-0.894
Median	-9.65	-10.61	-0.96
Std. dev.	0.370	0.269	–

H.2 Paired tethered covalent docking protocol

The paired design fixes the protein environment so that the structural comparison changes only the ligand. Both seed and optimized molecule use PDB 5V9U, Chain A, Cys12, the same crystallographic pocket, the same docking box, and the same AutoDock4 grids. The docking box is obtained from the crystallographic ligand bounding box with 8.0 Å padding. We follow the two-point tethered/flexible-sidechain covalent-docking formulation used by AutoDock-family workflows, in which the Cys C_α/C_β atoms provide positional anchors and the remaining side chain plus covalent ligand are treated as a flexible unit.

For each molecule, the preparation script recognizes the acrylamide pattern `C=C-C(=O)N`, converts the reactive double bond to the Michael-addition single-bond state, and constructs a C_α–C_β–S–ligand adduct. The C_α and C_β coordinates correspond to Cys12 and S denotes the cysteine sulfur. Seed and optimized molecule use the identical anchoring rule. Meeko prepares the rigid receptor and flexible residue representation; AutoGrid4 generates a shared set of receptor affinity maps. AutoDock-GPU v1.6 then runs a Lamarckian genetic algorithm with five searches per ligand, random seed 20260820, automatic stopping, and global-best output. The paired protocol is therefore

$$\text{same receptor/anchor/box/grid/search settings}, \qquad \text{seed} \rightarrow \text{optimized ligand only}. \tag{37}$$

This is a relative within-PDB comparison. In the tethered flexible-residue representation, the reported DLG ranking quantity is the AutoDock tethered score (recorded through the final total internal-energy term); it is not interpreted as an absolute binding free energy or as a covalent reaction-rate estimate.

H.3 Pose restoration, MCS mapping, and RMSD

Docked poses are exported from the AutoDock-GPU logs with Meeko. For the lowest-score pose, the artificial C_α–C_β–S anchors used for tethering are removed and the Michael-adduct single bond is restored to the original acrylamide double bond. The restored graph is checked for isomorphism to the corresponding input molecule before pose comparison.

The seed lowest-score pose is compared with the crystallographic ligand in the fixed receptor coordinate system and gives a heavy-atom RMSD of 2.209 Å. Because the seed and optimized graphs differ by one atom, their pose comparison uses an RDKit maximum common substructure with element, bond-order, and complete-ring matching. If the MCS contains N mapped heavy atoms with seed coordinates $\mathbf{x}_i^{\text{seed}}$ and optimized coordinates $\mathbf{x}_i^{\text{opt}}$, we compute:

$$\text{RMSD}_{\text{MCS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i^{\text{seed}} - \mathbf{x}_i^{\text{opt}}\right\|^2}. \tag{38}$$

No additional ligand-to-ligand superposition is performed because both poses are already expressed in the same receptor coordinate frame. Here $N = 29$, covering 96.67% of seed heavy atoms, and $\text{RMSD}_{\text{MCS}} = 1.503$ Å.

Table 13: Independent-seed reproducibility of CovMOLAgent. Each run corresponds to an independently sampled random seed under the same training and evaluation configuration. All joint rates use the fixed benchmark denominator. The final row reports mean ± sample standard deviation over the three independent runs.

Method	$\mathrm{JIR}_{\geq 4}$	$\mathrm{JIR}_{\geq 3}$	$\mathrm{JIR}_{\geq 2}$	$\mathrm{JIR}_{\geq 1}$	$\mathrm{JIR}_{\mathrm{QST}}$	$\overline{U}_{\mathrm{sc}}$	Improve (%)	Fail (%)
CovMOLAgent (random seed 1)	2.54	24.79	54.03	64.41	20.97	0.137374	59.11	15.04
CovMOLAgent (random seed 2)	2.33	25.42	55.08	63.56	22.03	0.120596	60.17	12.71
CovMOLAgent (random seed 3)	1.48	23.31	54.24	66.10	20.13	0.104089	61.23	14.62
CovMOLAgent (mean ± SD)	2.12 ± 0.56	24.51 ± 1.09	54.45 ± 0.56	64.69 ± 1.29	21.05 ± 0.96	0.120686 ± 0.016643	60.17 ± 1.06	14.12 ± 1.24

