LEARNING TO EVOLVE MOLECULES WITH LOCAL-GLOBAL CREDIT ASSIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) optimize molecules through trajectories of successive proposals and evaluator feedback, seeking property improvements under chemical validity and similarity constraints. Outcome-level reinforcement learning trains on these trajectories but assigns the same advantage to every turn, despite differences in molecular quality. We introduce *Learning to Evolve* (L2E), which trains LLMs through a *local-global principle*. For trajectories sharing an initial molecule and objectives, global advantage compares cumulative molecular scores across trajectories, while local advantage compares molecular scores within each trajectory. Adding weighted, centered local advantage distinguishes turns by molecular score while preserving global advantage as their mean turn advantage. Our analysis characterizes the local gradient contribution and gives sufficient conditions for improving the best feasible score over a global-only update. On S^2-Bench, L2E improves molar refractivity in 69.7% of cases versus 45.3% for the strongest outcome-level RL baseline under the same proposal budget, while mean relative improvement reaches 1.66× RLOO's. On MuMOInstruct, gains in joint BBBP, DRD2, and penalized LogP optimization extend to unseen instructions. L2E also benefits from additional refinement turns.

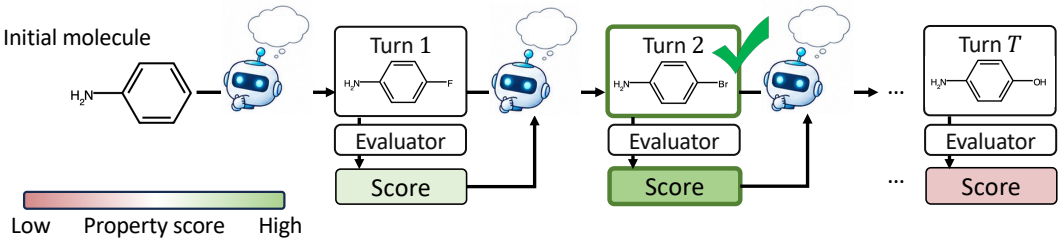

Figure 1: **Molecular optimization through evaluator feedback.** Each response contains reasoning and a proposed molecule. Evaluator feedback guides the next turn. Colors indicate property scores. We select the feasible molecule with the highest selection score, including the initial molecule.

1 INTRODUCTION

Molecular optimization seeks to improve a given molecule's target properties while maintaining structural similarity through search over chemical space (Jorgensen, 2009). A search iteratively modifies molecular structures and evaluates the resulting molecules, using feedback to guide further modifications. LLMs use chemical knowledge to improve search quality and efficiency (Wang et al., 2025a). In practice, molecular optimization helps identify molecules for drug discovery (Grebner et al., 2026), dielectric polymers (Sahu et al., 2026) and hydrogen storage (Harb et al., 2026).

Training-free methods use evaluator feedback to guide molecular optimization while keeping the model fixed (Wang et al., 2025a). Although search experience informs subsequent proposals, it does not update the model's parameters to improve future optimization. Outcome-level RL enables the model to learn from this experience (Shao et al., 2024; Ahmadian et al., 2024). These methods assign one advantage to the entire trajectory. Turns yielding different molecular scores receive the same advantage, so the advantage coefficient does not distinguish their individual molecular scores.

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.

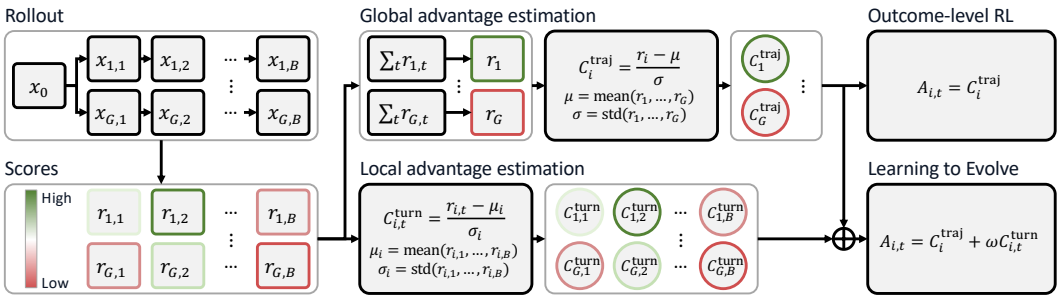

Figure 2: **Credit assignment for molecular optimization.** G trajectories of B turns start from the same instance x_0; $x_{i,t}$ denotes the instruction–molecule pair, with molecular score $r_{i,t}$. Global advantage standardizes cumulative molecular scores, and local advantage standardizes scores within each trajectory; (μ, σ) and (μ_i, σ_i) are the corresponding means and standard deviations. Outcome-level RL shares global advantage across turns. L2E adds local advantage weighted by ω, distinguishing turns while preserving global advantage as the mean turn advantage within the same trajectory.

We analyze outcome-level RL and find three patterns. First, improvements in final success and molecular quality vary across policies. Second, a policy can produce more feasible proposals yet gain few additional successes, while its later proposals deepen improvements in already successful trajectories. Third, successful trajectories contain both proposals that improve the best feasible score so far and proposals rejected by the evaluator. A shared trajectory advantage assigns these differently evaluated turns the same coefficient. These patterns motivate using individual molecular evaluations to distinguish turns while retaining an overall assessment of each complete trajectory.

A high-scoring trajectory can contain low-scoring proposals, while a proposal can outperform others in a low-scoring trajectory. Comparing complete trajectories overlooks internal differences; standardizing scores within each trajectory removes differences in mean molecular score across trajectories. These complementary limitations motivate a *local-global principle*: learning should distinguish proposals while retaining trajectory quality as a shared reference. Global comparisons establish this reference across trajectories, and local comparisons distinguish molecular quality within them.

We introduce *Learning to Evolve* (L2E), an RL algorithm that learns from molecular scores through local and global comparisons (Figure 2). Global advantage compares cumulative molecular scores across trajectories. Local advantage distinguishes higher and lower molecular scores within each trajectory, addressing the shared turn advantage of outcome-level RL. Preserving the mean turn advantage allows local differentiation while retaining the global assessment of each trajectory.

Concretely, L2E samples equal-length trajectories for each task and initial molecule. Molecular scores combine property objectives, chemical validity, and similarity penalties. Standardizing cumulative scores across trajectories yields *global advantage*; standardizing scores within each trajectory yields zero-mean *local advantage*. Adding weighted local advantage to shared global advantage gives the *turn advantage* used to train each complete response. The weight controls within-trajectory differentiation; global advantage remains the mean turn advantage for any local weight.

We evaluate L2E on single- and multi-property molecular optimization (Li et al., 2026; Dey et al., 2025). Under the same proposal budget, L2E improves molar refractivity in 69.7% of test cases, compared with 45.3% for the strongest outcome-level RL baseline. Alongside this broader success, L2E achieves larger property gains: mean relative improvements on lipophilicity and molar refractivity reach 1.94× and 1.66× those of RLOO, respectively. In multi-property optimization, L2E's gains extend to unseen instructions for selected property combinations. L2E also obtains further property improvements through successive proposal–feedback turns across the evaluated turn budgets.

Our contributions are threefold:

- We identify uneven gains from later proposals under outcome-level RL and show that successful trajectories mix improving and rejected proposals under a shared turn advantage (Section 3).
- We introduce L2E's local-global credit assignment and provide a theoretical analysis of its gradient effects and conditions for search improvement over a global-only policy update (Section 4).
- We show optimization gains and generalization to unseen instructions for selected property combinations, with further gains across the evaluated refinement horizons (Section 5).

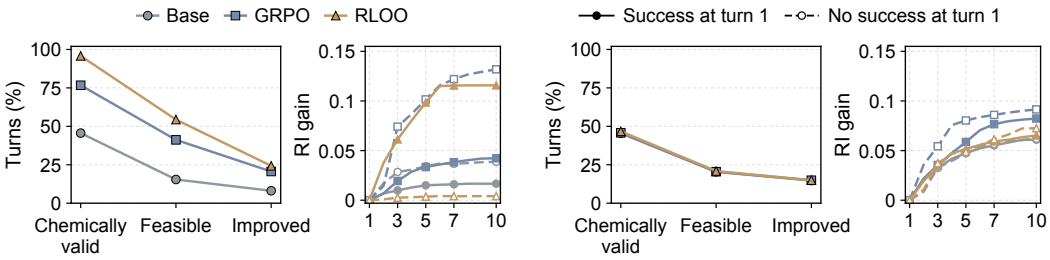

Figure 3: **Molecular quality and later gains.** Left pair: 3B; right pair: 7B. Quality panels show nested percentages of all turns: chemical validity, feasibility, and property improvement over the starting molecule. Gain panels split the increase in mean relative improvement (RI) since turn 1 between trajectories with and without success at turn 1.

2 Molecular Optimization with Evaluator Feedback

Trajectory generation. An instance x_0 pairs a chemically valid initial molecule with an instruction specifying the target properties and their desired directions. Over a B-turn trajectory, model f_θ generates responses from the history H_{t-1} of earlier molecules and evaluator feedback:

$$e_t \sim f_\theta(\cdot \mid x_0, H_{t-1}), \qquad e_t \mapsto x_t, \qquad t = 1, \ldots, B. \tag{1}$$

Each e_t contains reasoning and a proposed molecule; when parsing succeeds, x_t pairs that molecule with the unchanged instruction. The evaluator returns property values and the outcomes of chemical validity and similarity checks after each proposal, providing feedback for subsequent turns.

Property score. We aggregate n target properties as $q(x) = \sum_{j=1}^{n} \alpha_j g_j(P_j(x))$, where $P_j(x)$ is the jth property of the molecule in x. The weights $\alpha_j \geq 0$ and benchmark transforms g_j are fixed per task. Each transform assigns higher scores in the requested property direction.

A proposal is chemically valid if parsing and sanitization succeed, and feasible ($v_t = 1$) if it also passes all task-specific checks, including Morgan-fingerprint Tanimoto similarity to the initial molecule of at least 0.4 (Rogers & Hahn, 2010; Bajusz et al., 2015).

Let $S_{\text{sel}}(x; x_0)$ be the task-specific selection score and $\mathcal{F}_t = \{x_j : 1 \leq j \leq t,\ v_j = 1\}$ the feasible proposals. Including the initial molecule, the best feasible selection score after t turns is

$$b_t = \max_{x \in \{x_0\} \cup \mathcal{F}_t} S_{\text{sel}}(x; x_0). \tag{2}$$

The search returns a molecule attaining b_B. Including the initial molecule guarantees $b_B \geq b_0 = S_{\text{sel}}(x_0; x_0)$, even when no feasible proposal improves its score. All turns count toward B, including invalid, rejected, and duplicate proposals. The selection score is distinct from the molecular score r_t used for training below; the evaluation block fixes its formula and tie rule.

3 Analyzing Molecular Optimization Trajectories

We compare Base, GRPO (Shao et al., 2024), and RLOO (Ahmadian et al., 2024) using Qwen2.5-3B/7B-Instruct (Yang et al., 2024a) on LogP, MR, and QED optimization. Each evaluation trajectory contains ten proposal turns, with one proposed molecule per turn.

Observation 3.1. Chemical validity does not reliably translate into property improvement.

Notably, fewer than one third of chemically valid proposals both pass all evaluator checks and improve the target property over the given molecule (Figure 3). Thus, a substantial share of the search budget produces chemically valid molecules without yielding property gains.

Observation 3.2. Later-turn gains arise from different trajectory groups across methods.

