EVOMOL: SEMANTIC GRADIENT EVOLUTION FOR RULE-GUIDED MOLECULAR REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown strong potential for molecular reasoning, but their performance depends heavily on the reasoning context, and recurring errors are rarely retained as reusable strategies. We present **EvoMol**, a semantic gradient evolution framework that adapts the reasoning logic of frozen LLMs without updating model parameters. EvoMol represents the task context as a structured logic kernel containing task decomposition strategies, molecular constraints, and mechanistic heuristics. At each generation, EvoMol diagnoses model failures and organizes them as *Actionable Side Information*, including the observed error, underlying chemical principle, corrective action, and diagnostic confidence. This structured feedback guides rule-level mutations of the logic kernel. A multi-frontier Pareto consensus mechanism then selects candidates that perform consistently across instances and objectives. Across three molecular benchmarks and Qwen2.5 models from 3B to 14B parameters, EvoMol achieves the best performance in 49 of 60 settings under limited supervision and outperforms direct prompting and GRPO in most cases. Controlled analyses show that the evolved kernels capture task-relevant strategies beyond generic or irrelevant prompt content and transfer across mechanistically related tasks. EvoMol also improves multi-objective molecular optimization under competing property constraints. These results demonstrate that EvoMol provides a data-efficient and inspectable approach to adapting molecular reasoning with frozen LLMs. Code is available at: https://anonymous.4open.science/r/EvoMol-4371

1 INTRODUCTION

Large language models (LLMs) have emerged as a general-purpose engine for molecular science, transcending simple question answering to perform complex property reasoning (Li et al., 2026b; Jang et al., 2025; Pu et al., 2026), molecular optimization (Wang & Ding, 2025; Gao et al., 2026; Zeng et al., 2026), and reaction prediction (Yuan et al., 2025; Li et al., 2026a). Benchmarks such as ChemCoTBench (Hao et al., 2026), ChemLLMBench (Guo et al., 2023), and MolInstruction (Fang et al., 2024) have standardized evaluation across these domains. However, raw capability does not equate to robust scientific reasoning (Ma et al., 2024; 2026). Current LLM-based reasoning remains highly sensitive to the strategic context-the logical instructions and mechanistic constraints supplied to the model (Agrawal et al., 2026). Even minor sub-optimalities in this context can lead to catastrophic failures, where models repeat structured molecular errors such as valence violations, incorrect ring counts, or chemically implausible reaction centers (Ouyang et al., 2023; Cavanagh et al., 2026). Critically, these failures are typically discarded as "noise" after evaluation, rather than being internalized as persistent, reusable molecular knowledge to prevent recurrence. Consequently, reasoning experience fails to accumulate, leaving the model as a static observer in a dynamic chemical logic space. Two fundamental challenges hinder the development of trustworthy molecular reasoners. (i) **The Fragility of Static Reasoning Contexts:** molecular tasks require different output structures, chemical constraints, and reasoning strategies. Static hand-crafted contexts are therefore brittle and often transfer poorly across tasks, with performance degrading as task complexity increases. (ii) **The Knowledge Internalization Gap:** parameter-level adaptation methods such as GRPO (Shao et al., 2024) can improve behavior, but often require substantial data and leave acquired knowledge implicit in model weights rather than as explicit, interpretable, and reusable reasoning principles. These limitations motivate a central question: *Can a frozen LLM improve its*

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.

molecular reasoning by transforming systematic failures into structured and reusable logic kernels without parameter updates? In this paper, we propose **EvoMol**, a semantic gradient trajectory evolution framework for molecular reasoning. Moving beyond static prompting and opaque fine-tuning, EvoMol treats the reasoning context as a structural logic kernel that undergoes autonomous evolution. Starting from a seed configuration, EvoMol iteratively navigates the reasoning strategy space by evolving its optimization trajectory while keeping the backbone LLM frozen. Each generation (i) maps molecular failure modes into a latent failure space, (ii) derives Actionable Side Information (ASI) as a semantic gradient to provide directional feedback, (iii) executes knowledge-guided rule-level mutations, and (iv) selects the optimal advancement via a two-stage Pareto-consensus mechanism. The central insight of our work is that diagnostic feedback functions as a semantic gradient in the discrete space of reasoning strategies. Rather than reducing errors to scalar penalties, EvoMol structures each failure as an actionable triple consisting of an observed failure (e), an underlying chemical rule (r), and a correction strategy (a). This structured feedback provides a gradient-like signal for updating the logic kernel. Since the backbone model remains frozen, this optimization occurs along a discrete reasoning trajectory, ensuring high data efficiency and intrinsic interpretability. Under comparable evaluation budgets, EvoMol substantially outperforms GRPO (Shao et al., 2024) across all tasks, with particularly large gains on reasoning-intensive tasks such as molecular design. The evolved trajectories further exhibit task-adaptive convergence, characterized by progressive rule discovery and stable knowledge accumulation. We evaluate EvoMol across multiple molecular benchmarks using Qwen2.5 models at 3B, 7B, and 14B scales. EvoMol consistently outperforms direct prompting and parameter-level adaptation, while learning task-specific chemical semantics that transfer across related tasks. In multi-objective molecular optimization, it further balances competing objectives, demonstrating robust and adaptive molecular reasoning.