H.4 Interpretation and limitations

The 5V9U example supports a narrow structural claim: a one-atom defluorination that improves QED and SA while preserving the acrylamide warhead also yields consistently lower tethered AutoDock scores under a paired fixed-pocket setup, without a large rearrangement of the shared heavy-atom scaffold. It does *not* establish stronger experimental binding, faster covalent reaction kinetics, or an absolute free-energy improvement. The seed redocking RMSD of 2.209 Å is slightly above the frequently used 2 Å pose-recovery reference, so the case is better interpreted as relative structural consistency than as high-accuracy crystallographic pose reproduction. More extensive docking across benchmark targets, covalent geometry analysis, reactivity calculations, and experimental validation remain future work.

I Additional Experimental Results

I.1 Independent-Seed Reproducibility

To assess the reproducibility of CovMOLAgent, we conduct three independent training and evaluation runs with different random seeds. All runs use the same model initialization protocol, molecule-disjoint data partitions, training hyperparameters, prompt, tool interface, decoding configuration, and evaluation benchmark.

We report the joint improvement rate $\mathrm{JIR}_{\geq k}$, defined as the percentage of benchmark seeds that preserve all recognized source warheads and strictly improve at least k of the four objectives: QED, Lipinski, SA, and toxicity. $\mathrm{JIR}_{\mathrm{QST}}$ requires simultaneous improvement in QED, SA, and toxicity and excludes Lipinski. All rates use the fixed benchmark denominator. $\overline{U}_{\mathrm{sc}}$ is the mean utility over strictly numeric outputs, while Improve and Fail use the full benchmark denominator. The final row reports the mean ± sample standard deviation over the three independent random seeds.

Table 13 shows that CovMOLAgent maintains a consistent multi-objective optimization profile across independent random seeds. The rate of improving at least two properties while preserving all source warheads is $54.45\% \pm 0.56\%$, and the stricter QST criterion reaches $21.05\% \pm 0.96\%$. The positive optimization rate remains high across all runs, with a mean of $60.17\% \pm 1.06\%$, while the failure rate remains limited to $14.12\% \pm 1.24\%$.

The strict scored utility is positive in every independent run and has a mean of 0.120686 ± 0.016643. Together, the relatively small variation in the joint improvement rates, positive optimization rate, and failure rate indicates that the observed advantage of CovMOLAgent is reproducible across random seeds rather than being attributable to a single favorable run.

I.2 Complete-case paired comparison

The complete-case paired analysis in Table 14 yields a positive mean difference against every baseline, and every percentile interval remains above zero. Against the four frontier API systems, the paired mean advantage is 0.259–0.295. Pair counts range from 144 to 447 because strict parsing failures differ by system.

Table 14: Complete-case paired utility difference, CovMOLAgent minus each baseline. B/T/W gives the number of paired seeds on which CovMOLAgent is better, tied, or worse. Intervals use 50,000 paired bootstrap replicates.

Baseline	Pairs	Mean	95% CI	B/T/W
GPT-5.5	403	0.2953	[0.2248, 0.3689]	188/109/106
Gemini 3.1 Pro	439	0.2714	[0.2031, 0.3406]	222/96/121
Claude Opus 4.7	447	0.2627	[0.1962, 0.3325]	218/111/118
GPT-5.6 Sol	441	0.2590	[0.1931, 0.3276]	200/140/101
Qwen3-14B-Instruct	144	0.2631	[0.1430, 0.3894]	70/40/34
Qwen3-32B	162	0.3189	[0.2081, 0.4343]	89/40/33
DeepSeek-V4-Pro	210	0.3733	[0.2715, 0.4779]	122/43/45
GLM-5.2	191	0.3156	[0.2037, 0.4312]	92/44/55
Qwen3.7-Max	268	0.1900	[0.1036, 0.2807]	104/70/94

Table 15: Covalent fidelity, structural change, and exact-SMILES novelty. Full-WH, Changed, and Novel use denominator 472; Edited-WH is conditional on Changed. Full retention requires every recognized source label to occur in the final molecule.