For the 3B model trained with RLOO, trajectories successful at turn 1 contribute 96.6% of the later net increase in mean RI, while search success rate rises by only 1.0 percentage point. For 3B GRPO, later turns increase search success rate by 30.2 percentage points, with initially unsuccessful trajectories contributing most of the RI increase (Figure 3, second panel).

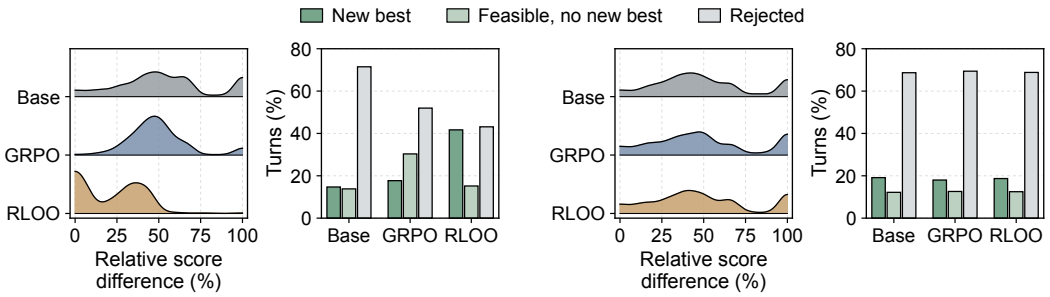

Figure 4: **Score differences and turn outcomes within successful trajectories.** Left pair: 3B; right pair: 7B. Densities show mean absolute evaluator-score differences among each trajectory's feasible molecules, as a percentage of its highest score. Bars show outcome shares across all ten turns. "Feasible" means passing all checks used in this analysis; "no new best" means the proposal does not improve the best feasible score so far.

Observation 3.3. Successful trajectories contain feasible molecules with different scores and rejected proposals.

Feasible molecules differ in evaluator score (Figure 4, first and third panels). Successful trajectories mix proposals that set a new best feasible score with rejected proposals that fail evaluator checks (second and fourth panels). These patterns motivate distinguishing turns by the molecular quality of their proposals while retaining trajectory-level comparisons of overall search outcomes.

4 Learning to Evolve

L2E assigns each turn a *turn advantage* combining global and local comparisons. Global advantage compares complete trajectories, while local advantage compares molecular scores within each trajectory. Their combination weights the generated response during training (Figure 2).

4.1 Rollout and Scores

Following the trajectory-generation protocol in Section 2, we sample G independent trajectories of B turns from the behavior policy f_{old} for each optimization instance x_0. Within each trajectory, we retain the most recent M molecule–feedback pairs as history. We use $e_{i,t}$ for the response and $x_{i,t}$ for the instruction–molecule pair at turn t of trajectory i, with the instruction fixed across turns.

We compute the molecular score $r_{i,t}$ using the following validity gate and similarity penalty. The *property score* $q(x)$ combines transformed property values using nonnegative weights, with each transformation oriented in the requested property direction. Let $c_{i,t} \in \{0, 1\}$ indicate successful parsing and chemical validity, and define $d(x; x_0) = \max\{0, \delta - \text{Sim}(x, x_0)\}$. We use

$$r_{i,t} = \begin{cases} q(x_{i,t}) - \lambda_{\text{sim}} d(x_{i,t}; x_0), & c_{i,t} = 1, \\ 0, & c_{i,t} = 0. \end{cases} \tag{3}$$

Here Sim denotes Morgan-fingerprint Tanimoto similarity (Bajusz et al., 2015), with $\delta = 0.4$ and $\lambda_{\text{sim}} = 1$ (Wang et al., 2025c). Parsing failures and chemically invalid molecules receive a molecular score of zero; chemically valid molecules below the similarity threshold incur a similarity penalty. Training uses these molecular scores, while output selection is restricted to feasible molecules.

4.2 Advantage Estimation

Assessing complete trajectories. We assess trajectory i by its cumulative molecular score $r_i = \sum_{t=1}^{B} r_{i,t}$, including all B turns. This training surrogate can distinguish trajectories with the same best feasible selection score but different molecular scores at other turns. During inference, we return the bestmolecule attaining b_B (Figure 1).

Assigning turn advantages. Following the local-global principle, we assign each turn the trajectory's global advantage, adjusted by local advantage. Let μ and σ be the mean and population

standard deviation of the G cumulative molecular scores. Let μ_i and σ_i be the mean and population standard deviation of the B molecular scores within trajectory i. We define *global advantage* C_i^{traj}, *local advantage* $C_{i,t}^{\text{turn}}$, and the resulting *turn advantage* $A_{i,t}$ as

$$C_i^{\text{traj}} = \frac{r_i - \mu}{\sigma + \epsilon_{\text{std}}}, \quad C_{i,t}^{\text{turn}} = \frac{r_{i,t} - \mu_i}{\sigma_i + \epsilon_{\text{std}}}, \quad A_{i,t} = C_i^{\text{traj}} + \omega C_{i,t}^{\text{turn}}, \quad \omega \geq 0, \tag{4}$$

where $\epsilon_{\text{std}} > 0$ stabilizes standardization. The *local weight* ω scales differences between turn advantages without changing their mean C_i^{traj}. For $\omega > 0$, higher molecular scores receive larger turn advantages within a trajectory. At $\omega = 0$, all turns receive C_i^{traj}, as in a global-only update.

4.3 Policy Optimization

The turn advantage $A_{i,t}$ is shared by all generated tokens in response $e_{i,t}$, including both reasoning and molecule tokens. We train on all GB responses, including those containing chemically invalid or repeated molecules. Molecular scores and turn advantages are held fixed during each policy update. Prompt and evaluator tokens receive no policy loss, while remaining available as input context.

We optimize the policy with a clipped token-level objective, averaging over tokens within each response and then over turns and trajectories. For token ℓ of response $e_{i,t}$, let $\rho_{i,t,\ell}(\theta)$ denote the ratio of its probability under the current policy f_θ to that under the behavior policy f_{old}, using the same context. We maximize the PPO-style clipped surrogate objective (Schulman et al., 2017) as:

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}\Bigg[\frac{1}{GB}\sum_{i=1}^{G}\sum_{t=1}^{B}\frac{1}{|e_{i,t}|}\sum_{\ell=1}^{|e_{i,t}|}\min\Big\{&\rho_{i,t,\ell}(\theta)A_{i,t},\\ &\text{clip}\big(\rho_{i,t,\ell}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\big)A_{i,t}\Big\}\Bigg] - \beta\mathcal{D}_{\text{KL}}(f_\theta \| f_{\text{ref}}).\end{aligned} \tag{5}$$

The expectation averages over optimization instances x_0 and the trajectories sampled from f_{old} for each instance. The clipping threshold is ϵ_{clip}, and β weights the KL penalty relative to the reference policy f_{ref}. Appendix A provides further details on the training algorithm and loss normalization.

4.4 Gradient Effects and Conditional Search Improvement

Local advantages can contribute a nonzero gradient through response-specific derivatives.

Proposition 4.1 (Gradient contribution of local advantage)**.** *At $\theta = \theta_{\text{old}}$, hold scores and advantages fixed and let $h_{i,t}$ denote the length-normalized response log-likelihood gradient. The unit-weight local gradient satisfies*

$$\widehat{d}_L = \frac{1}{G}\sum_{i=1}^{G}\frac{\text{Cov}_B(\mathbf{r}_i, h_i)}{\sigma_i + \epsilon_{\text{std}}}, \tag{6}$$

where Cov_B averages over the B responses within each trajectory, with population normalization.

We relate this direction to $J_\star(\theta) = \mathbb{E}_\theta[b_B]$, the expected best feasible selection score of fresh trajectories under the same budget, including the initial molecule among the feasible output candidates.

Theorem 4.2 (Search improvement from target-aligned local advantage)**.** *Consider one SGD ascent step from the same policy and sampled batch, with a common learning rate $\eta > 0$ and KL contribution. Let θ_G be the global-only endpoint and $\theta_L = \theta_G + \eta\omega\widehat{d}_L$ the L2E endpoint. Suppose $J_\star$ has a uniform finite gradient-Lipschitz bound on the endpoint segments, the local direction has finite second moment, and the relevant expectations exist, as specified in Appendix B.3. If $\mathbb{E}\langle\nabla J_\star(\theta_G), \widehat{d}_L\rangle > 0$, then, for fixed η and sufficiently small $\omega > 0$ chosen before sampling,*

$$\mathbb{E}\big[J_\star(\theta_L) - J_\star(\theta_G)\big] > 0. \tag{7}$$

Local advantage therefore improves expected final selection when its gradient contribution aligns with the search objective and its weight is sufficiently small. Proofs and the explicit gain bound appear in Appendices B.2 and B.3, for the covariance identity and conditional gain, respectively.

Table 1: **Single-property optimization.** Eight methods evaluated under a five-proposal budget. Entries show mean ± sample s.t.d. over inference/search repeats.

Method	QED		LogP		MR	
	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑
Evolutionary Algorithms						
GraphGA	**0.597**$_{\pm.015}$	0.068$_{\pm.011}$	0.747$_{\pm.067}$	0.532$_{\pm.075}$	**0.803**$_{\pm.015}$	0.160$_{\pm.032}$
REINVENT	0.000$_{\pm.000}$	0.000$_{\pm.000}$	0.000$_{\pm.000}$	0.000$_{\pm.000}$	0.000$_{\pm.000}$	0.000$_{\pm.000}$
MOLLEO	0.228$_{\pm.322}$	0.096$_{\pm.136}$	0.271$_{\pm.192}$	0.120$_{\pm.102}$	0.353$_{\pm.047}$	0.052$_{\pm.004}$
General-Purpose LLMs						
Qwen2.5-3B	0.267$_{\pm.047}$	0.035$_{\pm.005}$	0.343$_{\pm.035}$	0.358$_{\pm.192}$	0.313$_{\pm.055}$	0.036$_{\pm.014}$
Qwen3-4B	0.397$_{\pm.021}$	0.046$_{\pm.010}$	0.530$_{\pm.010}$	0.197$_{\pm.038}$	0.340$_{\pm.010}$	0.045$_{\pm.005}$
RL Fine-tuning (Qwen2.5-3B)						
GRPO	0.350$_{\pm.017}$	0.184$_{\pm.006}$	0.737$_{\pm.035}$	2.481$_{\pm.051}$	0.453$_{\pm.023}$	0.548$_{\pm.018}$
RLOO	0.407$_{\pm.006}$	0.378$_{\pm.005}$	0.993$_{\pm.006}$	9.984$_{\pm.124}$	0.410$_{\pm.000}$	2.496$_{\pm.045}$
L2E (Ours)	0.410$_{\pm.000}$	**0.386**$_{\pm.001}$	**1.000**$_{\pm.000}$	**19.357**$_{\pm.420}$	0.697$_{\pm.006}$	**4.138**$_{\pm.083}$

Table 2: **Multi-property optimization.** Seen and unseen instructions are shown side by side, with SR and RI for each objective combination under the same budget. Mean and sample s.t.d. over three inference repeats are shown on separate lines. Unsuccessful trajectories contribute zero RI.

