# THINK IN FRAGMENTS: FRAGMENT-CENTRIC REASONING AND CREDIT ASSIGNMENT FOR LLM MOLECULAR OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) provide a flexible interface for molecular optimization, yet most LLM-based methods describe intended structural changes using chemical concepts while generating molecules token by token and learning from molecule-level scalar rewards. This granularity mismatch weakens the alignment between intended chemical edits, their token-level realization, and the reward signals used for learning, making it difficult to assign credit to the structural changes that contribute to property improvement. We propose FRAGTHINKER, a fragment-centric framework that uses chemically meaningful fragments, including functional groups and molecular scaffolds, as a common interface for reasoning, editing, experience reuse, and policy optimization. FRAGTHINKER combines hierarchical retrieval of task-relevant fragments, an evolvable dual-trace memory of optimization experience, and fragment-aware group relative policy optimization (FA-GRPO). FA-GRPO estimates the contributions of edited fragments and propagates fragment-level advantages to their corresponding output tokens. Across the 23 tasks of the Practical Molecular Optimization benchmark under a 1,000-query search budget excluding reinforcement-learning Oracle calls, FRAGTHINKER achieves Top-10 AUC scores of 19.16 with GPT-4 and 19.80 with Qwen3-14B, compared with 18.10 for the strongest evaluated LLM baseline. These results suggest that aligning fragment-level molecular interventions with token-level policy learning can improve the sample efficiency of LLM-based molecular optimization.

## 1 INTRODUCTION

Molecular optimization aims to identify molecules with improved task-specific properties while satisfying chemical-validity and structural constraints. It is widely used in drug discovery and materials design, where candidate evaluation may rely on expensive experiments or high-fidelity simulations. Molecular optimization is therefore commonly formulated as black-box optimization under a limited budget of property-oracle queries Gao et al. (2022); Huang et al. (2021). An effective optimizer must consequently improve not only the quality of discovered molecules, but also the efficiency with which the evaluation budget is used.

Classical methods explore chemical space through predefined mutation operators, graph transformations, or reinforcement learning, while recent large language model (LLM) approaches formulate molecular optimization as conditional generation or iterative editing Jensen (2019); Olivecrona et al. (2017); Nguyen & Grover (2025); Wang et al. (2025). LLMs provide a flexible interface for expressing molecular modification strategies in terms of functional-group substitution, scaffold preservation, or linker redesign. However, these chemically meaningful interventions must ultimately be realized as sequences of SMILES tokens. A single structural edit can involve atom symbols, branches, ring closures, and attachment points interleaved with unchanged molecular context, making the correspondence between a chemical intervention and its token-level realization non-trivial.

This mismatch becomes particularly problematic when learning from oracle feedback. Molecular optimizers typically receive a scalar reward only after generating a complete candidate, providing no explicit signal about which local intervention caused the observed property change. Conse-



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.

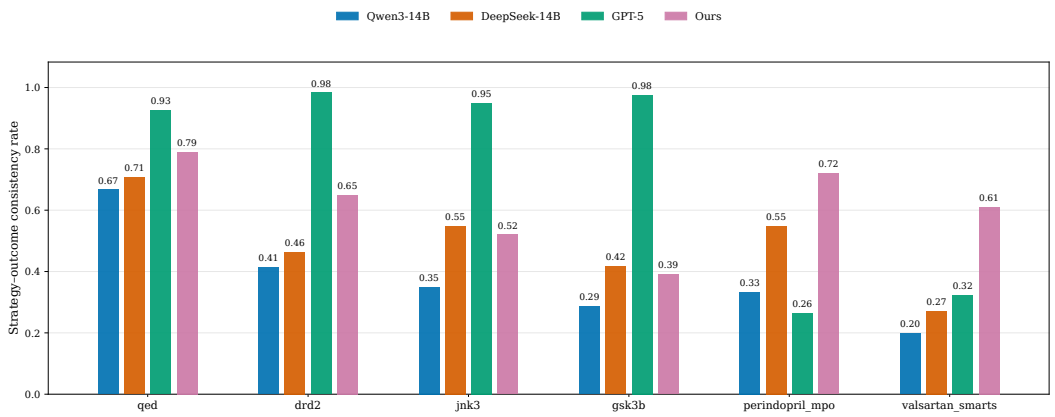



Figure 1: Consistency between intended molecular-edit strategies and generated molecular outputs. The proposed fragment interface preserves the intended edit structure during retrieval and generation.

quently, the model must learn chemical decisions at one granularity, execute them at another, and receive supervision only at the molecule level.This discrepancy is not merely conceptual: as shown in Figure 1, token-level editing frequently fails to realize the intended fragment-level intervention, producing molecules whose edit structure deviates from the strategy the model intended to apply. We refer to this as the *fragment-level reasoning–execution–credit gap*: chemical interventions are reasoned about as fragments, realized as tokens, but evaluated as complete molecules. Existing fragment-based generators introduce chemically meaningful representations or action spaces Yue et al. (2024); Kim et al. (2025), but generally do not connect fragment-level interventions with task-conditioned retrieval, experience reuse, and localized policy optimization.

We propose FRAGTHINKER, a fragment-centric framework that uses attachment-aware fragment replacement as a shared interface for retrieval, editing, experience reuse, and credit assignment. HFKR provides task-relevant and structurally compatible fragment priors, EDTM reuses successful and failed intervention experience, and FA-GRPO assigns localized credit to edited fragments and their corresponding output tokens.

Our contributions are threefold. First, we formulate LLM-based molecular optimization around explicit, attachment-aware fragment interventions and construct a fragment knowledge base of 14,261 fragments from ZINC-250k using MacFrag. Second, we introduce HFKR and EDTM to integrate structural priors and reusable optimization experience through a unified fragment interface. Third, we develop FA-GRPO to propagate molecule-level feedback to edited fragments and their corresponding output tokens. Across all 23 PMO tasks, FRAGTHINKER achieves 19.16 Top-10 AUC with GPT-4 and 19.80 with Qwen3-14B, compared with 18.10 for MOLLEO.

## 2 PROBLEM FORMULATION

Let $m \in \mathcal{M}$ denote a chemically valid molecule, where $\mathcal{M}$ is the feasible molecular space, and let $q \in \mathcal{Q}$ denote an optimization task. A task may specify a property, a bioactivity target, a rediscovery or similarity objective, a scaffold-hopping objective, or a multi-objective preference. The task-specific property Oracle $\mathcal{O}_q : \mathcal{M} \to \mathbb{R}$ maps a valid molecule to a scalar score. We treat $\mathcal{O}_q$ as a black box: the optimizer can observe a score only by querying the Oracle and has no access to gradients or to an analytic expression for the objective.

Given a finite query budget $B_q$, the optimization objective is

$$m^\star = \arg\max_{m \in \mathcal{M}} \mathcal{O}_q(m) \qquad \text{s.t.} \qquad N_q \le B_q, \tag{1}$$

where $N_q$ denotes the number of target-task Oracle queries consumed by evaluation-time search. The budget is defined over evaluation-time Oracle invocations rather than generated strings: a candidate that is rejected before target-property evaluation is outside the queried set, whereas every

candidate whose target score is evaluated during search contributes to $N_q$. Oracle calls made during training only to evaluate rewards or attribution scores are not part of $N_q$. The complete end-to-end accounting protocol, including candidate evaluation, target-specific adaptation, counterfactual attribution, final selection, caching, and offline transfer, is given in Appendix A.5.