Method	Full-WH	Edited-WH	Changed	Novel
GPT-5.5	382/472 (80.93)	219/292 (75.00)	292/472 (61.86)	289/472 (61.23)
Gemini 3.1 Pro	393/472 (83.26)	248/317 (78.23)	317/472 (67.16)	317/472 (67.16)
Claude Opus 4.7	405/472 (85.81)	257/323 (79.57)	323/472 (68.43)	321/472 (68.01)
GPT-5.6 Sol	403/472 (85.38)	206/274 (75.18)	274/472 (58.05)	271/472 (57.42)
Qwen3-14B-Instruct	126/472 (26.69)	62/84 (73.81)	84/472 (17.80)	82/472 (17.37)
Qwen3-32B	139/472 (29.45)	73/102 (71.57)	102/472 (21.61)	102/472 (21.61)
DeepSeek-V4-Pro	180/472 (38.14)	77/115 (66.96)	115/472 (24.36)	114/472 (24.15)
GLM-5.2	161/472 (34.11)	104/142 (73.24)	142/472 (30.08)	142/472 (30.08)
Qwen3.7-Max	238/472 (50.42)	183/225 (81.33)	225/472 (47.67)	224/472 (47.46)
Chem-R	49/472 (10.38)	20/27 (74.07)	27/472 (5.72)	27/472 (5.72)
Ether0	0/472 (0.00)	-	0/472 (0.00)	0/472 (0.00)
BioMedGPT-7B	0/472 (0.00)	-	0/472 (0.00)	0/472 (0.00)
BioMistral-7B	0/472 (0.00)	-	0/472 (0.00)	0/472 (0.00)
CovMOLAgent	**429/472 (90.89)**	**307/326 (94.17)**	**326/472 (69.07)**	**323/472 (68.43)**

I.3 Covalent fidelity, edit coverage, and novelty

Table 15 answers RQ3. Over all 472 seeds, CovMOLAgent retains every recognized source warhead in 429 outputs (90.89%). Conditional on actually changing the molecule, it retains all source warheads in 307 of 326 outputs (94.17%), compared with 75.00–79.57% for the four frontier API systems and 66.96–81.33% for the additional baselines.

High retention is not achieved by simply echoing the source. CovMOLAgent changes 326 structures (69.07%), the largest edit coverage in the comparison. Of all inputs, 323 (68.43%) produce an exact-SMILES-novel final molecule relative to RL-training and benchmark sources. Three changed outputs are not novel under this reference set. These results support the intended combination of active local editing and source-mechanism preservation, although exact-SMILES novelty remains a weak notion of chemical novelty.

I.4 Failure analysis

The selected checkpoint emits 456 non-empty final strings; 448 are RDKit-valid and strictly scoreable. Of the 448 valid molecules, 429 retain all source warheads. There are 326 actual structural changes, 307 of which retain all source warheads. The two most visible remaining failure modes are therefore output reliability (24 unscorable examples) and mechanism loss after an otherwise valid edit (19

examples). Grammar-constrained decoding or graph-native terminal actions could address the former, while atom-mapped warhead identity and a lexicographic constraint could strengthen the latter.

The reward also has an intentional saturation regime: SA improvements beyond two raw units and toxicity improvements beyond one raw unit receive no additional utility. This stabilizes training but discards magnitude information. The raw property totals in Table 2 are therefore necessary for interpreting the clipped aggregate score.