Model	Seen Instructions						Unseen Instructions					
	BDP		BDQ		BPQ		BDP		BDQ		BPQ	
	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑
Evolutionary Algorithms												
GraphGA	0.134 ±.074	0.130 ±.079	0.137 ±.034	0.151 ±.071	**0.475** ±.010	**0.543** ±.150	0.137 ±.048	0.179 ±.110	**0.163** ±.020	0.149 ±.042	**0.409** ±.035	**0.348** ±.038
REINVENT	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000	0.000 ±.000
MOLLEO	0.052 ±.043	0.056 ±.041	0.029 ±.013	0.047 ±.055	0.020 ±.014	0.093 ±.124	0.041 ±.015	0.136 ±.037	0.034 ±.007	0.045 ±.031	0.015 ±.007	0.010 ±.012
General-Purpose LLMs												
Qwen2.5-3B	0.100 ±.057	0.171 ±.131	0.127 ±.026	0.068 ±.023	0.033 ±.011	0.010 ±.006	0.103 ±.021	0.187 ±.011	0.111 ±.032	0.085 ±.053	0.056 ±.015	0.026 ±.012
Qwen3-4B	0.007 ±.006	0.008 ±.008	0.029 ±.000	0.008 ±.001	0.000 ±.000	0.000 ±.000	0.045 ±.033	0.098 ±.131	0.039 ±.026	0.041 ±.039	0.013 ±.006	0.005 ±.003
RL Fine-tuning (Qwen2.5-3B)												
GRPO	0.196 ±.037	0.499 ±.145	0.059 ±.020	0.203 ±.262	0.076 ±.015	0.174 ±.059	0.247 ±.010	1.014 ±.656	0.052 ±.006	0.056 ±.032	0.099 ±.000	0.213 ±.036
RLOO	0.282 ±.016	1.461 ±.727	**0.160** ±.032	**0.294** ±.162	0.188 ±.017	0.141 ±.070	0.237 ±.037	0.842 ±.416	0.147 ±.026	**0.323** ±.061	0.132 ±.025	0.108 ±.021
L2E (Ours)	**0.347** ±.046	**4.111** ±2.216	0.157 ±.020	0.254 ±.168	0.139 ±.010	0.154 ±.008	**0.323** ±.016	**3.627** ±2.501	0.137 ±.020	0.158 ±.081	0.125 ±.006	0.140 ±.013

5 Experiments

We evaluate L2E on instruction-based single- and multi-property molecular optimization. We describe the experimental setup (Section 5.1) and present the quantitative results (Section 5.2). We then examine the credit components and reward definition (Section 5.3), evaluate gains from additional refinement turns (Section 5.4), assess model scaling and robustness across initial property values (Section 5.5), and report EGFR docking results (Section 5.6).

5.1 Experimental Setup

Datasets. We employ two instruction-based molecular optimization benchmarks: S^2-Bench (Li et al., 2026) for single-property optimization and MuMOInstruct (Dey et al., 2025) for multi-property optimization. Both benchmarks require molecular edits that satisfy specified constraints. MuMOInstruct combines BBBP (B), DRD2 (D), penalized LogP (P), and QED (Q) in three task cells: BDP, BDQ, and BPQ. Each objective combination appears in both the supplied seen and unseen instruction partitions.

Baselines. We compare L2E with molecular optimizers (GraphGA (Jensen, 2019), REINVENT (Olivecrona et al., 2017), and MOLLEO (Wang et al., 2025a)), instruction-tuned Qwen models (Yang et al., 2024a; 2025), and outcome-level RL (GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024)). The main RL comparisons use Qwen2.5-3B-Instruct with the same budgets.

Table 3: **Comparison with GEPA and POLO.** Single-property evaluation under the same budget on QED, LogP, and MR, measured by success rate (SR) and relative improvement (RI).

Method	QED		LogP		MR	
	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑
GEPA	0.110	0.041	0.180	0.652	0.130	0.187
POLO	0.387	0.340	0.913	11.268	0.633	3.245
L2E	**0.410**	**0.386**	**1.000**	**19.357**	**0.697**	**4.138**

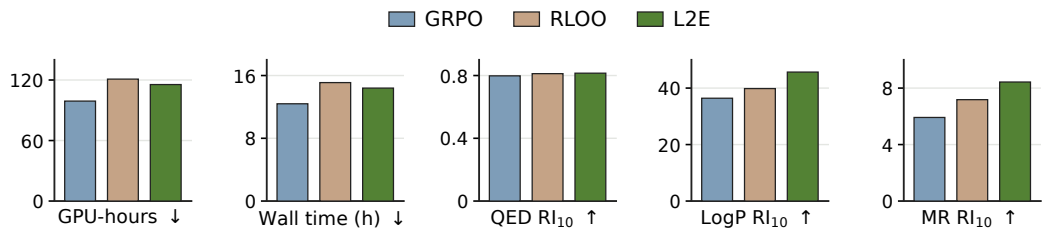

Figure 5: **Training cost and Top-10% RI.** The first two panels report GPU-hours and wall-clock training time; the remaining three show Top-10% relative improvement for QED, LogP, and MR.

Evaluation Metrics. Following prior work (Dey et al., 2025), we report two metrics. *Success Rate* (SR) is the fraction of best-scoring molecules that improve all target properties. A molecule is counted as successful only if it is chemically valid and its similarity to the initial molecule is at least 0.4. *Relative Improvement* (RI) averages directed relative gains over target properties and test cases, with failed cases contributing zero. SR and RI are higher-is-better.

5.2 Main Results

Single-property optimization. L2E achieves the highest mean RI on all three single-property tasks, with its largest gains over the RL baselines on LogP and MR (Tab. 1). Its RI reaches 19.357 on LogP and 4.138 on MR, approximately 2× and 1.7× RLOO, respectively. The training-free baselines in the table achieve at most 0.532 RI. GRPO and RLOO already improve RI over these baselines on every task, and L2E further increases the magnitude of improvement, particularly on LogP and MR.

The success rates clarify how these gains differ between tasks. On LogP, L2E and RLOO both succeed on nearly all inputs (SR of 1.000 and 0.993), while L2E nearly doubles RI. This larger improvement accompanies lower mean similarity to the initial molecule (0.768 vs. 0.812). On MR, L2E raises SR to 0.697, compared with 0.410 for RLOO and 0.453 for GRPO, combining a higher frequency of successful optimization with a larger average relative gain.

On QED, L2E and RLOO obtain similar RI (0.386 vs. 0.378) and SR (0.410 vs. 0.407), indicating similar performance on this task. GraphGA achieves the highest SR on QED and MR (0.597 and 0.803), but its RI remains 0.068 and 0.160, respectively. These results distinguish the frequency of finding a feasible property improvement from the size of the resulting gains: GraphGA succeeds more often on these two tasks, whereas L2E achieves larger average relative improvements.

Multi-property optimization. L2E achieves the highest mean RI on BDP (Tab. 2), with RI of 4.111 on seen and 3.627 on unseen instructions, significantly exceeding RLOO and GRPO, respectively. On BDQ and BPQ, L2E's relative performance varies across metrics and instruction splits. RLOO has the highest BDQ RI, while GraphGA has the highest BPQ SR and RI on both splits.

GraphGA's BPQ advantage accompanies substantial molecular simplification. Its successful candidates contain, on average, 5.58 fewer heavy atoms on seen instructions and 5.78 fewer on unseen instructions. These observations suggest that graph edits producing smaller molecules may jointly benefit this objective combination, which does not require the additional DRD2 improvement.

The relative-gain metric also amplifies improvements from low initial BBBP scores. In one successful BPQ trajectory, a candidate with 12 fewer heavy atoms retains similarity 0.517 and changes BBBP from 0.17 to 0.90, pLogP from −2.22 to 0.97, and QED from 0.46 to 0.56. The per-property relative gains are 4.29 (BBBP), 1.44 (pLogP), and 0.22 (QED), giving a mean RI of approximately 1.98. This illustrates how joint property improvement and a small reference denominator can yield high RI; BPQ retains RI of 0.543 and 0.348 after unsuccessful trajectories are assigned zero.

Table 4: **Full L2E and component ablations.** Global-only and local-only retain the named component; L2E retains both. Entries are means under the same budget.

Task	Variant	Global	Local	SR↑	RI↑	Similarity
QED	L2E	✓	✓	0.410	0.386	**0.918**
	Global-only	✓	–	**0.570**	**0.409**	0.863
	Local-only	–	✓	0.480	0.387	0.676
LogP	L2E	✓	✓	**1.000**	19.357	0.768
	Global-only	✓	–	0.757	13.724	**0.788**
	Local-only	–	✓	0.920	**157.305**	0.678
MR	L2E	✓	✓	0.697	4.138	0.740
	Global-only	✓	–	0.650	3.450	**0.812**
	Local-only	–	✓	**0.910**	**13.135**	0.502

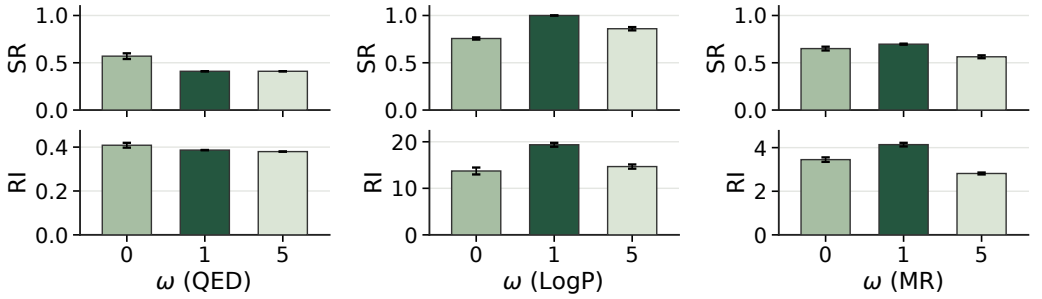

Figure 6: **Local-weight sensitivity.** Bars show means and s.t.d. over three inference repeats, using the absolute molecular score as the training reward. All bar axes start at zero.

Table 5: **Reward ablation.** Absolute molecular scores are compared with gains over the previous or best-so-far molecule as training rewards, using the same global and local advantage construction. Bold marks the best SR/RI within each task; shading indicates relative performance.

Reward	QED		LogP		MR	
	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑
L2E: absolute	**0.410**	**0.386**	**1.000**	**19.357**	**0.697**	**4.138**
Previous-state Δ	0.380	0.351	0.643	11.207	0.587	2.871
Best-so-far Δ	0.397	0.366	0.913	12.998	0.620	3.140

Comparison with GEPA and POLO. We also compare L2E with GEPA, a training-free reflective prompt optimizer (Agrawal et al., 2026), and POLO, which learns from trajectory outcomes and intermediate-molecule preferences (Wang et al., 2025c). L2E reaches higher RI than the POLO variant evaluated here: 19.357 versus 11.268 on LogP, 4.138 versus 3.245 on MR, and 0.386 versus 0.340 on QED (Tab. 3). Appendix C.2 gives the reproduction settings.

Training cost and candidate quality. Figure 5 compares GPU-hours and wall-clock training time alongside Top-10% RI. L2E requires more training compute than GRPO and slightly less than RLOO, while attaining higher Top-10% RI on the three evaluated properties.

5.3 Ablation Studies

Credit variants and local-weight sensitivity. The credit variants show a tradeoff between property gains and similarity to the initial molecule (Tab. 4). The local-only variant produces extreme RI on LogP (157.3) and MR (13.1), together with extreme molecular growth. Relative to the global-only variant, the L2E variant raises SR from 0.757 to 1.000 on LogP and from 0.650 to 0.697 on MR, while QED SR falls from 0.570 to 0.410.

The weight sweep is non-monotonic (Fig. 6). LogP and MR SR increase from $\omega = 0$ to 1, then decrease at $\omega = 5$; QED SR is highest at $\omega = 0$. At the default weight $\omega = 1$, LogP and MR attain their highest SR among the tested weights. Appendix C.3 discusses the credit-variant results.