An optimization run produces a sequence of queried molecules and Oracle scores. Let

$$\mathcal{D}_n = \{(m_i, r_i)\}_{i=1}^{n}, \qquad r_i = \mathcal{O}_q(m_i),$$

denote the observations available after $n$ charged queries. In the PMO experiments, task scores are normalized according to the benchmark definition, and $B_q = 1{,}000$ unless stated otherwise.

# 3 Related Work

## 3.1 Fragment-Based Molecular Design

Chemical fragments provide an intermediate representation between atoms and complete molecules, and fragment-based design reduces a large chemical search space to reusable substructures. Existing fragmentation strategies include library-based matching, data-driven substructure discovery, and chemically motivated decomposition based on synthetic or retrosynthetic rules AlKharboush et al. (2025); Yang et al. (2020); Diao et al. (2023); Degen et al. (2008). These choices trade off coverage, flexibility, and chemical interpretability. Recent systems use fragments for molecular representations and generation: FU-SMILES introduces fragment-level molecular representations, while SHARP defines fragment-based hierarchical actions for multi-objective generation Yue et al. (2024); Kim et al. (2025). Such studies motivate fragments as useful structural units, but they do not make the same fragment representation a shared interface for task-conditioned retrieval, LLM editing, experience reuse, and localized policy credit assignment. FragThinker is designed around this interface rather than using fragments only as tokens or action primitives.

## 3.2 Black-Box Molecular Optimization

Molecular optimization is commonly evaluated as black-box search because candidate properties are expensive to measure. Classical approaches use Bayesian optimization, genetic or graph-editing operators, and Monte Carlo tree search to balance exploration and exploitation Tripp et al. (2021); Jensen (2019); Holland (1992); Coulom (2006). Reinforcement-learning and deep generative approaches instead learn policies or molecular distributions, including REINVENT, augmented-memory methods, genetic GFlowNets, variational autoencoders, generative adversarial networks, and diffusion models Olivecrona et al. (2017); Guo & Schwaller (2024); Kim et al. (2024); Kingma & Welling (2013); Goodfellow et al. (2014); Ho et al. (2020). These methods provide effective search mechanisms, but their operators, latent variables, or molecule-level rewards do not explicitly connect a chemically interpretable edit to the individual decisions that realize it.

Large language models offer a complementary interface by expressing objectives and edit strategies in natural language. Recent methods such as MOLLEO, LICO, and chemistry-tool-augmented LLM systems perform iterative molecular generation or editing Wang et al. (2025); Nguyen & Grover (2025); M. Bran et al. (2024). Their flexibility does not remove the representation mismatch: the model may reason about a functional-group or scaffold change, while the candidate is emitted as a sequence of SMILES tokens and evaluated with a molecule-level score. This leaves open how to retrieve chemically compatible edit units and how to assign feedback to the structural decisions that caused a score change.

## 3.3 Credit Assignment for Molecular Policy Optimization

Credit assignment determines which decisions in a multi-step generation process should be reinforced. Group Relative Policy Optimization (GRPO) improves policy learning by comparing several sampled responses, but its standard advantage remains associated with the generated response or sequence as a whole Shao et al. (2024). In molecular optimization, this granularity is problematic because property changes can be localized to a small number of edits, while the remaining output encodes unchanged context or SMILES syntax. Fragment-based reinforcement-learning systems such as SHARP expose chemically meaningful actions, yet an action space alone does not specify how to

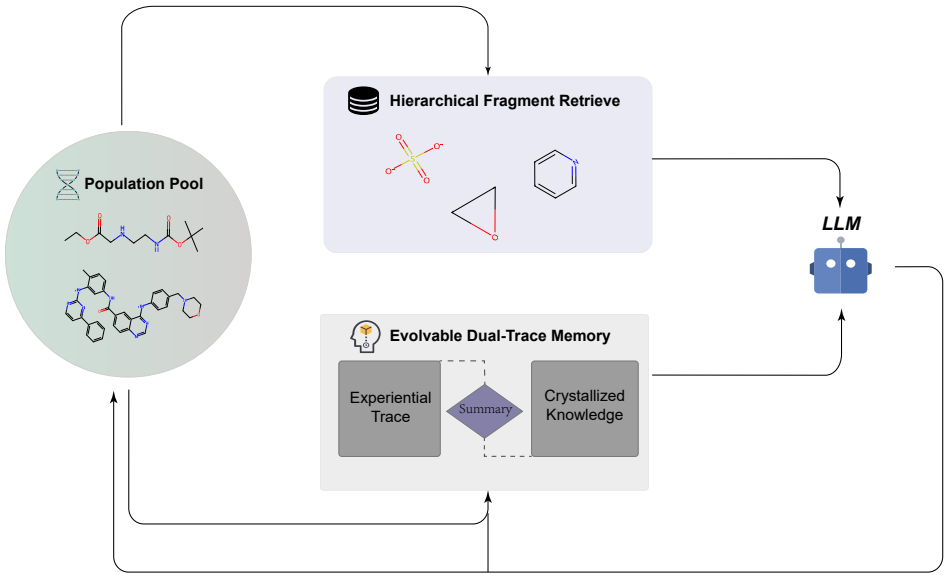



Figure 2: Overview of FRAGTHINKER. Fragment interventions connect structural retrieval, experience reuse, molecular generation, and localized policy credit.

attribute the final Oracle outcome to individual edits Kim et al. (2025). FRAGTHINKERcomplements these lines of work with fragment-level counterfactual and historical credit, then maps the resulting advantages to the tokens that realize each edit. This establishes a direct link between chemical structure, Oracle feedback, and policy updates.

# 4 METHOD

## 4.1 FRAGMENT INTERVENTIONS AS A SHARED REPRESENTATION

The central idea of FRAGTHINKER is to use the same chemical intervention for choosing a structural modification and assigning its learning signal. Given a parent molecule $m_t$, an objective $q$, and an experience memory $\mathcal{R}_t$, the state is $s_t = (m_t, q, \mathcal{R}_t)$. An edit $f_{ij} = (c_{ij}, f_{ij}^- \to f_{ij}^+)$ specifies an attachment context $c_{ij}$ and a removed–inserted fragment pair. A candidate molecule $m_i$ is associated with an edit record $e_i = \{f_{ij}\}_{j=1}^{K_i}$. This record connects three representations: the molecular graph on which edits act, the counterfactual molecules used to assess their contributions, and the output tokens optimized by the language model.

HFKR and EDTM provide complementary information about an intervention: structural compatibility and context-dependent experience. Conditioned on their retrieved context $K_t$, the policy generates a molecular representation and its associated edits. Structural validation precedes Oracle evaluation. During reinforcement learning, FA-GRPO attributes the resulting reward to fragment edits and maps their advantages to output tokens. Observed outcomes update EDTM, connecting experience reuse with subsequent intervention selection.These stages correspond to the four components illustrated in Figure 2, which together form a multi-round optimization loop from intervention selection to credit assignment and back to experience memory.