Label-level structural fidelity. Full-WH is the primary source-label measure: it is the fraction of all 472 benchmark outputs that are RDKit-valid and retain every warhead label recognized in the corresponding source. Among the 326 outputs that actually change the molecular graph, Edited-WH is $307/326 = 94.17\%$. These two complementary rates separate overall execution reliability from preservation conditional on making an edit. Because the evaluator uses SMARTS labels rather than atom-mapped subgraphs, they provide an explicit and reproducible structural diagnostic, not a guarantee of atom identity, attachment context, covalent geometry, or reaction kinetics.

I.5 Closed-Loop Adaptation versus Local-Search Controls

To quantify the contribution of closed-loop adaptation, we replay the executable mining cache for the same 472 benchmark sources against controls that lack state-dependent recovery. CovMOLAgent conditions each subsequent tool choice on the latest molecular measurements and failure feedback, can change its edit strategy after an invalid or non-improving result, and makes a state-dependent stopping decision. The controls instead apply a predefined candidate-selection or fixed-neighborhood rule to the same executable evidence.

Each source has 25 independently recorded one-step edit attempts, including valid, rejected, and failed executions. A candidate is not considered improved unless it is valid, retains every recognized source-warhead label, and strictly improves the required property subset. These predicates are applied before computing every joint-improvement rate.

We compare three controls. **Random-**3 samples three candidates without replacement and returns the best feasible candidate observed; its values are averaged over 1,000 fixed-seed permutations. **Greedy-**3 scans three candidates in the serialized tool-enumeration order and keeps the best source-relative utility observed so far. **One-step neighborhood oracle** evaluates all 25 recorded edits and selects the highest feasible utility. It is therefore an upper bound for the one-step vocabulary, not a global oracle and not a budget-matched competitor.

One candidate evaluation is charged four executable calls: edit, analysis, canonicalization, and similarity. This gives 12 calls for Random-3 and Greedy-3, and 100 calls for the one-step oracle. CovMOLAgent is reported with its measured mean of 12.2 executable calls per source. We normalize cost by this value, so CovMOLAgent has relative cost $1.00\times$, the 12-call controls have $0.98\times$, and the exhaustive oracle has $8.20\times$ the executable cost. The corresponding absolute call counts are retained for reproducibility.

Table 16 reports only joint multi-objective outcomes. $\mathrm{JIR}_{\geq k}$ is the percentage of sources that satisfy the strict validity and source-warhead-retention predicates while improving at least k of QED, Lipinski, SA, and toxicity. $\mathrm{JIR}_{\mathrm{QST}}$ requires simultaneous improvement in QED, SA, and toxicity. All rates use the fixed denominator of 472.

CovMOLAgent exceeds both non-adaptive controls on all four reported joint-improvement criteria at approximately the same normalized cost. The one-step oracle is retained only to quantify the remaining gap to exhaustive one-step enumeration, whose executable cost is more than eight times larger.

I.6 Reward Sensitivity Analysis

To further understand why CovMOLAgent achieves reliable multi-objective covalent optimization, we conduct a detailed sensitivity analysis of the WCPO utility. The analysis examines whether the learned policy outcomes are supported by a coherent multi-objective preference structure, rather than by a single dominant property or an arbitrary coefficient choice. We independently vary the warhead constraint, property coefficients, and normalization scales while retaining the remaining

Table 16: Joint multi-objective comparison between the adaptive agent and non-adaptive local-search controls. All entries are percentages over the fixed 472-source benchmark. Relative cost is normalized by the measured CovMOLAgent cost of 12.2 executable calls per source. The corresponding absolute costs are 12, 12, 100, and 12.2 calls. Random-3 reports mean $\pm$ permutation standard deviation; the one-step oracle is an exhaustive 100-call upper bound.