Reward ablation. Tab. 5 compares the absolute molecular score, the difference from the previous score ($r_t - r_{t-1}$), and the difference from the best previous score ($r_t - \max_{0 \le s < t} r_s$), with $r_0 = q(x_0)$. The absolute-score variant has the highest SR and RI on all three tasks, followed by best-so-far differences and then previous-state differences. On LogP, their RI values are 19.357, 12.998, and 11.207, respectively. The absolute molecular score yields the largest gains in this comparison.

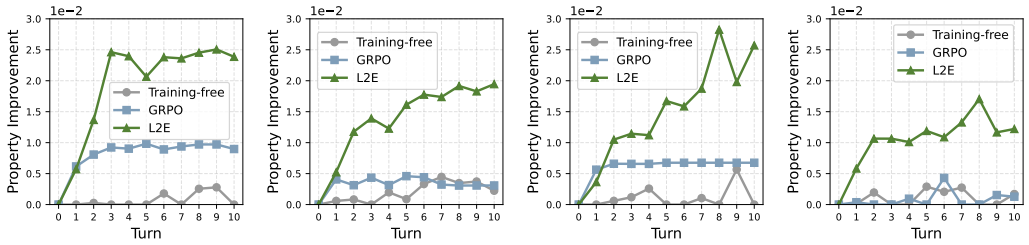

Figure 7: **Property gains from additional refinement.** Left to right: seen BDP, BDQ, BPQ, and unseen BDQ. L2E gains extend beyond the early turns.

Table 6: **Scaling to Qwen2.5-7B.** Mean and standard deviation over three inference repeats on the QED, LogP, and MR single-property molecular optimization tasks.

Method	QED		LogP		MR	
	SR↑	RI↑	SR↑	RI↑	SR↑	RI↑
Qwen2.5-7B	$0.510_{\pm.075}$	$0.086_{\pm.021}$	$0.547_{\pm.035}$	$0.412_{\pm.069}$	$0.573_{\pm.032}$	$0.113_{\pm.019}$
GRPO	$0.577_{\pm.012}$	$0.408_{\pm.006}$	$0.893_{\pm.015}$	$8.924_{\pm.419}$	$\mathbf{0.603}_{\pm.012}$	$2.431_{\pm.037}$
RLOO	$0.573_{\pm.023}$	$0.406_{\pm.012}$	$\mathbf{0.930}_{\pm.017}$	$10.058_{\pm.419}$	$0.560_{\pm.046}$	$2.603_{\pm.084}$
L2E	$\mathbf{0.613}_{\pm.045}$	$\mathbf{0.422}_{\pm.011}$	$0.843_{\pm.006}$	$\mathbf{19.899}_{\pm.747}$	$0.457_{\pm.006}$	$\mathbf{6.055}_{\pm.161}$

5.4 Sequential Refinement

Value of additional refinement turns. We examine how property improvement changes with the number of proposal–feedback turns on seen BDP, BDQ, and BPQ and unseen BDQ instructions. L2E obtains further property gains at later turns over the evaluated horizon (Fig. 7). Additional turns yield little consistent improvement for the training-free policy, while GRPO obtains most of its gain early and then plateaus. The L2E curves show continued gains across the evaluated horizons, with their magnitude varying by objective combination.

5.5 Model Scaling and Robustness

Results with a larger model. With a 7B backbone, L2E achieves LogP RI of 19.899 ($\approx 2\times$ RLOO's 10.058) and MR RI of 6.055 ($\approx 2.3\times$ RLOO's 2.603) (Tab. 6). Paired per-anchor RI differences are significant for QED, LogP, and MR.

Robustness across initial property values. Under the five-proposal budget, L2E has higher LogP and MR RI than RLOO across all three initial-property tertiles, while QED remains close (Fig. 9). In the highest LogP tertile, L2E reaches 10.085 ± 0.272 RI versus 4.955 ± 0.246 for RLOO. LogP RI decreases with the initial property value for both methods; this metric expresses improvement relative to the initial property value. Appendix D.1 reports SR and RI for each tertile.

5.6 EGFR Docking Analysis

Figure 8 shows responses during training with EGFR docking feedback using DOCKSTRING (García-Ortegón et al., 2022). Their mean scores are -8.271, -8.030, and -7.646 kcal/mol, respectively, with best scores of -12.40, -11.80, and -10.10. Among the best candidates, L2E's mean is -10.309 versus GRPO's -9.996, indicating a better-scoring shortlist.

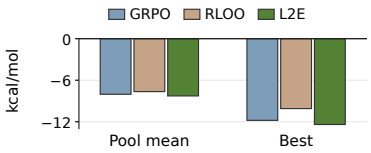

Figure 8: **EGFR docking.** Mean and best score. (lower is better).

6 Conclusion

L2E trains molecular optimizers from search experience through global trajectory assessment and local molecular-score comparisons. Under matched proposal budgets, it achieves larger relative improvements on LogP and MR than outcome-level RL, with BDP gains extending to unseen instructions. Additional refinement turns yield further progress across the evaluated objectives. These results support local-global credit assignment as a way to learn which optimization decisions to repeat, linking turn-level molecular scores with trajectory outcomes and continued search progress.

AI USE STATEMENT

We used generative AI tools to assist with manuscript revision, language editing, and LaTeX formatting. The authors verified the claims, equations, citations, and results and retain full responsibility for all content, including AI-assisted text. The language models used as molecular optimization policies are described in the method and experimental settings.

ETHICS STATEMENT

L2E studies evaluator-guided molecular optimization, with potential applications in drug discovery and materials design. The experiments use molecular benchmark datasets and computational evaluators and involve no human-subject research, personally identifiable information, or deceptive data collection. More capable molecular optimizers could also be directed toward harmful chemical objectives. Practical use therefore requires careful choice of optimization targets, expert review, and independent assessment of candidate safety. Synthesizability, biological activity, and suitability for use require experimental validation.

REPRODUCIBILITY STATEMENT

Section 4 specifies the molecular scoring rules, local and global advantages, and policy update. Section 5.1 identifies the benchmarks, baselines, and evaluation metrics. Appendix A documents the training procedure, baseline configurations, sampling settings, prompt templates, output-selection rules, and metric definitions; Appendix A.2 lists the training hyperparameters and proposal budget. Appendix C reports additional paired comparisons, and Appendix B provides the theoretical assumptions and proofs. Each experiment specifies its proposal budget, selection rule, and statistical unit.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *ICLR*, 2026.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *ACL*, 2024.

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

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

Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 2015.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *NeurIPS*, 2021.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 2012.

Jingyi Chai, Shuo Tang, Rui Ye, Yuwen Du, Xinyu Zhu, Mengcheng Zhou, Yanfeng Wang, Weinan E, Yuzhi Zhang, Linfeng Zhang, and Siheng Chen. Scimaster: Towards general-purpose scientific ai agents, part i. x-master as foundation: Can we lead on humanity's last exam? *arXiv preprint arXiv:2507.05241*, 2025.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Vishal Dey, Xiao Hu, and Xia Ning. Gellm^{3}o: Generalizing large language models for multi-property molecule optimization. In *ACL*, 2025.

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

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In *NeurIPS*, 2025.

Tianfan Fu, Wenhao Gao, Connor Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. In *NeurIPS*, 2022.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In *AAAI*, 2024.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: a benchmark for practical molecular optimization. In *NeurIPS*, 2022.

Miguel García-Ortegón, Gregor NC Simm, Austin J Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. Dockstring: easy molecular docking yields better benchmarks for ligand design. *Journal of chemical information and modeling*, 2022.

Christoph Grebner, Alejandro Corrochano-Navarro, Christian Buning, Hans Matter, María Méndez, Sven Ruf, Jan H Griwatz, Elisabeth Speckmeier, Thorsten Sadowski, Jiří Vymětal, et al. Exploration and validation of large language models as tools for molecular optimization. *Communications Chemistry*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 2025a.

Jeff Guo and Philippe Schwaller. Augmented memory: Capitalizing on experience replay to accelerate de novo molecular design. *arXiv preprint arXiv:2305.16160*, 2023.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. In *NeurIPS*, 2025b.

Hassan Harb, Magali S Ferrandon, Timothy A Goetjen, Seryeong Lee, Omar K Farha, Massimiliano Delferro, and Rajeev Surendran Assary. Discovery of hydrogen storage molecules using large language models and machine learning. *Digital Discovery*, 2026.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *ICLR*, 2024.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 2022.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 2019.

William L Jorgensen. Efficient drug lead discovery and optimization. *Accounts of chemical research*, 2009.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *TPAMI*, 2026.

Robert Lange, Yuki Imajuku, and Edoardo Cetin. Shinkaevolve: Towards open-ended and sample-efficient program evolution. In *ICLR*, 2026.

R Jo W Le Fevre. Molecular refractivity and polarizability. In *Advances in Physical Organic Chemistry*. 1965.

Jiatong Li, Junxian Li, Weida Wang, Yunqing Liu, Changmeng Zheng, Yatao Bian, Dongzhan Zhou, Xiao-Yong Wei, and Qing Li. Speak-to-structure: Evaluating llms in open-domain natural language-driven molecule generation. In *KDD*, 2026.

Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, Xinrui Xiong, et al. Chemvlm: Exploring the power of multimodal large language models in chemistry area. In *AAAI*, 2025.

Christopher Lipinski and Andrew Hopkins. Navigating chemical space for biology and medicine. *Nature*, 2004.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 2001.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Conversational drug editing using retrieval and domain feedback. In *ICLR*, 2024a.

Wei Liu, Jiawei Xu, Yingru Li, Longtao Zheng, Tianjian Li, Qian Liu, and Junxian He. Dr. kernel: Reinforcement learning done right for triton kernel generations. *arXiv preprint arXiv:2602.05885*, 2026. URL https://arxiv.org/abs/2602.05885v2.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In *EMNLP*, 2023.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, et al. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. In *NeurIPS*, 2024b.

Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

Kenneth López-Pérez, Juan F Avellaneda-Tamayo, Lexin Chen, Edgar López-López, K Eurídice Juárez-Mercado, José L Medina-Franco, and Ramón Alain Miranda-Quintana. Molecular similarity: Theory, applications, and perspectives. *Artificial Intelligence Chemistry*, 2(2):100077, 2024.

Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. Octotools: An agentic framework with extensible tools for complex reasoning. *arXiv preprint arXiv:2502.11271*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Siddharth Narayanan, James Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Michael Pieler, Sam Rodriques, and Andrew White. Training a scientific reasoning model for chemistry. In *NeurIPS*, 2025.

Tung Nguyen and Aditya Grover. Lico: Large language models for in-context molecular optimization. In *ICLR*, 2025.

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

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 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.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *EMNLP*, 2023.

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

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *ICLR*, 2024.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Harikrishna Sahu, Wei Xiong, Anagha Savit, Shivank S Shukla, and Rampi Ramprasad. Polyt5: an encoder-decoder foundation chemical language model for generative polymer design. *npj Artificial Intelligence*, 2(1):30, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ning Shang, Yifei Liu, Yi Zhu, Li Lyna Zhang, Weijiang Xu, Xinyu Guan, Buze Zhang, Bingcheng Dong, Xudong Zhou, Bowen Zhang, et al. rstar2-agent: Agentic reasoning technical report. *arXiv preprint arXiv:2508.20722*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300v3*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Xingyu Su, Xiner Li, Masatoshi Uehara, Sunwoo Kim, Yulai Zhao, Gabriele Scalia, Ehsan Hajiramezanali, Tommaso Biancalani, Degui Zhi, and Shuiwang Ji. Iterative distillation for reward-guided fine-tuning of diffusion models in biomolecular design. In *ICLR*, 2026.

Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. TR2-D2: Tree search guided trajectory-aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 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 *ICLR*, 2025a.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073v1*, 2025b.

Ziqing Wang, Yibo Wen, William Pattie, Xiao Luo, Weimin Wu, Jerry Yao-Chieh Hu, Abhishek Pandey, Han Liu, and Kaize Ding. Polo: Preference-guided multi-turn reinforcement learning for lead optimization. *arXiv preprint arXiv:2509.21737*, 2025c.

Ziqing Wang, Yibo Wen, Abhishek Pandey, Han Liu, and Kaize Ding. Molmem: Memory-augmented agentic reinforcement learning for sample-efficient molecular optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 43694–43712. ACL, 2026. doi: 10.18653/v1/2026.acl-long.2024. URL https://aclanthology.org/2026.acl-long.2024/.

Di Wu, Qi Chen, Xiaojie Chen, Feng Han, Zhong Chen, and Yi Wang. The blood–brain barrier: Structure, regulation and drug delivery. *Signal transduction and targeted therapy*, 8(1):217, 2023.

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, et al. Nature language model: Deciphering the language of nature for scientific discovery. *arXiv preprint arXiv:2502.07527*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115v1*, 2024a.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *ICLR*, 2024b.

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.

Oussama Zekri and Nicolas Boullé. Fine-tuning discrete diffusion models with policy gradient methods. In *Advances in Neural Information Processing Systems*, 2025.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025b.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *arXiv preprint arXiv:2507.22844*, 2025c.

Zihan Zhao, Bo Chen, Ziping Wan, Lu Chen, Xuanze Lin, Shiyang Yu, Situo Zhang, Da Ma, Zichen Zhu, Danyang Zhang, et al. Chemdfm-r: A chemical reasoning llm enhanced with atomized chemical knowledge. *arXiv preprint arXiv:2507.21990v3*, 2025a.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, et al. Developing chemdfm as a large language foundation model for chemistry. *Cell Reports Physical Science*, 6(4), 2025b.

Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Chang-Yu Hsieh, Tingjun Hou, and Jian Wu. Sample-efficient multi-objective molecular optimization with gflownets. In *NeurIPS*, 2023.

APPENDIX CONTENTS

A Implementation and Evaluation Protocols **17**
A.1 Training procedure . . . 17
A.2 Training hyperparameters . . . 17
A.3 Benchmark settings . . . 17
A.4 Prompt templates . . . 18
A.5 Evaluation metrics . . . 19
A.6 Molecular properties . . . 20

B Proofs and Boundary Conditions for Local–Global Advantages **20**
B.1 Gradients and objectives . . . 21
B.2 Centered credit and the covariance identity . . . 21
B.3 Proof of conditional search improvement . . . 22
B.4 Immediate and delayed contributions . . . 23
B.5 An illustrative family with improved final selection . . . 24

C Additional Benchmark Results **24**
C.1 Paired benchmark comparisons . . . 24
C.2 Supplementary GEPA and POLO results . . . 25
C.3 Credit variants . . . 25

D Robustness and Failure Analyses **25**
D.1 Robustness across initial property values . . . 25
D.2 Joint constraints at an extended proposal budget . . . 25
D.3 Structural Growth in the Local-only Variant . . . 25

E Related Work **25**
E.1 Training-free Evolution . . . 26
E.2 RL for Multi-turn Optimization . . . 27
E.3 Tool Learning and Scientific Agents . . . 27
E.4 Molecular Optimization and Related Search Problems . . . 27

F Limitations and Broader Impact **28**
F.1 Limitations . . . 29
F.2 Broader impact . . . 29

A Implementation and Evaluation Protocols

This appendix specifies the score-standardized, token-mean training procedure in Section 4, followed by training hyperparameters, evaluation settings, and prompt templates.

A.1 Training procedure

Score standardization. For a group of G trajectories with B responses each, the statistics used in Eq. 4 are

$$\mu_i = \frac{1}{B}\sum_{t=1}^{B} r_{i,t}, \quad \sigma_i = \sqrt{\frac{1}{B}\sum_{t=1}^{B}(r_{i,t}-\mu_i)^2}, \\ \mu = \frac{1}{G}\sum_{i=1}^{G} r_i, \qquad \sigma = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(r_i-\mu)^2}. \tag{8}$$

The score stabilizer $\epsilon_{\text{std}} > 0$ and clipping threshold ϵ_{clip} are distinct. Equal molecular scores within a trajectory give zero local advantage; equal cumulative molecular scores across trajectories give every trajectory in the group zero global advantage.

Token probabilities and masking. Let $s_{i,t-1}$ contain the task, initial molecule, and recent molecule–feedback history. For token ℓ of response $e_{i,t}$, the probability ratio is

$$\rho_{i,t,\ell}(\theta) = \frac{f_\theta(e_{i,t,\ell} \mid s_{i,t-1}, e_{i,t,<\ell})}{f_{\text{old}}(e_{i,t,\ell} \mid s_{i,t-1}, e_{i,t,<\ell})}. \tag{9}$$

The response includes its generated reasoning and molecule text. These tokens share $A_{i,t}$ from Eq. 4. Prompt and evaluator tokens receive no policy loss. The clipped loss averages tokens within each response, then turns and trajectories, as in Eq. 5. The molecular scores and the resulting advantages are held fixed throughout each policy update.

Algorithm 1 Learning to Evolve

Require: Policy f_θ, reference policy f_{ref}, optimization instances, group size G, proposal budget B, memory length M, and local advantage weight ω.

```
Freeze the behavior policy f_old ← f_θ.
for each optimization instance x_0 do
    for each trajectory i = 1, ..., G do
        Initialize history with the initial molecule and its evaluator feedback.
        for each turn t = 1, ..., B do
            Generate full response e_{i,t} using f_old and the recent history.
            Evaluate the proposed molecule, if parsed, and compute r_{i,t} using Eq. 3.
            Retain the full response for training; append any parsed molecule and feedback to history.
        end for
    end for
    Compute cumulative molecular scores as defined in Section 4.2.
    Compute global advantage, local advantage, and turn advantages using Eq. 4.
end for
Update f_θ with the clipped objective and reference-policy KL penalty in Eq. 5.
Repeat with fresh trajectories until the training budget is exhausted.
```

A.2 Training hyperparameters

Table 7 reports the training configuration for the single-property experiments.

A.3 Benchmark settings

We use `verl` (Sheng et al., 2025) for training. For general-purpose LLMs, we use the model from the corresponding HuggingFace repositories. All system prompts remain defaults.

Table 7: Training configuration for the single-property experiments. The default local weight is $\omega = 1$ and the similarity threshold is 0.4.

Setting	Value
Batch size (prompts per step)	128
Group size (trajectories per prompt)	16
Max turns per trajectory (train)	5
Evaluation proposals	5
Training seed	42
Learning rate	$1e{-6}$
KL coefficient	$1e{-3}$
Optimizer	AdamW (β_1=0.9, β_2=0.999)
Gradient clipping	1.0

Baseline configurations. GraphGA uses PMO's default graph operators (Gao et al., 2022), with the proposal budget matched to L2E. REINVENT uses the default PMO configuration. For the LLM-assisted evolution method, MOLLEO, we use Qwen2.5-3B as the base model.

Property aggregation. We assign a uniform weight $\alpha_j = 1$ to every target property in the aggregate score defined in Section 2, giving $q(x) = \sum_{j=1}^{n} g_j(P_j(x))$. These weights introduce no prior preference for one target property over another. The aim is for the policy to explore molecular modifications and learn how to balance multiple properties through evaluator feedback.

Search protocol. For the main LLM and RL comparisons, each inference run uses one trajectory per initial molecule, with a fixed seed for each run. Final output selection includes the initial molecule and the feasible generated candidates, as in Eq. 2. The initial molecule provides an available output when no candidate improves the selection objective.

Sampling parameters. All LLMs are inferred with consistent sampling parameters; we set the temperature as 0.4 for single-property optimization, and the context window for each search turn is set to 2048. For multi-property optimization, we set the context window as 4096.

Training and statistical units. Appendix A.2 gives the common training configuration. For the main benchmark comparisons, we repeat inference three times and report the mean and sample standard deviation. Main benchmark results use fixed checkpoints; repeated inference or search measures decoding or search variability. Paired benchmark comparisons first average inference repeats within each shared anchor and then apply a two-sided Wilcoxon signed-rank test across anchors.

Multi-property task cells and splits. B, D, P, and Q denote BBBP, DRD2, penalized LogP, and QED. BDP, BDQ, and BPQ each require improvement in all three named properties in the instructed directions. We use the supplied MuMOInstruct seen and unseen instruction partitions. Both partitions contain the same three objective sets. Throughout the paper, "unseen" refers to this instruction partition. The split evaluates generalization to new instructions over the same property combinations.

A.4 Prompt templates

Tables 8 and 9 give the single- and multi-property prompts. At the initial turn, omit the marked history block. At subsequent turns, insert it after the task description. The task and memory placeholders are filled with the instruction and recent evaluator history. Each template specifies its output and feedback tags.

Table 8: Single-property prompt. The marked history block is inserted only after the initial turn; italic labels describe template assembly and are not sent to the model.

You are an expert medicinal chemist specializing in molecular optimization. Your goal is to propose a new molecule that satisfies the given constraints. Ensure the proposed molecule in SMILES format is valid, plausible, and does not duplicate the previous molecules. Prefer small edits that preserve chemical similarity.
Your task:
{task_description}

Insert the following block at subsequent turns only:
Prior to this step, you have already taken {step_count} step(s).
Below is the interaction history, where `<answer> </answer>` wrapped your past molecules and <results> </results> wrapped the corresponding evaluator feedback:
{memory_context}

Continue at every turn:
Now it's your turn to respond to the current step.
You should first conduct a reasoning process, which MUST be enclosed within `<think> </think>` tags.
After finishing your reasoning, suggest a valid molecule in SMILES format that either extends a previously successful modification or explores a new promising direction expected to satisfy the given constraints.
If you are ready to provide the self-contained solution, provide the molecule only inside `<answer>` ... `</answer>`.

Table 9: Multi-property prompt. The history block is inserted only after the initial turn. Output and feedback tags are shown as used during evaluation.

You are an expert medicinal chemist specializing in multi-property molecular optimization. Your goal is to propose a new molecule that satisfies the given constraints across all target properties. Ensure the proposed molecule in SMILES format is valid, plausible, and does not duplicate previous molecules. Prefer small edits that preserve chemical similarity.
Your task:
{task_description}

Insert the following block at subsequent turns only:
Prior to this step, you have already taken {step_count} step(s).
Below is the interaction history, where `<SMILES> </SMILES>` wrapped your past molecules and <result> </result> wrapped the corresponding evaluator feedback:
{memory_context}

Continue at every turn:
Now it's your turn to respond to the current step.
You should first conduct a reasoning process, which MUST be enclosed within `<think> </think>` tags.
After finishing your reasoning, suggest a valid molecule in SMILES format that either extends a previously successful modification or explores a new promising direction expected to satisfy the given constraints.
If you are ready to provide the self-contained solution, provide the molecule only inside `<SMILES>` ... `</SMILES>`.