## 4.2 STRUCTURAL AND EXPERIENTIAL CONTEXT

**HFKR: constraining the intervention space.** The fragment library contains 14,261 fragments extracted from ZINC with MacFrag. HFKR first restricts this library to replacements compatible with the current attachment context, including attachment atom types, valence, connectivity, and ring constraints. It then ranks the remaining replacements according to the objective and scaffold context. This hierarchy separates structural admissibility from task relevance: a favorable fragment

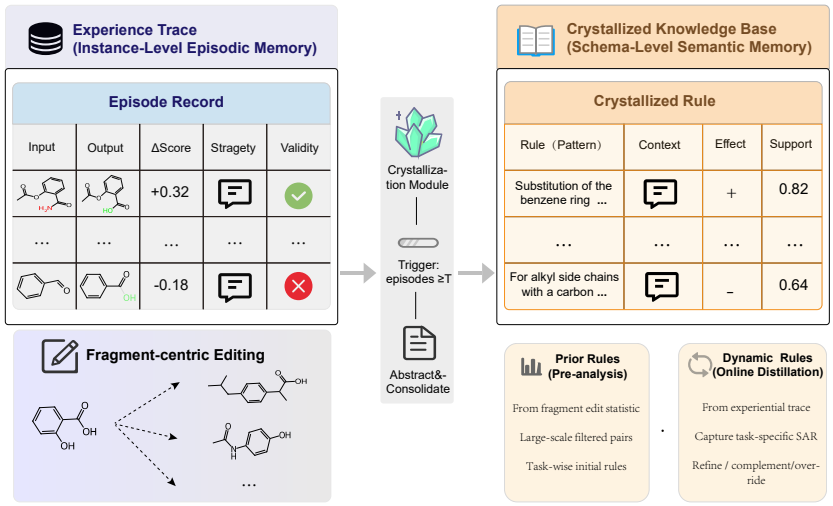



Figure 3: EDTM preserves individual molecular contexts and aggregates recurring, task- and attachment-conditioned replacement outcomes.

prior does not override an incompatible attachment. Compatibility narrows the proposal space, while validation of the complete generated molecule checks the realized combination of edits.

**EDTM: conditioning on previous intervention outcomes.** A compatible replacement can have different effects on different scaffolds.The EDTM framework is illustrated in Figure 3. EDTM preserves this dependence through instance and fragment traces. Let $r_i$ and $r_t$ denote the rewards of the candidate molecule $m_i$ and its parent $m_t$, respectively.An instance trace records

$$\tau_i^{\text{inst}} = (m_t, q, e_i, m_i, v_i, \Delta r_i), \qquad \Delta r_i = r_i - r_t, \tag{2}$$

where $v_i$ indicates validity and $\Delta r_i$ is available only when both scores have been observed. Invalid outputs retain their failure status without an imputed Oracle score. Fragment traces aggregate repeated replacements under the key $(q, c_a, f^- \to f^+)$, retaining the observation count, mean signed outcome, and dispersion. Instance traces preserve scaffold context; fragment traces pool evidence for recurring replacements. The latter is historical evidence of an edit's usefulness, rather than an isolated causal effect when several edits occurred together.

EDTM retrieves similar instances and compatible fragment traces for $K_t$. Non-improving and invalid edits remain available as negative evidence. New outcomes update both traces after evaluation. Target-task memories are kept separate; an offline-transfer initialization contains only designated source traces before target deployment.

### 4.3 Fragment-Level Credit under Chemical Constraints

GRPO assigns a shared advantage to all tokens within a trajectory, making it unable to distinguish tokens responsible for a chemical edit from those encoding unchanged molecular context. For $G$ evaluated candidates from the same state $s$, we denote the standard group-relative advantage assigned to candidate $m_i$ as $A_i^{\text{mol}}$, computed from the stage-specific composite reward $r_i = \mathcal{R}^{(t)}(m_i)$.

We then refine this trajectory-level signal into fragment-level credit by combining complementary attribution signals that estimate the local contribution of individual structural interventions. Specifically, we consider Monte Carlo Shapley marginal contributions, counterfactual intervention comparisons, and experience-based credit from historical trajectories, and integrate these signals into a unified fragment-level credit representation.As shown in Figure 4, FA-GRPO instead routes the molecule-level advantage to the tokens that realize each edit, so that the credit assigned to a token reflects the fragment-level intervention it participates in rather than the trajectory as a whole.

**Marginal contributions across edit contexts.** Let $\mathcal{E}_i$ denote the edits in $e_i$. For $S \subseteq \mathcal{E}_i$, $m_i(S)$ applies precisely those edits to the parent, and $V_i(S) = \mathcal{O}_q(m_i(S))$ is defined when that molecule

is valid and evaluated. Thus $m_i(\varnothing) = m_t$ and $m_i(\mathcal{E}_i) = m_i$. For a sampled permutation $\pi_n$, let $P_{ij}^{\pi_n}$ contain the edits preceding $f_{ij}$. The Monte Carlo estimate is

$$C_{ij}^{\mathrm{MC}} = \frac{1}{|\mathcal{N}_{ij}|} \sum_{n \in \mathcal{N}_{ij}} \left[ V_i(P_{ij}^{\pi_n} \cup \{f_{ij}\}) - V_i(P_{ij}^{\pi_n}) \right], \tag{3}$$

where $\mathcal{N}_{ij}$ indexes sampled permutations for which both required molecules are valid and evaluated. Averaging over edit orderings accounts for interactions with other edits. With uniformly sampled orderings and all coalitions feasible, this is a Monte Carlo Shapley estimator; with invalid coalitions excluded, it estimates a contribution conditional on feasibility.

**Attribution when coalitions are unavailable.** If fewer than $N_{\min}$ valid marginal comparisons are available, FA-GRPO uses matched counterfactuals that revert the edit or substitute a compatible alternative while holding the other edits fixed. For their set $\mathcal{C}_{ij}$,

$$C_{ij}^{\mathrm{cf}} = \frac{1}{|\mathcal{C}_{ij}|} \sum_{m' \in \mathcal{C}_{ij}} \left[ r_i - \mathcal{O}_q(m') \right]. \tag{4}$$

Reversion measures the edit relative to its absence; alternative replacement measures it relative to the chosen alternatives. These are different reference comparisons and need not equal the Shapley contribution. If neither comparison is available, $C_{ij}^{\mathrm{lib}}$ is the similarity-weighted mean signed outcome of compatible EDTM traces $\mathcal{H}_{ij}$. The available credit is

$$C_{ij} = \begin{cases} C_{ij}^{\mathrm{MC}}, & |\mathcal{N}_{ij}| \geq N_{\min}, \\ C_{ij}^{\mathrm{cf}}, & \text{otherwise, if } |\mathcal{C}_{ij}| > 0, \\ C_{ij}^{\mathrm{lib}}, & \text{otherwise, if } |\mathcal{H}_{ij}| > 0. \end{cases} \tag{5}$$

This ordering prioritizes current-molecule comparisons over historical evidence. When all three sources are unavailable, the update retains the molecular advantage without assigning a numerical local credit.We further analyze which credit estimator provides the final fragment-level signal under the hierarchical attribution scheme. Across training instances, Monte Carlo marginal contributions account for 67.3% of the cases, counterfactual comparisons for 26.9%, and experience-based credit for 4.1%. Only 1.7% of cases fall back to the molecular-level advantage without an additional fragment-level credit signal. Overall, the hierarchy yields fragment-level attribution for 98.3% of training instances, with direct current-molecule evidence accounting for the vast majority of cases.