Method	Relative cost	$\mathrm{JIR}_{\geq 4}$	$\mathrm{JIR}_{\geq 3}$	$\mathrm{JIR}_{\geq 2}$	$\mathrm{JIR}_{\mathrm{QST}}$
Random-3	$0.98\times$	0.18 ± 0.18	4.02 ± 0.85	14.01 ± 1.46	3.36 ± 0.81
Greedy-3	$0.98\times$	0.00	8.90	29.03	8.90
One-step neighborhood oracle	$8.20\times$	1.06	20.55	55.08	16.95
CovMOLAgent	$1.00\times$	**2.54**	**24.79**	54.03	**20.97**

Table 17: Detailed reward-sensitivity analysis of WCPO. Pos. and Neg. denote the numbers of positive and negative utility outcomes under each utility configuration. ρ_s measures Spearman rank agreement with the full WCPO utility.

Utility setting	Pos.	Neg.	ρ_s
Full utility			
Full WCPO utility	279	47	1.000000
Constraint and objective necessity			
$\lambda_{\mathrm{WH}} = 0$ (remove source-warhead penalty)	297	29	0.777973
$\lambda_{\mathrm{WH}} = 1$ (weaken source-warhead penalty)	292	34	0.881625
$\alpha_{\mathrm{QED}} = 0$	253	73	0.815271
$\alpha_{\mathrm{Lip}} = 0$	277	49	0.978286
$\alpha_{\mathrm{SA}} = 0$	279	47	0.988986
$\alpha_{\mathrm{Tox}} = 0$	247	79	0.689279
Local robustness around the selected setting			
$\lambda_{\mathrm{WH}} \in \{3, 4\}$	279	47	0.999999
$\alpha_{\mathrm{QED}} \in \{1, 3\}$	277–286	40–49	0.960–0.967
$\alpha_{\mathrm{Lip}} \in \{0.15, 0.6\}$	279–281	45–47	0.991–0.997
$\alpha_{\mathrm{SA}} \in \{0.1, 0.4\}$	280–282	44–46	0.994–0.995
$\alpha_{\mathrm{Tox}} \in \{0.5, 1.5\}$	277–279	47–49	0.961–0.976
$\kappa_{\mathrm{SA}} \in \{1, 4\}$	280–282	44–46	0.994–0.995
$\kappa_{\mathrm{Tox}} = 2$	277	49	0.960705

utility configuration. Table 17 summarizes how these controlled perturbations affect positive and negative optimization outcomes as well as the relative ordering induced by the utility.

Table 17 first demonstrates the role of the source-warhead penalty. Removing this term increases the apparent number of positive outcomes from 279 to 297, but the additional outcomes include edits that fail the source-label retention diagnostic. The accompanying decrease in rank agreement ($\rho_s = 0.778$) shows that the penalty changes the learned preference ordering rather than acting as a cosmetic regularizer. In contrast, increasing λ_{WH} to 3 or 4 leaves the positive/negative outcomes and ranking essentially unchanged, indicating that $\lambda_{\mathrm{WH}} = 2$ is a stable calibration for the supported edit space. The reachable-utility analysis in Appendix E.2 explains this plateau: the four property terms jointly attain only about 1.53 in the best observed case, so larger penalty values add little information while preserving the same preference structure.

The property-term analysis in Table 17 further shows that QED and toxicity provide the principal continuous optimization signals. Removing toxicity yields the largest ranking change ($\rho_s = 0.689$), decreases positive outcomes from 279 to 247, and increases negative outcomes from 47 to 79. Removing QED similarly reduces positive outcomes to 253 and increases negative outcomes to 73. These results support the central design of WCPO: the learned policy is encouraged to improve overall molecular quality while explicitly avoiding toxicity-oriented regressions, rather than optimizing a single developability proxy.

Table 18: Offline source-relative Pareto coverage on the stored 472-source outputs. Eligible denotes RDKit-valid and Full-WH candidates. Pareto candidates counts non-dominated final outputs across sources, while Pareto-source coverage is the percentage of sources for which the method has at least one non-dominated candidate.