A.5 Evaluation metrics

Success Rate (SR). SR is the fraction of test cases in which the optimized molecule improves all target properties while satisfying the similarity constraint $\mathrm{sim}(m, m') \geq 0.4$. A successful optimization must meet the similarity constraint and achieve improvement in the desired direction

for every target property. Formally, let $\Delta P_{i,j} = P_j(m_i') - P_j(m_i)$ and define $\mathrm{sgn}(w_j) = +1$ for properties to maximize and -1 for properties to minimize. SR is computed as:

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[v(m_i') = 1 \ \wedge \ \mathrm{sim}(m_i, m_i') \geq 0.4 \ \wedge \ \bigwedge_{j=1}^{n}\Big(\mathrm{sgn}(w_j)\cdot\Delta P_{i,j} > 0\Big)\right], \tag{10}$$

where $v(m_i')$ is the feasibility indicator for the evaluation block, N is the test set size, m_i is the i-th molecule, m_i' is its optimized variant, and n is the number of target properties.

Similarity (Sim). Sim is the average Tanimoto similarity across all test cases:

$$\mathrm{Sim} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{sim}(m_i, m_i'). \tag{11}$$

When the initial molecule is selected because no feasible proposal improves the selection objective, we set $m_i' = m_i$, so $\mathrm{sim}(m_i, m_i') = 1.0$.

Relative Improvement (RI). For test case i, let s_i be the success indicator in the SR definition. We average directed relative gains over the n target properties, then over the N test cases:

$$\mathrm{RI}_i = \begin{cases} \dfrac{1}{n}\displaystyle\sum_{j=1}^{n}\mathrm{sgn}(w_j)\frac{P_j(m_i') - P_j(m_i)}{d_{i,j}}, & s_i = 1, \\ 0, & s_i = 0, \end{cases}$$
$$\mathrm{RI} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{RI}_i. \tag{12}$$

Here $d_{i,j} = |P_j(m_i)|$ when the initial property value is nonzero, and $d_{i,j} = 10^{-8}$ otherwise. Thus N counts test cases and n counts target properties; $n = 1$ for single-property optimization. Failed cases contribute zero and remain in the denominator.

A.6 Molecular properties

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

- `QED` (Quantitative Estimation of Drug-likeness) (Bickerton et al., 2012): A composite drug-likeness score that summarizes how well a molecule matches common medicinal-chemistry priors.
- `LogP` (lipophilicity) (Lipinski et al., 2001): The octanol/water partition coefficient. Larger LogP indicates higher lipophilicity.
- `plogP` (penalized logP): A modified LogP objective that subtracts penalties related to synthetic accessibility and undesirable ring structures (e.g., overly large rings), encouraging molecules that are both lipophilic and chemically plausible.
- `MR` (molar refractivity) (Le Fevre, 1965): A physicochemical descriptor associated with molecular volume and electronic polarizability, commonly used as a proxy for how strongly a molecule may interact with its environment.
- `BBBP` (blood-brain barrier permeability) (Wu et al., 2023): A measure of whether a molecule can cross the blood-brain barrier.
- `DRD2` (dopamine receptor D2 binding affinity) (Fan et al., 2020): An indicator of a molecule's binding strength to the D2 dopamine receptor. Higher affinity implies stronger receptor engagement.

B Proofs and Boundary Conditions for Local–Global Advantages

We prove the covariance identity and the conditional search-improvement result in Section 4.4, then analyze the effect of delayed rewards. All results use the score-standardized credit and response-level token mean in Eqs. 4 and 5. All GB turns contribute to training, including those with chemically invalid, rejected, or repeated proposals. Standard deviations use population denominators, $\epsilon_{\mathrm{std}} > 0$, and $G, B \geq 2$ unless stated otherwise. When $B = 1$, local advantage is zero.

B.1 GRADIENTS AND OBJECTIVES

Fix an optimization instance x_0 and sample G independent trajectories with B responses. For a complete response of length $\ell_{i,t} = |e_{i,t}|$, define

$$g_{i,t} = \nabla_\theta \log f_\theta(e_{i,t} \mid x_0, H_{i,t-1}), \qquad h_{i,t} = g_{i,t}/\ell_{i,t}. \tag{13}$$

The response probability is that of the sampling distribution, including its stopping rule and end-of-sequence probability. Prompt and evaluator tokens are excluded. Omitting generated action tokens or stopping events changes the complete-response score and requires a separate correction. For action-dependent lengths, $h_{i,t}$ need not have the zero conditional mean of $g_{i,t}$.

At $\theta = \theta_{\text{old}}$, with scores and advantages fixed, the initial reward gradient of Eq. 5 is

$$\begin{aligned} \widehat{d}_G &= \frac{1}{GB}\sum_{i,t} C_i^{\text{traj}} h_{i,t}, \quad \widehat{d}_L = \frac{1}{GB}\sum_{i,t} C_{i,t}^{\text{turn}} h_{i,t}, \\ \widehat{d}_\omega &= \widehat{d}_G + \omega \widehat{d}_L. \end{aligned} \tag{14}$$

Token ratios equal one at this point, so clipping is locally inactive. A common KL gradient can be added to both compared updates. Multiple PPO epochs, gradient clipping, and adaptive optimizer states can change the resulting parameter displacement.

We distinguish the cumulative-score objective and final-selection objective:

$$J_\Sigma(\theta) = \mathbb{E}_\theta \sum_{t=1}^{B} r_t, \qquad J_\star(\theta) = \mathbb{E}_\theta b_B. \tag{15}$$

Here b_B uses the task's selection score and includes the initial molecule, as in Eq. 2. These objectives can rank trajectories differently even when training and selection scores agree on feasible proposals. Under that agreement, if $b_0 \le 0.6$, the feasible score sequences $(0.6, 0.6)$ and $(0, 1)$ have cumulative scores 1.2 and 1, but best scores 0.6 and 1, respectively.

B.2 CENTERED CREDIT AND THE COVARIANCE IDENTITY

The following identity proves Proposition 4.1 and gives the mean and between-turn differences of the resulting turn advantages.

Proposition B.1 (Centered score credit)**.** *For every realized batch, let $\bar{h}_i = B^{-1}\sum_t h_{i,t}$. Then*

$$\begin{aligned} \frac{1}{B}\sum_t A_{i,t} &= C_i^{\text{traj}}, \\ A_{i,t} - A_{i,u} &= \frac{\omega(r_{i,t} - r_{i,u})}{\sigma_i + \epsilon_{\text{std}}}, \\ \widehat{d}_L &= \frac{1}{G}\sum_i \frac{\text{Cov}_B(\mathbf{r}_i, h_i)}{\sigma_i + \epsilon_{\text{std}}}, \end{aligned} \tag{16}$$

where $\text{Cov}_B(\mathbf{r}_i, h_i) = B^{-1}\sum_t (r_{i,t} - \mu_i)(h_{i,t} - \bar{h}_i)$. Equivalently,

$$\widehat{d}_L = \frac{1}{GB^2}\sum_i \frac{\sum_{t<u}(r_{i,t} - r_{i,u})(h_{i,t} - h_{i,u})}{\sigma_i + \epsilon_{\text{std}}}. \tag{17}$$

For $\omega > 0$, turn advantages preserve the strict ordering and ties of molecular scores within each trajectory.

Proof. The equality $\sum_t (r_{i,t} - \mu_i) = 0$ gives the mean-advantage identity and permits subtraction of $\bar{h}_i$ inside the local gradient. Subtracting two advantages cancels both the global coefficient and the local mean, giving the turn-advantage difference. Because $\sigma_i + \epsilon_{\text{std}} > 0$, the ordering follows for $\omega > 0$.

For the pairwise representation, expansion gives

$$\begin{aligned}\sum_{t<u}(r_{i,t}-r_{i,u})(h_{i,t}-h_{i,u}) &= B\sum_t r_{i,t}h_{i,t}-\Big(\sum_t r_{i,t}\Big)\Big(\sum_t h_{i,t}\Big)\\ &= B\sum_t (r_{i,t}-\mu_i)(h_{i,t}-\bar{h}_i).\end{aligned} \tag{18}$$

Dividing by $GB^2(\sigma_i+\epsilon_{\mathrm{std}})$ and summing over trajectories proves Eq. 17. These finite-batch identities hold for dependent turns, variable response lengths, and trajectory-specific standard deviations. □

Corollary B.2 (Permutation null)**.** *Condition on a realized batch $\mathcal{B}$, including its responses and histories. Uniformly permute the molecular scores within each trajectory while holding response gradients fixed. The resulting local direction satisfies*

$$\mathbb{E}_\pi[\widehat{d}_L^{\pi} \mid \mathcal{B}] = 0, \qquad \widehat{d}_G^{\pi} = \widehat{d}_G. \tag{19}$$

Proof. Every permuted position has expected score μ_i, including when scores are tied. Its expected centered coefficient is therefore zero. The score sum and standard deviation are invariant under permutation, so the global coefficient and local denominator remain unchanged. Linearity proves the result for both independent and dependent permutations across trajectories. □

The permutation changes the advantage–response pairing on a fixed batch. The covariance measures how molecular scores and response gradients vary together across the observed histories; comparing alternative responses from a common history requires conditioning on that history. The L2E gradient differs from the global-only gradient by $\omega\widehat{d}_L$, which is determined by this covariance. Score variation can coexist with zero local covariance.

B.3 Proof of conditional search improvement

Fix $\eta > 0$. All expectations below are over the batch sampled from the common initial policy. Define

$$a_\star = \mathbb{E}\langle \nabla J_\star(\theta_G), \widehat{d}_L\rangle, \qquad M_L = \mathbb{E}\|\widehat{d}_L\|^2. \tag{20}$$

Assume the alignment product is absolutely integrable, $0 < a_\star < \infty$, and $M_L < \infty$. There exist deterministic constants $\bar{\omega} > 0$ and $0 < L_J < \infty$ and a common region on which $J_\star$ is differentiable with L_J-Lipschitz gradient. Almost surely, this region contains every segment from θ_G to $\theta_G+\eta\omega\widehat{d}_L$ for $0 \le \omega \le \bar{\omega}$. Also assume $J_\star(\theta_G)$ and $J_\star(\theta_G+\eta\omega\widehat{d}_L)$ are integrable throughout this interval. The result requires these local regularity and moment assumptions in addition to score centering.

Proof of Theorem 4.2. Write $\Delta = \theta_L - \theta_G$. Under the uniform gradient-Lipschitz bound above, integration along the segment between the endpoints gives

$$\begin{aligned}J_\star(\theta_G+\Delta) - J_\star(\theta_G) &= \langle \nabla J_\star(\theta_G), \Delta\rangle\\ &\quad + \int_0^1 \langle \nabla J_\star(\theta_G+t\Delta) - \nabla J_\star(\theta_G), \Delta\rangle\, dt\\ &\ge \langle \nabla J_\star(\theta_G), \Delta\rangle - \frac{L_J}{2}\|\Delta\|^2.\end{aligned} \tag{21}$$

The two one-step SGD updates share the initial policy, batch, learning rate, and KL contribution, hence $\Delta = \eta\omega\widehat{d}_L$. Taking expectations yields

$$\mathbb{E}[J_\star(\theta_L) - J_\star(\theta_G)] \ge \eta\omega a_\star - \frac{L_J}{2}(\eta\omega)^2 M_L. \tag{22}$$

The assumption $a_\star > 0$ implies $M_L > 0$, since $M_L = 0$ would force $\widehat{d}_L = 0$ almost surely. Thus the right-hand side is positive when $0 < \eta\omega < 2a_\star/(L_J M_L)$ and $\omega \le \bar{\omega}$. This nonempty deterministic interval proves the claim. The bound allows dependence between the global endpoint and local direction. □

The target-alignment quantity. Define $z_{i,t} = \langle \nabla J_\star(\theta_G), h_{i,t}(\theta_{\text{old}}) \rangle$. For each batch, the target gradient is shared by all responses. Proposition B.1 therefore gives

$$a_\star = \mathbb{E}\left[\frac{1}{G} \sum_i \frac{\text{Cov}_B(\mathbf{r}_i, z_i)}{\sigma_i + \epsilon_{\text{std}}} \right]. \tag{23}$$

For fixed η, the global endpoint and unit-weight local direction are determined without ω. The sufficient interval for ω uses one deterministic L_J valid on a common region containing the endpoint segments under consideration. Establishing $a_\star > 0$ requires alignment with the target gradient in Eq. 23.