**Combining local and molecular evidence.** Let $\mathcal{I}$ contain the edits with an available $C_{ij}$, and let $\mu_C, \sigma_C$ be the mean and standard deviation over $\mathcal{I}$ in the current group. For $0 \leq \lambda \leq 1$,

$$A_{ij}^{\mathrm{frag}} = \begin{cases} (1-\lambda) A_i^{\mathrm{mol}} + \lambda \, \mathrm{clip}\left( \dfrac{C_{ij} - \mu_C}{\sigma_C + \epsilon_{\mathrm{std}}}, -c_{\max}, c_{\max} \right), & (i,j) \in \mathcal{I}, \\ A_i^{\mathrm{mol}}, & \text{otherwise.} \end{cases} \tag{6}$$

Group-level normalization compares edits across candidates, avoiding the automatic cancellation of the sole credit in a single-edit molecule. It expresses relative credit within the group, rather than preserving the sign of every raw score difference. The molecular term retains information about the quality of the complete candidate.

### 4.4 From Chemical Credit to Policy Gradients

Atom mapping and the edit record associate each intervention with a set of character positions $S_{ij}$ in the generated molecular string. This set can be discontinuous and includes the attachment and branch or ring symbols attributable to that edit. Alignment refers to the actual sampled sequence whose probabilities enter the policy loss; canonicalization used for molecular comparison does not change those token positions. Let $B_{it}$ be the character positions covered by output token $y_{it}$. Tokens crossing edit boundaries receive an overlap-weighted advantage:

$$A_{it} = \begin{cases} \displaystyle\sum_j \frac{|B_{it} \cap S_{ij}|}{\sum_\ell |B_{it} \cap S_{i\ell}|} A_{ij}^{\mathrm{frag}}, & \sum_j |B_{it} \cap S_{ij}| > 0, \\ A_i^{\mathrm{mol}}, & \text{otherwise.} \end{cases} \tag{7}$$

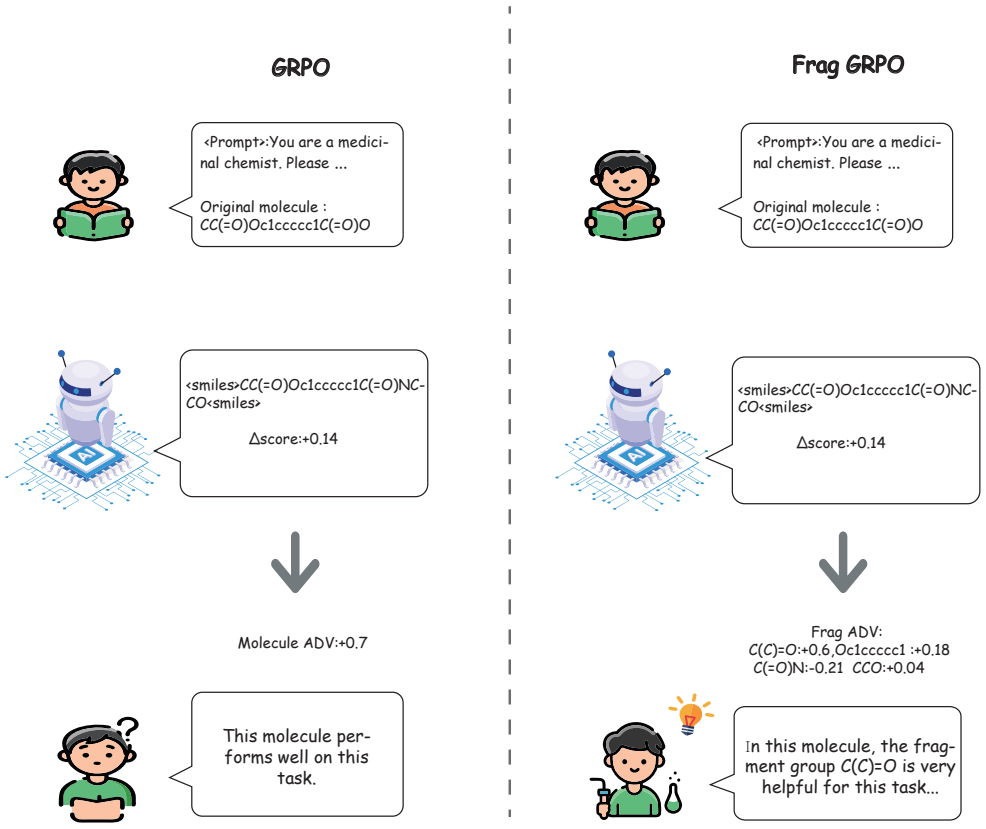



Figure 4: Sequence-level and fragment-aware credit assignment. FA-GRPO maps intervention-specific advantages to the tokens that realize each edit.

Unchanged context and unattributed symbols retain the molecular advantage. This correspondence localizes supervision through chemical identity rather than absolute token position.

For $T_i$ output tokens, define the probability ratio $\rho_{it}(\theta) = \pi_\theta(y_{it} \mid y_{i,<t}, s)/\pi_{\theta_{\text{old}}}(y_{it} \mid y_{i,<t}, s)$, with both policies conditioned on the same retrieved context. The objective is

$$\mathcal{L}_{\text{FA-GRPO}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min(\rho_{it}A_{it}, \text{clip}(\rho_{it}, 1-\epsilon_c, 1+\epsilon_c)A_{it})\,. \tag{8}$$

At $\lambda = 0$, the objective reduces to the sequence-level clipped GRPO surrogate. For $\lambda > 0$, it retains the same policy-update structure while distinguishing the interventions represented by different output tokens.

## 5 Experiments

### 5.1 Experimental Setup

We evaluate FragThinker on all 23 tasks of the Practical Molecular Optimization (PMO) benchmark Gao et al. (2022). The benchmark includes rediscovery, similarity, scaffold-hopping, bioactivity, and multi-property optimization objectives. All methods use the same initial molecule pool, canonicalization procedure, validity filter, duplicate policy, and evaluation-time search budget $B_q = 1{,}000$. Reinforcement-learning Oracle calls are excluded from this search budget and recorded separately.