Method	Eligible	Pareto candidates	Pareto-source coverage (%)
GPT-5.5	382/472	298	63.14
Gemini-3.1-Pro	393/472	294	62.29
Claude-Opus-4.7	405/472	315	66.74
GPT-5.6-Sol	403/472	301	63.77
Qwen3.7-Max	238/472	211	44.70
GLM-5.2	161/472	127	26.91
DeepSeek-V4-Pro	180/472	132	27.97
Qwen3-32B	139/472	100	21.19
Qwen3-14B	126/472	89	18.86
CovMOLAgent	**429/472**	**351**	**74.36**

Meanwhile, SA and Lipinski provide stable auxiliary guidance without dominating the optimization objective. As shown in Table 17, removing SA or Lipinski preserves high rank agreement ($\rho_s = 0.989$ and 0.978, respectively), and moderate changes to their coefficients maintain nearly identical preference orderings. This behavior is desirable for multi-objective optimization: SA and Lipinski refine the preference among otherwise competitive candidates, while QED, toxicity, and warhead preservation retain the primary decision-making roles.

Finally, the local robustness results in Table 17 show that the selected utility is not dependent on a brittle coefficient setting. Moderate perturbations of the QED, Lipinski, SA, and toxicity coefficients retain high rank agreement ($\rho_s \geq 0.960$), while perturbations of the SA and toxicity normalization scales also preserve the overall preference structure. In the locally linear region, α_k and κ_k jointly control the effective slope α_k/κ_k; however, the clipping operation additionally allows κ_{SA} and κ_{Tox} to regulate saturation. Together, these observations show that WCPO achieves a calibrated balance: it strongly favors retention of the source-warhead labels, emphasizes the two most consequential continuous objectives, and incorporates SA and Lipinski as bounded stabilizing signals.

I.7 Source-Relative Pareto Analysis

To complement the fixed WCPO scalarization, we perform an offline source-relative Pareto analysis over the stored outputs of all evaluated methods. For each of the 472 sources, the candidate set contains one final molecule from each method. We retain only RDKit-valid candidates that preserve every source warhead label under the shared legacy SMARTS evaluator. The four coordinates are $d_{\mathrm{QED}} = \mathrm{QED}_{\mathrm{final}} - \mathrm{QED}_{\mathrm{seed}}$, the signed Lipinski pass transition, $d_{\mathrm{SA}} = \mathrm{SA}_{\mathrm{seed}} - \mathrm{SA}_{\mathrm{final}}$, and $d_{\mathrm{Tox}} = \mathrm{Tox}_{\mathrm{seed}} - \mathrm{Tox}_{\mathrm{final}}$, so larger values are better for every coordinate. A candidate is non-dominated if no other eligible candidate is at least as good in all four coordinates and strictly better in one.

CovMOLAgent has the largest eligible candidate pool and the highest Pareto-source coverage. This indicates that its advantage is distributed across the four objectives: for most sources, its stored final output is not dominated by the final outputs of the comparison methods after enforcing validity and source-warhead retention. The analysis is a post-hoc comparison of stored outputs, rather than a Pareto-trained policy or a new optimization run.

Scope of warhead fidelity. Full-WH and Edited-WH are evaluated using the shared label-level SMARTS patterns. They measure whether the recognized source-warhead labels remain present in the final molecule, but they do not establish atom identity, attachment-site identity, maximum-common-substructure correspondence, reaction center preservation, covalent geometry, or reaction kinetics. Consequently, these metrics should be interpreted as label-level structural fidelity rather than atom-mapped mechanism preservation.

I.8 Ethical and safety considerations

Covalent inhibitors can be therapeutically valuable but can also produce irreversible off-target modification. CovMOLAgent is intended for computational hypothesis generation, not autonomous synthesis, clinical decision making, or direct recommendation of compounds for human use. Any candidate requires expert review, orthogonal reactivity and selectivity assessment, synthesis planning, and experimental validation. Dataset and model release should respect source licenses and include safeguards against overinterpreting proxy scores or using the system to design harmful reactive compounds.