The smoothness inequality in Eq. 21 also applies to endpoints of more general optimizers. For adaptive optimization or repeated PPO updates, it uses their actual parameter displacement in place of the one-step SGD displacement $\eta\omega\widehat{d}_L$.

B.4 Immediate and delayed contributions

Assume a fixed horizon, parameter-independent evaluator transitions, regular policy support, integrable score–reward products, and interchange of differentiation and expectation. For complete-response gradients, define

$$D = \sum_t \mathbb{E}[r_t g_t], \qquad F = \sum_{t<k} \mathbb{E}[r_k g_t]. \tag{24}$$

The complete causal history $\mathcal{F}_{t-1}$ fixes the policy's memory window. The score identity $\mathbb{E}[g_t \mid \mathcal{F}_{t-1}] = 0$ implies $\mathbb{E}[r_k g_t] = 0$ for $k < t$. Consequently,

$$g_\Sigma := \nabla J_\Sigma = \mathbb{E}\left[\left(\sum_k r_k \right) \left(\sum_t g_t \right) \right] = D + F. \tag{25}$$

Thus D describes immediate contributions and F describes effects on subsequent scores.

Proposition B.3 (Temporal alignment at a fixed reference scale)**.** *For fixed $d, \ell > 0$, let $d_L^0 = (Bd\ell)^{-1}\mathbb{E}\sum_t (r_t - \mu) g_t$, with $\mu = B^{-1}\sum_t r_t$. Then*

$$d_L^0 = \frac{(B-1)D - F}{B^2 d\ell}. \tag{26}$$

If $D \neq 0$, write $F = \alpha D + F_\perp$, where $\langle D, F_\perp \rangle = 0$, and let $\rho = \|F_\perp\|/\|D\|$. Its alignment with the cumulative objective is

$$\langle g_\Sigma, d_L^0 \rangle = \frac{\|D\|^2}{B^2 d\ell} \left[(1+\alpha)(B-1-\alpha) - \rho^2 \right]. \tag{27}$$

Proof. Expanding the trajectory mean and using Eq. 25 gives

$$\mathbb{E}\sum_t (r_t - \mu) g_t = D - \frac{1}{B}\sum_{t,k} \mathbb{E}[r_k g_t] = \frac{(B-1)D - F}{B}. \tag{28}$$

Multiplication by $1/(Bd\ell)$ proves the first identity. Taking its inner product with $D + F$ and using $F_\perp \perp D$ proves the second. □

If $D \neq 0$ and $\|F\| \leq \delta \|D\|$ for $0 \leq \delta < 1$, Cauchy–Schwarz gives the sufficient positive margin

$$\begin{aligned} B^2 d\ell \, \langle g_\Sigma, d_L^0 \rangle &= (B-1)\|D\|^2 + (B-2)\langle D, F \rangle - \|F\|^2 \\ &\geq (1-\delta)(B-1+\delta)\|D\|^2 > 0. \end{aligned} \tag{29}$$

Outside this sufficient condition, Eq. 27 determines the alignment sign.

Random normalization and token averaging. The fixed-scale reference differs from the actual local update. For one trajectory, let $Z_t = r_t - \mu$, let σ_{within} be its population score standard deviation, and set $c_t = [(\sigma_{\text{within}} + \epsilon_{\text{std}})\ell_t]^{-1}$ and $c_0 = (d\ell)^{-1}$. If the reference expectations exist and

$$\varepsilon_{\text{norm}}^2 := \mathbb{E}\frac{1}{B}\sum_t Z_t^2 \|g_t\|^2 (c_t - c_0)^2 < \infty, \tag{30}$$

then

$$\mathbb{E}\widehat{d}_L = d_L^0 + R, \qquad R = \mathbb{E}\frac{1}{B}\sum_t Z_t g_t (c_t - c_0), \qquad \|R\| \le \varepsilon_{\text{norm}}. \tag{31}$$

Indeed, adding and subtracting c_0 gives the equality, and Jensen's inequality bounds the squared norm of the expected average by the expected average squared norm. The trajectories have the same marginal law, so their group average has the same expectation. Hence

$$\langle g_\Sigma, \mathbb{E}\widehat{d}_L \rangle \ge \langle g_\Sigma, d_L^0 \rangle - \|g_\Sigma\| \varepsilon_{\text{norm}}. \tag{32}$$

This bound allows dependence among scores, lengths, actions, and histories. A reference margin larger than the displayed normalization-error bound ensures positive alignment.

B.5 An illustrative family with improved final selection

Consider independent latent proposals $a_t = \theta + sZ_t$, with $Z_t \sim \mathcal{N}(0,1)$ and $s > 0$. Suppose every proposal is feasible and its training and selection scores agree: $r_t = S_{\text{sel}}(x_t; x_0) = \phi(a_t)$ for a bounded strictly increasing function ϕ. Assume $b_0 < \sup \phi$ and response length fixed at $\ell > 0$.

Couple fresh proposals at $\theta' > \theta$ using the same Z_t. Every molecular score strictly increases, so J_Σ strictly increases. The maximum including b_0 is nondecreasing and strictly increases on the positive-probability event that an old proposal exceeds b_0 in score. Boundedness then implies that $J_\star$ is strictly increasing as well.

For a sampled trajectory, $g_t = (a_t - \theta)/s^2$. The pairwise representation gives

$$\frac{\sum_t (r_t - \mu)(a_t - \theta)}{B\ell s^2(\sigma_{\text{within}} + \epsilon_{\text{std}})} = \frac{\sum_{t<u}(r_t - r_u)(a_t - a_u)}{B^2 \ell s^2 (\sigma_{\text{within}} + \epsilon_{\text{std}})} > 0 \tag{33}$$

almost surely: independent continuous proposals are distinct and ϕ is strictly increasing. Averaging over trajectories preserves positivity. One SGD step from a shared batch with a common KL contribution and $\eta, \omega > 0$ consequently gives $\theta_L - \theta_G = \eta\omega\widehat{d}_L > 0$, and hence $J_\star(\theta_L) > J_\star(\theta_G)$. A common fixed positive scalar preconditioner preserves this ordering.

This example establishes improvement over the global-only endpoint in a one-dimensional monotone proposal model. At the fixed learning rate, local advantage increases the displacement in the favorable direction, and the update norms may differ. The common global and KL terms can make both endpoints worse than the initial policy. This family illustrates how the effect of local advantage depends on the relation between molecular scores, response gradients, and the final-selection objective.

C Additional Benchmark Results

The main benchmark tables report means and sample standard deviations over inference repeats from fixed checkpoints. This appendix reports paired comparisons, additional baselines, and docking candidate-pool summaries. Appendix D analyzes initial-property strata, joint constraints, and structural failures.

C.1 Paired benchmark comparisons

For QED, the median paired RI difference between L2E and RLOO is zero, and the SR comparison gives $p = 0.317$. For BDP, the RI comparisons with RLOO on seen instructions and GRPO on unseen instructions give $p < 0.001$; the corresponding SR differences are not significant. In the 7B comparison (Table 6), paired per-anchor RI differences are significant for QED, LogP, and MR ($p < 0.001$). LogP SR is lower than RLOO ($p = 0.009$), and MR SR is lower than GRPO ($p < 0.001$). The paired test and aggregation unit are specified in Appendix A.3.

C.2 Supplementary GEPA and POLO results

GEPA evolves prompts without updating model weights (Agrawal et al., 2026), while POLO combines trajectory-level policy optimization with preferences between intermediate molecules (Wang et al., 2025c). Table 3 reports the comparison results. POLO and L2E use the Qwen2.5-3B backbone. Neither POLO nor L2E uses molecular SFT in this comparison.

C.3 Credit variants

Table 4 compares L2E and its global-only and local-only ablations under a five-proposal evaluation budget. L2E retains both advantages; each ablation retains only the named component. The comparison measures how each variant affects relative property improvement (RI), success rate (SR), and molecular similarity to the initial molecule.

The local-only run has LogP RI of 157.305 and MR RI of 13.135, with mean similarities of 0.678 and 0.502. These large relative gains accompany the structural failures described in Appendix D. The L2E variant improves LogP and MR SR over the global-only run, but reduces QED SR. The L2E variant therefore trades a lower QED success rate for higher LogP and MR success rates.

D Robustness and Failure Analyses

D.1 Robustness across initial property values

We test how the single-property results vary across initial property values. For each objective, we sort the 100 shared RLOO/L2E initial molecules by their property value and divide them into low-, middle-, and high-value tertiles. We evaluate three stochastic inference repeats from one checkpoint, with one five-turn trajectory per initial molecule, as in Tab. 1. The repeats measure stochastic decoding variability within each tertile.

QED has a hard upper bound, while RI for LogP and MR divides the gain by the magnitude of the initial value. Higher initial values therefore reduce the available QED improvement and rescale relative gains on LogP and MR. Comparing L2E and RLOO within each initial-property tertile accounts for this dependence on the initial property value.

D.2 Joint constraints at an extended proposal budget

Under these joint constraints, L2E exceeds GRPO's LogP/MR SR by 8.3/10.3 percentage points and RI by 3.8×/6.7×. This evaluation selects the best output across 50 proposals per initial molecule; Table 1 uses five proposals.

D.3 Structural Growth in the Local-only Variant

The local-only variant combines high RI with extreme molecular growth. We analyze generated candidates using median heavy-atom count (HAC), molecular weight (MW), and similarity over chemically valid candidates, together with the fraction of all candidates satisfying the Tanimoto-0.4 gate. The analysis includes candidates rejected during output selection. In one evaluation run on LogP, local-only training increases median HAC from 71 to 1,218 and MW from 1,075 to 18,294. The full L2E variant reduces this size inflation by more than an order of magnitude and raises similarity. Property gains in this comparison thus accompany substantial changes in molecular size and similarity.

E Related Work

We review training-free search, multi-turn reinforcement learning, and molecular optimization.

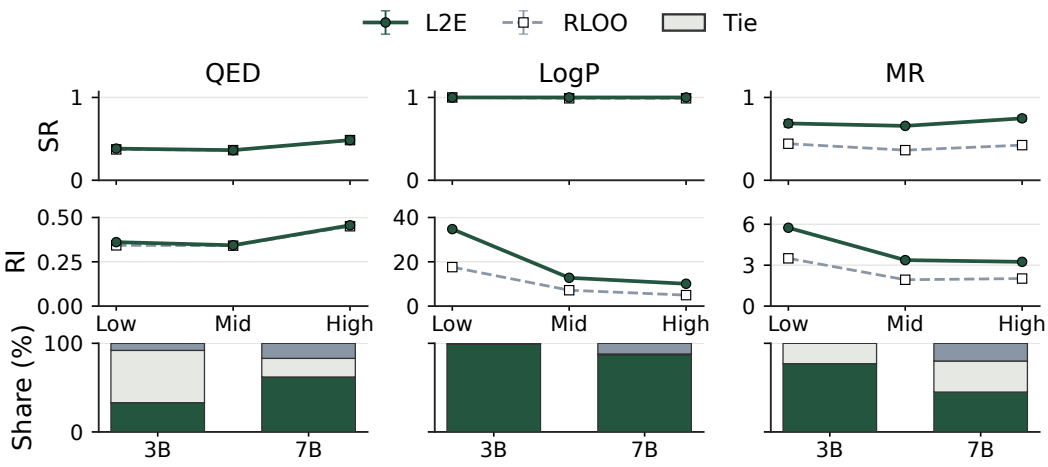

Figure 9: **Initial-property strata and paired RI coverage.** Top/middle: SR/RI by initial-property tertile (34/33/33 shared anchors), three-repeat mean/s.t.d. Bottom: L2E wins/ties/RLOO wins among all 100 anchors at 3B/7B, after averaging repeats within anchor; ties use $|\Delta\mathrm{RI}| \leq 10^{-12}$. Colors identify methods; gray denotes ties.