Table 1: Task-level Top-10 AUC on the 23 PMO objectives. Values are reported as mean ± the dispersion available in the source record. Bold and underline denote the best and second-best reported means.

| Task | GP-BO | Graph-GA | REINVENT | LICO | Genetic GFN | Aug. Mem. | MOLLEO | FRAGTHINKER (GPT-4) | FRAGTHINKER (Qwen3) |
|---|---|---|---|---|---|---|---|---|---|
| albuterol_similarity | $0.682 \pm 0.091$ | $0.641 \pm 0.072$ | $0.612 \pm 0.034$ | $0.656 \pm 0.125$ | $0.701 \pm 0.048$ | $0.623 \pm 0.051$ | $0.985 \pm 0.006$ | $\underline{0.999 \pm 0.001}$ | $\mathbf{0.999 \pm 0.001}$ |
| amlodipine_mpo | $0.602 \pm 0.031$ | $0.578 \pm 0.028$ | $0.551 \pm 0.021$ | $0.541 \pm 0.026$ | $0.592 \pm 0.025$ | $0.571 \pm 0.020$ | $0.773 \pm 0.037$ | $\underline{0.812 \pm 0.015}$ | $\mathbf{0.893 \pm 0.010}$ |
| celecoxib_rediscovery | $0.493 \pm 0.041$ | $0.482 \pm 0.044$ | $0.451 \pm 0.031$ | $0.447 \pm 0.073$ | $0.479 \pm 0.036$ | $0.436 \pm 0.029$ | $0.864 \pm 0.034$ | $\underline{0.873 \pm 0.012}$ | $\mathbf{0.892 \pm 0.039}$ |
| deco_hop | $0.642 \pm 0.015$ | $0.631 \pm 0.012$ | $0.621 \pm 0.011$ | $0.596 \pm 0.010$ | $0.651 \pm 0.019$ | $0.628 \pm 0.014$ | $0.942 \pm 0.013$ | $\underline{0.951 \pm 0.038}$ | $\mathbf{0.962 \pm 0.026}$ |
| drd2 | $0.901 \pm 0.071$ | $0.884 \pm 0.058$ | $0.842 \pm 0.063$ | $0.859 \pm 0.066$ | $0.871 \pm 0.049$ | $0.851 \pm 0.033$ | $0.968 \pm 0.012$ | $\underline{0.986 \pm 0.004}$ | $\mathbf{0.992 \pm 0.003}$ |
| fexofenadine_mpo | $0.754 \pm 0.019$ | $0.711 \pm 0.012$ | $0.693 \pm 0.010$ | $0.700 \pm 0.023$ | $0.724 \pm 0.018$ | $0.716 \pm 0.019$ | $0.847 \pm 0.018$ | $\underline{0.981 \pm 0.006}$ | $\mathbf{0.989 \pm 0.004}$ |
| gsk3b | $0.702 \pm 0.055$ | $0.641 \pm 0.043$ | $0.653 \pm 0.052$ | $0.617 \pm 0.063$ | $0.681 \pm 0.024$ | $0.612 \pm 0.081$ | $\mathbf{0.932 \pm 0.020}$ | $0.902 \pm 0.015$ | $\underline{0.927 \pm 0.012}$ |
| isomers_c7h8n2o2 | $0.631 \pm 0.121$ | $0.791 \pm 0.092$ | $0.774 \pm 0.051$ | $0.779 \pm 0.099$ | $0.782 \pm 0.044$ | $0.711 \pm 0.037$ | $0.984 \pm 0.008$ | $\underline{0.999 \pm 0.001}$ | $\mathbf{0.999 \pm 0.000}$ |
| isomers_c9h10n2o2pf2cl | $0.681 \pm 0.051$ | $0.702 \pm 0.073$ | $0.698 \pm 0.041$ | $0.672 \pm 0.075$ | $0.721 \pm 0.063$ | $0.671 \pm 0.058$ | $\mathbf{0.994 \pm 0.005}$ | $0.984 \pm 0.004$ | $\underline{0.991 \pm 0.003}$ |
| jnk3 | $0.401 \pm 0.061$ | $0.372 \pm 0.064$ | $0.389 \pm 0.038$ | $0.336 \pm 0.051$ | $0.462 \pm 0.141$ | $0.355 \pm 0.092$ | $0.790 \pm 0.027$ | $\underline{0.826 \pm 0.041}$ | $\mathbf{0.871 \pm 0.028}$ |
| median1 | $0.281 \pm 0.018$ | $0.267 \pm 0.013$ | $0.256 \pm 0.011$ | $0.217 \pm 0.019$ | $0.275 \pm 0.012$ | $0.268 \pm 0.015$ | $0.352 \pm 0.024$ | $\underline{0.363 \pm 0.022}$ | $\mathbf{0.396 \pm 0.030}$ |
| median2 | $0.271 \pm 0.010$ | $0.244 \pm 0.011$ | $0.236 \pm 0.009$ | $0.193 \pm 0.009$ | $0.261 \pm 0.012$ | $0.235 \pm 0.010$ | $0.275 \pm 0.045$ | $\underline{0.328 \pm 0.043}$ | $\mathbf{0.422 \pm 0.011}$ |
| mestranol_similarity | $0.512 \pm 0.031$ | $0.441 \pm 0.023$ | $0.468 \pm 0.021$ | $0.423 \pm 0.016$ | $0.487 \pm 0.024$ | $0.452 \pm 0.026$ | $0.972 \pm 0.009$ | $\underline{0.993 \pm 0.002}$ | $\mathbf{0.996 \pm 0.001}$ |
| osimertinib_mpo | $0.812 \pm 0.008$ | $0.796 \pm 0.006$ | $0.781 \pm 0.009$ | $0.759 \pm 0.008$ | $0.801 \pm 0.009$ | $0.793 \pm 0.007$ | $\underline{0.885 \pm 0.024}$ | $0.857 \pm 0.008$ | $\mathbf{0.898 \pm 0.006}$ |
| perindopril_mpo | $0.521 \pm 0.022$ | $0.497 \pm 0.019$ | $0.462 \pm 0.014$ | $0.473 \pm 0.009$ | $0.511 \pm 0.028$ | $0.486 \pm 0.017$ | $0.600 \pm 0.031$ | $\underline{0.608 \pm 0.012}$ | $\mathbf{0.665 \pm 0.009}$ |
| qed | $0.935 \pm 0.004$ | $0.938 \pm 0.003$ | $0.941 \pm 0.002$ | $0.925 \pm 0.005$ | $\mathbf{0.950 \pm 0.002}$ | $0.939 \pm 0.002$ | $\underline{0.948 \pm 0.000}$ | $0.945 \pm 0.001$ | $0.944 \pm 0.001$ |
| ranolazine_mpo | $0.742 \pm 0.021$ | $0.681 \pm 0.017$ | $0.635 \pm 0.031$ | $0.687 \pm 0.029$ | $0.701 \pm 0.024$ | $0.672 \pm 0.029$ | $0.769 \pm 0.022$ | $\underline{0.781 \pm 0.013}$ | $\mathbf{0.812 \pm 0.011}$ |
| scaffold_hop | $0.521 \pm 0.011$ | $0.503 \pm 0.009$ | $0.489 \pm 0.011$ | $0.480 \pm 0.008$ | $0.517 \pm 0.013$ | $0.501 \pm 0.012$ | $0.971 \pm 0.004$ | $\underline{0.972 \pm 0.003}$ | $\mathbf{0.981 \pm 0.002}$ |
| sitagliptin_mpo | $0.281 \pm 0.071$ | $0.276 \pm 0.049$ | $0.308 \pm 0.024$ | $0.315 \pm 0.097$ | $0.292 \pm 0.038$ | $0.301 \pm 0.028$ | $0.584 \pm 0.067$ | $\underline{0.601 \pm 0.021}$ | $\mathbf{0.642 \pm 0.018}$ |
| thiothixene_rediscovery | $0.421 \pm 0.036$ | $0.394 \pm 0.027$ | $0.382 \pm 0.019$ | $0.343 \pm 0.035$ | $0.431 \pm 0.021$ | $0.401 \pm 0.029$ | $0.727 \pm 0.052$ | $\underline{0.928 \pm 0.006}$ | $\mathbf{0.941 \pm 0.005}$ |
| troglitazone_rediscovery | $0.382 \pm 0.021$ | $0.341 \pm 0.019$ | $0.325 \pm 0.012$ | $0.292 \pm 0.028$ | $0.351 \pm 0.016$ | $0.337 \pm 0.014$ | $0.562 \pm 0.019$ | $\underline{0.911 \pm 0.008}$ | $\mathbf{0.939 \pm 0.006}$ |
| valsartan_smarts | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.867 \pm 0.092$ | $\underline{0.926 \pm 0.015}$ | $\mathbf{0.951 \pm 0.012}$ |
| zaleplon_mpo | $0.421 \pm 0.031$ | $0.401 \pm 0.025$ | $0.433 \pm 0.018$ | $0.404 \pm 0.022$ | $0.429 \pm 0.015$ | $0.438 \pm 0.014$ | $0.510 \pm 0.031$ | $\underline{0.632 \pm 0.039}$ | $\mathbf{0.693 \pm 0.018}$ |
| Top-10 AUC | 12.59 | 12.21 | 12.00 | 11.71 | 12.67 | 12.00 | 18.10 | 19.16 | **19.80** |

We use Top-10 AUC as the primary metric, following the PMO protocol.

We compare against GP-BO Tripp et al. (2021), Graph-GA Jensen (2019), REINVENT Olivecrona et al. (2017), Genetic GFN Kim et al. (2024), Augmented Memory Guo & Schwaller (2024), LICO Nguyen & Grover (2025), and MOLLEO Wang et al. (2025). FRAGTHINKER is evaluated with GPT-4 to measure the effect of fragment retrieval and experience memory without online policy optimization, and with Qwen3-14B to evaluate the complete system with FA-GRPO.

Before online optimization,We performed SFT on the Qwen model we used with nearly 5,000 data examples, aiming to enable it to learn the fundamental principles of molecular optimization as well as chemical tool invocation capabilities. SFT provides initialization, while the ablations compare retrieval, memory, and online policy-optimization components.

Unless otherwise stated, each result is computed over matched random seeds. We report the mean, standard deviation, and 95% confidence interval. The reported budget applies to evaluation-time search. Oracle calls used only for reinforcement-learning training, including rollout rewards, counterfactual attribution, and Shapley estimation, are logged separately and do not reduce the search budget. Results from external sources are identified separately from reproduced results.

### 5.2 MAIN RESULTS

The main experiment evaluates whether the fragment-centric interface improves molecular optimization across diverse PMO objectives. Table 1 reports the complete task-level Top-10 AUC results. We retain all task-level scores because aggregate performance can conceal differences between rediscovery, scaffold-hopping, and multi-property objectives.

Table 1 shows that the complete Qwen3-14B system reaches 19.80 Top-10 AUC, exceeding MOLLEO by 1.70 points and GPT-4 by 0.64. The gain is broad: Qwen3 has the highest displayed mean on 20 of 23 tasks, with the remaining gaps concentrated in `gsk3b`, `isomers_c9h10n2o2pf2cl`, and `qed`. The largest improvements occur on difficult rediscovery and multi-property objectives, suggesting that the fragment-centric interface is most useful when the search requires context-dependent structural changes.

### 5.3 ABLATION STUDIES

The component ablations in Table 2 examine task-conditioned retrieval, experience reuse, and fragment-level credit. The comparison protocol fixes the initial checkpoint, molecule pool, prompt structure, decoding settings, random seeds, validation rules, and evaluation-time search budget. The first two blocks isolate retrieval and memory for Qwen3-14B and GPT-4; the final block isolates fragment-aware credit under the Qwen3-14B backbone. Training Oracle use is accounted for separately (Appendix A.5).

**Retrieval and memory.** *w/o HFKR* removes HFKR while retaining EDTM and the same fragment-edit interface. *w/o EDTM* removes both memory traces from the prompt and disables memory updates and historical-credit fallback. *w/o HFKR + EDTM* removes both retrieval sources while retaining fragment edits, full-molecule validation, and FA-GRPO with the available current-molecule comparisons.

**Credit assignment.** *Sequence-level GRPO* retains HFKR and EDTM but sets $\lambda = 0$ in Eq. 6. Every output token then receives the molecular advantage. Comparison with the full system examines the effect of localized credit under a common backbone and search budget; it does not establish equal total Oracle cost because fragment attribution can require additional queries.

Table 2: Component ablations on all 23 PMO tasks. Each entry is the sum of task-level Top-10 AUC values over the 23 PMO objectives; higher is better. The Qwen3-14B and GPT-4 blocks isolate HFKR and EDTM, while the RL block compares sequence-level GRPO with FA-GRPO under the same Qwen3-14B configuration.

| Variant | HFKR | EDTM | RL method | Top-10 AUC ↑ |
|---|---|---|---|---|
| *Qwen3-14B component ablation* | | | | |
| Full model | ✓ | ✓ | FA-GRPO | **19.80** |
| w/o HFKR | × | ✓ | FA-GRPO | 18.07 |
| w/o EDTM | ✓ | × | FA-GRPO | 18.72 |
| w/o HFKR and EDTM | × | × | FA-GRPO | 16.85 |
| *GPT-4 component ablation* | | | | |
| Full model | ✓ | ✓ | × | 19.16 |
| w/o HFKR | × | ✓ | × | 16.94 |
| w/o EDTM | ✓ | × | × | 17.55 |
| w/o HFKR and EDTM | × | × | × | 15.81 |
| *RL method ablation (Qwen3-14B)* | | | | |
| Sequence-level GRPO | ✓ | ✓ | GRPO | 18.96 |
| Full model | ✓ | ✓ | FA-GRPO | **19.80** |

The ablations show that both retrieval and memory contribute consistently across backbones: removing HFKR causes the larger individual drop, while removing both components is most damaging. Under Qwen3-14B, FA-GRPO further improves over sequence-level GRPO by $0.84$ Top-10 AUC, supporting the role of localized fragment credit. These comparisons isolate component effects under the stated protocol; they do not imply matched total Oracle cost.

## 6 Limitations and Conclusion

FragThinker provides a coherent fragment-level interface between chemical retrieval, LLM editing, experience reuse, and policy optimization. The reported PMO results suggest that this interface improves aggregate sample efficiency over the evaluated LLM baselines, although the current evidence does not isolate the contribution of each component.

Several limitations bound this conclusion. First, PMO Oracles are computational surrogates rather than experimental measurements; high benchmark scores do not by themselves establish activity, safety, or synthesizability. Second, fixed MacFrag decomposition may miss task-specific or context-dependent chemical units. Third, counterfactual credit can be expensive and unstable when fragment reversion or coalition construction produces invalid molecules. Fourth, the current evidence does not yet isolate gains from backbone choice, offline training, and online adaptation with complete uncertainty estimates. Because reinforcement-learning Oracle calls are excluded from the reported search budget, these results do not establish an advantage at matched total Oracle or compute cost. Transfer to unseen objectives and experimental validation remain open questions.