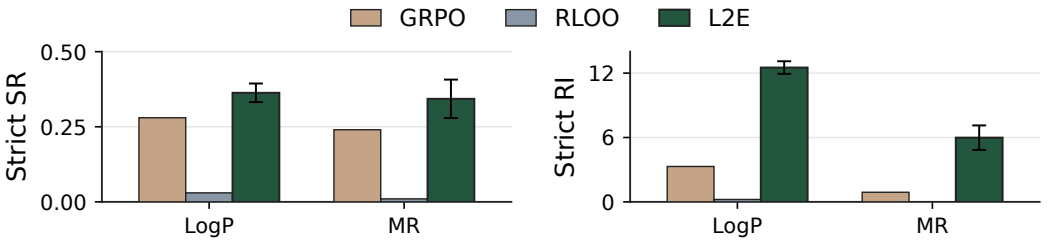

Figure 10: **Joint-constraint results at 50 proposals.** All 100 anchors; five ten-turn trajectories, best final output. Joint-constraint success requires chemical validity, one component, Sim≥ 0.4, and directional gain; RI is zero for failures. L2E: three-repeat mean/s.t.d.; GRPO/RLOO: one evaluation repeat.

E.1 Training-free Evolution

Early approaches to training-free improvement focused on iterative refinement, where a model critiques and updates its initial response. Self-Refine (Madaan et al., 2023) demonstrated that LLMs could improve their performance by generating intrinsic feedback. This concept was formalized by Reflexion (Shinn et al., 2023), which utilizes verbal reinforcement learning to maintain a memory of past errors, effectively evolving the agent's context over multiple trials. However, the limit of this approach is that models cannot often self-correct without external oracles (Huang et al., 2024). Other methods use LLMs as explicit optimizers (Yang et al., 2024b) to guide the refinement trajectory, treating the prompt or the answer as a variable to be optimized at test time.

Moving beyond single-instance refinement, population-based evolution mimics natural selection to improve reasoning at test time. The Darwin-Godel Machine (Zhang et al., 2025b) proposes an open-ended framework for evolving agents. In the domain of scientific discovery, AlphaEvolve (Novikov et al., 2025) utilizes a coding agent to evolve algorithms dynamically.

ShinkaEvolve improves program evolution through parent sampling, novelty filtering, and adaptive selection of the proposing LLM (Lange et al., 2026). These systems illustrate how search can refine candidate programs. A broader survey places such adaptation alongside changes to agent models, memory, and tools (Gao et al., 2025).

E.2 RL FOR MULTI-TURN OPTIMIZATION

Post-training spans supervised fine-tuning, reinforcement learning, and inference-time reasoning, with different sources of feedback and adaptation (Kumar et al., 2026). For iterative optimization, a central question is how feedback is assigned to the individual responses that form a search trajectory.

GRPO constructs relative advantages from groups of sampled outputs (Shao et al., 2024). RLOO uses the other samples to form a leave-one-out baseline (Ahmadian et al., 2024). When a single trajectory score supplies the advantage, every response in that trajectory receives the same credit. Other methods use turn-specific returns. For example, TRLOO compares reward-to-go across trajectories at each turn with a leave-one-out baseline (Liu et al., 2026).

GiGPO forms local comparison groups from repeated environment states across trajectories (Feng et al., 2025). POLO combines trajectory-level optimization with preferences between intermediate molecules (Wang et al., 2025c). L2E derives global advantage from cumulative molecular scores and adjusts it with local advantage within each trajectory, applying the resulting turn advantage to the complete response, including its reasoning and proposed molecule.

Segment Policy Optimization estimates advantages for segments of a reasoning chain using Monte Carlo continuations (Guo et al., 2025b). RLVMR combines final outcomes with rule-based rewards for explicitly marked planning, exploration, and reflection behaviors (Zhang et al., 2025c). These approaches study complementary choices of credit granularity and reward content. L2E uses molecular evaluator scores to assign credit to complete responses within refinement trajectories.

RL also trains models for extended reasoning and tool use. DeepSeek-R1 studies large-scale RL for reasoning (Guo et al., 2025a). RAGEN (Wang et al., 2025b) and rStar2-Agent (Shang et al., 2025) study agent training. A complementary survey organizes the broader landscape of agentic reinforcement learning across methods and application domains (Zhang et al., 2025a). L2E focuses on responses in evaluator-guided molecular search.

E.3 TOOL LEARNING AND SCIENTIFIC AGENTS

Learning to use external tools requires suitable interaction data. ToolLLM constructs instruction-following trajectories over real-world APIs and fine-tunes a tool-use model (Qin et al., 2024). APIGen synthesizes function-calling examples with format, execution, and semantic checks (Liu et al., 2024b). Confucius combines an easy-to-difficult tool curriculum with iterative data construction from introspective feedback (Gao et al., 2024).

Agent architectures also organize tool use during inference. SciMaster's X-Master uses code to interact with scientific tools (Chai et al., 2025); OctoTools coordinates planning and execution through extensible tool cards (Lu et al., 2025); and Alita constructs and reuses task-related external capabilities (Qiu et al., 2025). These systems provide context for interacting with external evaluators, while L2E studies how the resulting molecular search experience can train the proposal policy.

E.4 MOLECULAR OPTIMIZATION AND RELATED SEARCH PROBLEMS

Molecular optimization. Molecular optimization modifies a lead to improve target properties, often under a structural-similarity constraint (Bickerton et al., 2012; López-Pérez et al., 2024; Lipinski & Hopkins, 2004). Structural similarity alone does not guarantee preserved biological function. Because property oracles can be expensive, PMO evaluates the quality of molecules found under a limited oracle budget (Gao et al., 2022).

DOCKSTRING provides docking software, data, and benchmark tasks for target-dependent molecular design (García-Ortegón et al., 2022).

Molecular representations and language models. Chemical pretraining supplies representations and generative priors for downstream tasks. ChemBERTa studies self-supervised molecular representations for property prediction (Chithrananda et al., 2020); Chemformer pretrains a transformer for computational chemistry (Irwin et al., 2022); and MolGPT uses a transformer decoder for molecular generation (Bagal et al., 2021). BioT5 connects molecular and protein sequences with natural language (Pei et al., 2023), while MolCA connects molecular graphs to language models through a cross-modal projector (Liu et al., 2023).

Domain-specific language models extend these capabilities to chemical dialogue and broader scientific inputs. ChemLLM and ChemDFM incorporate chemical knowledge and instruction data (Zhang et al., 2024; Zhao et al., 2025b). ChemVLM integrates chemical text and images (Li et al., 2025); Intern-S1 combines scientific multimodal pretraining with reinforcement learning (Bai et al., 2025); and NatureLM models sequences across molecular, biological, and materials domains (Xia et al., 2025).

Chemical reasoning models further adapt how a model reasons before producing an answer. ChemDFM-R combines atomized chemical knowledge, reasoning distillation, and domain-specific reinforcement learning (Zhao et al., 2025a). The ether0 model is trained with reinforcement learning on chemistry tasks and generates both natural-language reasoning and chemical structures (Narayanan et al., 2025). These developments establish chemical representation and reasoning capabilities that motivate studying how models learn from successive molecular proposals and evaluator feedback.

GA-based search. GraphGA applies chemically valid graph mutations and crossovers, then uses oracle scores for population selection (Jensen, 2019). Its evaluated population and archive provide memory within each optimization run, while its mutation and crossover operators remain fixed. Subsequent methods improve this search interface in different ways. Parallel-tempered GA preserves diversity across concurrently evolving populations (Nigam et al., 2022); RGA pretrains and fine-tunes target-aware neural policies for parent and mutation selection, explicitly exploiting binding knowledge shared across protein targets (Fu et al., 2022); and MOLLEO uses pretrained chemistry-aware LLMs as evolutionary operators (Wang et al., 2025a). RGA and MOLLEO combine population-based selection with learned policies or pretrained molecular knowledge.

RL-based optimization. REINVENT and ReLeaSE established score-directed fine-tuning of molecular sequence generators (Olivecrona et al., 2017; Popova et al., 2018); REINVENT 4 generalizes this recipe across modern generator architectures and scoring workflows (Loeffler et al., 2024). Augmented Memory increases sample efficiency through experience replay and SMILES augmentation (Guo & Schwaller, 2023), while GFlowNet methods target diverse high-reward sets rather than only a single mode (Bengio et al., 2021; Zhu et al., 2023). More recent LLM-based methods address molecular modification under natural-language instructions. POLO trains on complete multi-turn trajectories, combining trajectory-level RL with preferences between intermediate molecules (Wang et al., 2025c). MolMem adds static exemplar retrieval, trajectory-derived skill memory, and dense step-wise policy optimization so that information can be reused across optimization runs (Wang et al., 2026). These methods use different sources of supervision and memory to improve molecular generation and reuse optimization experience.

Complementary LLM systems use retrieval, dialogue, or tools to guide optimization without making evaluator-driven policy learning the central object (Ye et al., 2025; Liu et al., 2024a; Nguyen & Grover, 2025). Instruction-tuned molecular models also target generalization across multi-property requests (Dey et al., 2025). These systems broaden the interface to molecular optimization. Our analysis examines how molecular evaluator scores guide turn-level policy learning.

Reward-guided generative fine-tuning. Related methods optimize molecular generators through diffusion trajectories. SEPO develops a policy-gradient objective for discrete diffusion under non-differentiable rewards (Zekri & Boullé, 2025). Iterative Distillation uses off-policy roll-ins, reward-guided roll-outs, and distillation, while TR2-D2 constructs replay data through tree search for off-policy discrete-diffusion fine-tuning (Su et al., 2026; Tang et al., 2025). These approaches optimize diffusion denoising processes. L2E trains an autoregressive policy from groups of evaluator-guided refinement trajectories.

Mathematical discovery. Evaluator-guided search also appears in mathematical discovery, where language models propose programs and external checks determine which candidates to retain. FunSearch evolves programs under automated evaluation (Romera-Paredes et al., 2024). This setting provides context for learning from search feedback, while the experiments here concern molecular optimization.

F Limitations and Broader Impact

F.1 LIMITATIONS

Evaluation uses computational property scores; synthesizability and biological activity require separate validation. Similarity is penalized during training and constrained at output selection, and score exploitation remains possible within these controls. The main benchmark uses one checkpoint per method, with repeats measuring stochastic decoding variability. Independent training repetitions are needed to characterize variability across training seeds.

Evaluation covers two model sizes and the tested trajectory lengths and chemical distributions. Longer trajectories, new-scaffold generalization, new evaluator families, and slow or noisy evaluators remain directions for further evaluation.

F.2 BROADER IMPACT

Learning to Evolve trains language models from evaluator feedback during molecular optimization. It aims to find better candidates within a limited proposal budget. The experiments measure computational property scores. Assessing biological activity, safety, and practical drug-discovery value requires further chemical and experimental validation of the candidate molecules.