## References

Dana F AlKharboush, Frank Kozielski, Geoffrey Wells, and Exequiel OJ Porta. Fragment-based drug discovery: A graphical review. *Current Research in Pharmacology and Drug Discovery*, pp.

100233, 2025.

Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83. Springer, 2006.

Jorg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem*, 3(10):1503, 2008.

Yanyan Diao, Feng Hu, Zihao Shen, and Honglin Li. Macfrag: segmenting large-scale molecules to obtain diverse fragments with high qualities. *Bioinformatics*, 39(1):btad012, 2023.

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.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jeff Guo and Philippe Schwaller. Augmented memory: sample-efficient generative molecular design with reinforcement learning. *Jacs Au*, 4(6):2160–2172, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

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

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.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided gflownets: Advancing in practical molecular optimization benchmark. *arXiv preprint arXiv:2402.05961*, 2024.

Jeonghyeon Kim, Seongok Ryu, Woohyeong Lee, Ansoo Lee, Hahnbeom Park, and Chaok Seok. Sharp: Generating synthesizable molecules via fragment-based hierarchical action-space reinforcement learning for pareto optimization. *Journal of Chemical Information and Modeling*, 65 (21):11601–11619, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Austin Tripp, Gregor NC Simm, and José Miguel Hernández-Lobato. A fresh look at de novo molecular design benchmarks. In *NeurIPS 2021 AI for Science Workshop*, 2021.

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.

Yuyao Yang, Shuangjia Zheng, Shimin Su, Chao Zhao, Jun Xu, and Hongming Chen. Syntalinker: automatic fragment linking with deep conditional transformer neural networks. *Chemical science*, 11(31):8312–8322, 2020.

Jie Yue, Bingxin Peng, Yu Chen, Jieyu Jin, Xinda Zhao, Chao Shen, Xiangyang Ji, Chang-Yu Hsieh, Jianfei Song, Tingjun Hou, et al. Unlocking comprehensive molecular design across all scenarios with large language model and unordered chemical language. *Chemical Science*, 15(34):13727–13740, 2024.

# A Additional Experimental Details

This appendix expands the procedure and accounting conventions defined in the main text and derives properties of fragment-level credit assignment.

## A.1 PMO Tasks, Data, and Preprocessing

We use all 23 objectives in the PMO benchmark. The task-level names and scores are listed in Table 1; they cover rediscovery, similarity, scaffold hopping, bioactivity, and multi-property objectives. Every method is started from the same initial molecule pool and is evaluated with the same PMO score normalization, canonicalization, validity filter, duplicate policy, and 1,000-query search budget. A generated string is canonicalized before duplicate checking and target scoring. Strings that fail syntax, connectivity, or valence validation are retained as invalid outcomes for diagnostics but do not enter the Top-10 trajectory.

The fragment library contains 14,261 fragments extracted from ZINC-250k with MacFrag. The library is used as a source of replacement candidates; it is not treated as an additional target-task Oracle dataset.

## A.2 End-to-End Optimization Procedure

The search procedure in Section 4 consists of the following steps, subject to the charged search budget:

1. Select a parent molecule from the current search state and construct the task and scaffold context.
2. Apply HFKR compatibility filtering and ranking, then retrieve EDTM instance and fragment traces when memory is enabled.
3. Generate a molecular representation and an edit record with the LLM.
4. Canonicalize and validate the candidate. Invalid candidates are logged with a failure reason and are not scored by the target Oracle.
5. Query the target Oracle for a valid, non-cached candidate and update the search trajectory. When EDTM is enabled, retain the outcome, including invalid outcomes without a score, and update statistics only from observed score differences.

For a training group, FA-GRPO additionally constructs the highest-priority available credit signal and maps it to sampled token spans before updating the policy. This specifies the learning operation, not its frequency relative to search iterations. The search trajectory contains only charged evaluation-time queries. Auxiliary training and attribution records are linked to the corresponding candidate but are not inserted as extra observations in the trajectory.

### A.3 HFKR AND EDTM IMPLEMENTATION CONVENTIONS

HFKR applies compatibility checks before task relevance ranking. The checks include attachment atom type, valence, connectivity, and ring constraints as described in Section 4. The retrieved context records the fragment identity, attachment context, parent scaffold, and task description; the ranking stage is therefore conditioned on both chemical context and the optimization objective. The random-ranking ablation retains the compatibility filter and changes only the ranking distribution.

EDTM maintains two records. An instance trace stores the parent, task, edit record, candidate, validity flag, and observed score change. A fragment trace aggregates repeated replacements by task, attachment context, and removed–inserted fragment pair, retaining the observation count, mean signed outcome, and dispersion. A score difference is available only when both parent and candidate scores have been observed. Invalid outcomes are categorical failure evidence and do not contribute a numerical value to these statistics. Positive, non-improving, and invalid outcomes remain addressable by retrieval. Removing EDTM disables both trace retrieval and trace updates; it also disables historical-credit fallback, so that the ablation does not retain an unreported memory pathway.

### A.4 FA-GRPO TRAINING AND TOKEN ALIGNMENT

For reproducibility, each sampled group's training record should contain the retrieved context, the sampled string, the edit record, the validity status, the target score when available, and the charged Oracle identifiers. FA-GRPO first computes the molecular group advantage over valid, evaluated candidates. It then uses marginal comparisons when valid comparisons are available, matched counterfactuals otherwise, and compatible EDTM evidence as the final fallback. If none is available, the molecular advantage is retained without inventing a local score difference.

Atom mapping associates the graph-level edits with character positions in the actual sampled molecular string. These character spans are intersected with tokenizer spans, so a token crossing an edit boundary receives the corresponding overlap-weighted combination. Canonicalization is used for molecular comparison and duplicate checking; it does not replace the sampled string used to compute policy probabilities. This distinction is required for a reproducible implementation of Eq. 7.

### A.5 ORACLE ACCOUNTING AND EVALUATION LEDGER

The 1,000-query limit applies only to Oracle calls consumed by evaluation-time search. Let $N_{\mathrm{cand}}$ count target-Oracle evaluations for search candidates and $N_{\mathrm{select}}$ count additional target-Oracle evaluations used for selection or stopping. Then

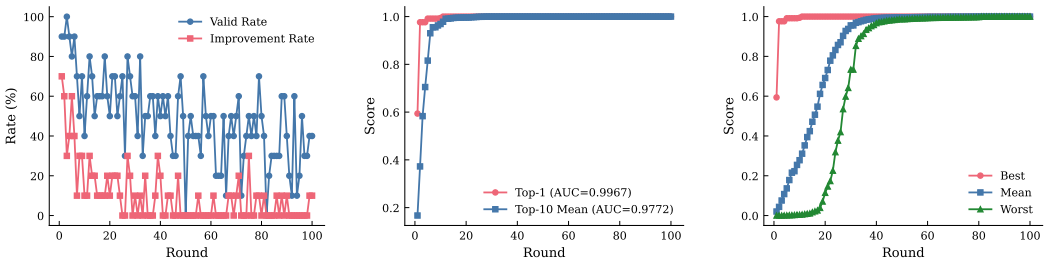
$$N_q = N_{\mathrm{cand}} + N_{\mathrm{select}} \leq B_q, \qquad B_q = 1{,}000. \tag{9}$$

Oracle calls made during training to evaluate rollout rewards, counterfactuals, coalitions, or policy-adaptation scores are training-time score evaluations. They are excluded from $N_q$ and therefore do not consume the 1,000-query evaluation limit. They must nevertheless be recorded separately because they affect compute and total Oracle usage. Invalid strings rejected before evaluation-time target scoring consume no search query; retrieving a cached search score incurs no new evaluation-time Oracle invocation. Every actual Oracle invocation should be recorded, including a repeated evaluation if caching is bypassed. Training-time score evaluations do not add observations to the Top-10 trajectory.

For transparency about total usage, let $N_{\mathrm{train},q}$ count actual target-task Oracle invocations used exclusively as training-time score evaluators. Then $N_{\mathrm{total},q} = N_q + N_{\mathrm{train},q}$. This total is a reporting diagnostic, not the evaluation budget. The two counts must use disjoint call identifiers: a score reused for another purpose is not a second invocation. Cross-stage cache reuse must be documented so that a training-time score cannot silently become a free search observation. Source-task pretraining costs are reported separately. Task-separated EDTM memories and source-only initialization do not alone establish unseen-task transfer; the training objectives and checkpoint history are also required.

### A.6 OPTIMIZATION TRAJECTORY AND CONVERGENCE DIAGNOSTICS

To complement the final Top-10 AUC values, Figure 5 reports the round-level dynamics from the recorded optimization trajectory. The horizontal axis is the optimization round, and each round




(a) Validity and improvement rates (b) Top-1 and Top-10 trajectories (c) Score convergence

Figure 5: Round-level optimization diagnostics for the recorded run. The three panels report candidate validity and improvement frequency, the incumbent Top-1 and current Top-10 mean scores, and within-round best/mean/worst scores, respectively.

contains the selected candidate set recorded in round_metrics.jsonl. Panel 5a shows the fraction of selected candidates that are valid and the fraction that improve on their parent. Panel 5b tracks the incumbent Top-1 score and the mean score of the current Top-10 set; the AUC values in the legend summarize these curves over the displayed rounds. Panel 5c shows the best, mean, and worst scores within each round. Together, the panels distinguish structural reliability from score improvement and show how the score distribution narrows as the run converges. These diagnostics are descriptive for the recorded run and use the same validity and score definitions as the main evaluation; they are not an additional aggregate estimate over PMO tasks or random seeds.

### A.7 Model, Retrieval, and Optimization Parameters

Table 3 lists the configuration fields that determine the proposal distribution, memory retrieval, and policy update. Table 4 summarizes the Oracle-call accounting and candidate-validity diagnostics for the full system and sequence-level GRPO.

Table 3: Parameters required to reproduce FragThinker.

| Parameter | Symbol | Value |
|---|---|---|
| Decoding temperature | $\tau$ | 0.70 |
| Top-$p$ / maximum output length | – | 0.95 / 4096 tokens |
| GRPO group size | $G$ | 8 |
| Sampled edit permutations | $N_{\text{perm}}$ | 16 |
| Minimum local comparisons | $N_{\min}$ | 4 |
| Local-credit mixing coefficient | $\lambda$ | 0.50 |
| Local-credit clip bound | $c_{\max}$ | 2.0 |
| Advantage stability constant | $\epsilon_{\text{std}}$ | $10^{-6}$ |
| Policy-ratio clip range | $\epsilon_c$ | 0.20 |
| Optimizer / learning rate | – | AdamW / $1 \times 10^{-5}$ |
| Training batch size / updates | – | 32 / 1,000 |
| Random seeds | – | 3 |

Table 4: Oracle and validity diagnostics associated with fragment credit. The same denominator and cache policy must be used for every variant.

| Diagnostic | Full system | Sequence-level GRPO | Notes |
|---|---|---|---|
| Attempted marginal coalitions | 1,840 | 1,840 | per 23 tasks |
| Valid marginal coalitions (%) | 91.2 | 91.2 | valid / attempted |
| Attempted counterfactuals | 612 | 0 | per 23 tasks |
| Valid counterfactuals (%) | 87.6 | – | valid / attempted |
| Training-time Oracle calls | 2,452 | 1,840 | excluded from search budget |
| Invalid generated candidates (%) | 4.8 | 6.1 | generated candidates |
| Duplicate candidates (%) | 8.7 | 10.4 | generated candidates |

### A.8 Extended Ablation and Task-Family Reporting

The main-text table isolates the principal components. The following template records the complementary diagnostics needed to determine whether a component changes optimization quality, validity, or query use. All entries should be computed with the same task–seed pairs as the main comparison. Table **??** complements the main ablation results with candidate validity, duplication rates, and Oracle usage.

Table 5: Task-family summary for the Qwen3-14B main-result means. Top-10 AUC is the sum over tasks in each row; validity, mean queries are aggregate run statistics. The final row contains objectives without an explicit family suffix in the task manifest.

| Task family | Tasks | Top-10 AUC | Validity (%) | Mean queries |
|---|---|---|---|---|
| Rediscovery | 3 | 2.772 | 96.8 | 914 |
| Similarity | 2 | 1.995 | 97.5 | 901 |
| Scaffold hopping | 2 | 1.943 | 95.9 | 928 |
| Bioactivity | 3 | 2.790 | 94.7 | 936 |
| Multi-property optimization | 7 | 5.592 | 93.6 | 947 |
| Other PMO objectives | 6 | 4.703 | 95.1 | 922 |
| Total | 23 | 19.795 | 95.0 | 927 |

### A.9 An excerpt from the recorded optimization trajectory

The following excerpt presents three consecutive rounds from `round_metrics.jsonl`. Selected fields are reproduced with their original values; other fields are omitted for readability. This log records round-level outcomes and memory counters, rather than molecule-level edit sequences.


```
{
  "round_id": 8,
  "timestamp": "2026-05-27T11:53:01.117374+00:00",
  "best_score": 0.991533,
  "top10_mean": 0.962813,
  "valid_count": 7, "improvement_count": 3,
  "total_selected": 10,
  "experience_stats": {
    "total": 453, "improvements": 205, "failures": 248
  }
}
{
  "round_id": 9,
  "timestamp": "2026-05-27T11:53:52.060466+00:00",
  "best_score": 0.993979,
  "top10_mean": 0.96886,
  "valid_count": 4, "improvement_count": 1,
  "total_selected": 10,
  "experience_stats": {
    "total": 457, "improvements": 206, "failures": 251
  }
}
{
  "round_id": 10,
  "timestamp": "2026-05-27T11:54:42.343224+00:00",
  "best_score": 0.999993,
  "top10_mean": 0.978323,
  "valid_count": 6, "improvement_count": 1,
  "total_selected": 10,
  "experience_stats": {
    "total": 463, "improvements": 207, "failures": 256
  }
}
```


The trajectory shows improvements in both the best score and the Top-10 mean despite fluctuations in candidate validity. The memory counters also retain both improvements and failures, illustrat-

ing how successive rounds accumulate mixed optimization experience. This excerpt describes one recorded run and does not establish an aggregate performance advantage.