Our contributions are summarized as follows: **(i)** We propose EvoMol, a failure-driven evolution framework that adapts frozen LLMs to molecular tasks by optimizing an explicit logic kernel, turning recurring errors into reusable and inspectable reasoning strategies without parameter updates. **(ii)** We develop structured semantic gradients that connect observed failures with chemical principles and corrective actions, guiding knowledge-informed rule mutations and Pareto-consensus selection. **(iii)** Across three molecular benchmarks, EvoMol outperforms direct prompting and GRPO in the majority of evaluated settings under low-resource adaptation budgets. Further analyses demonstrate the task specificity and selective cross-task transfer of evolved strategies, alongside improvements in multi-objective molecular optimization.

2 Related Work

LLMs for Molecular Reasoning. The integration of large language models into chemistry has expanded molecular modeling from conventional discriminative prediction toward more general molecular reasoning and generation (Ramos et al., 2025; Zhang et al., 2025; Jablonka et al., 2024; Wang et al., 2023). Early foundation models, such as Galactica (Taylor et al., 2022) and ChemBERTa (Singh et al., 2026), demonstrated that large-scale pre-training over scientific text and molecular representations can capture useful chemical knowledge. Subsequent instruction-tuned models, including Mol-Instruction (Fang et al., 2024) and ChemLLM (Zhang et al., 2024), further improved the ability of LLMs to follow chemical instructions and solve diverse molecular tasks. Despite this progress, current models remain prone to chemical hallucinations, including invalid molecular structures, violations of chemical constraints, and incorrect mechanistic reasoning. More importantly, most existing approaches treat the reasoning strategy as fixed after training or instruction tuning, limiting their ability to adapt when systematic failures emerge. EvoMol instead equips a frozen LLM with a dynamic reasoning layer that transforms recurring failures into structured, reusable chemical principles for continual strategy refinement.

Optimization of LLM Reasoning Strategies (Khattab et al., 2024). Optimizing reasoning strategies in natural language is challenging because the search space is discrete and non-differentiable. Early prompt optimization methods, such as APE (Zhou et al., 2022) and OPRO (Yang et al., 2024), formulate instruction search as a black-box optimization problem, while evolutionary approaches such as EvoPrompt (Guo et al., 2024) and ProTeGi (Pryzant et al., 2023; Ramnath et al., 2025) iteratively improve prompts through mutation, feedback, and selection. However, these methods typically rely on scalar performance signals and provide limited information about why a reasoning strategy fails. A related line of work improves reasoning through self-reflection and iterative cor-

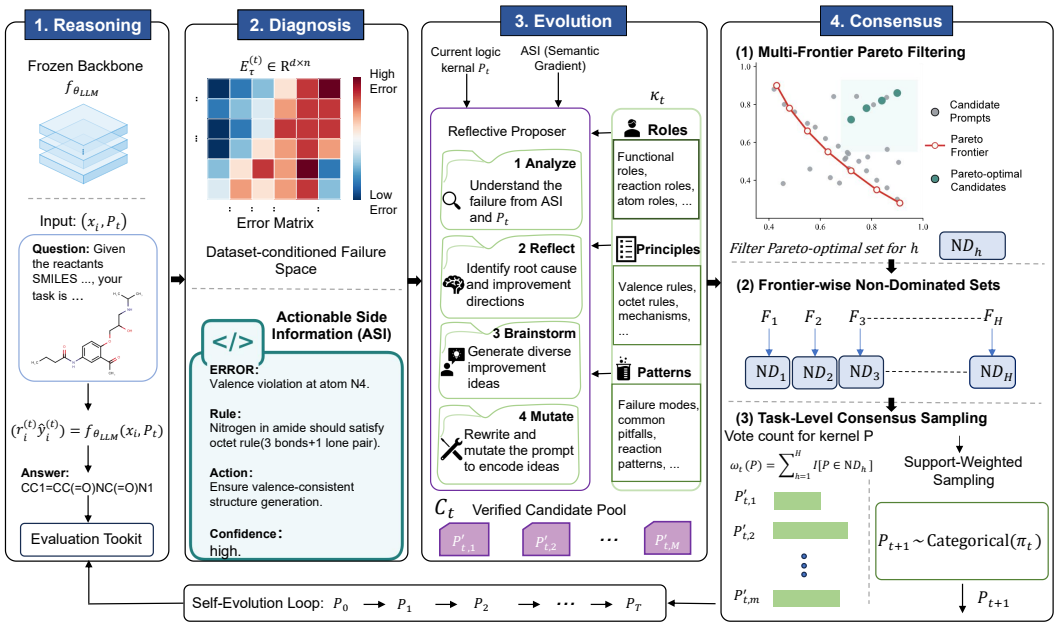


Figure 1: Overview of EvoMol. A frozen LLM reasons with kernel P_t, and diagnosed failures are converted into ASI semantic gradients. Knowledge-guided reflective mutation generates candidate kernels, followed by Pareto filtering, preferred-kernel selection, and task-level voting to obtain P_{t+1}.

rection. Methods such as Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) use model-generated feedback to revise intermediate reasoning or final answers, while systems such as CRITIC (Gou et al., 2024), GEPA (Agrawal et al., 2026), and ChemCrow (Bran et al., 2023) incorporate external tools to verify model outputs. These approaches mainly operate at the instance level, where feedback is used to improve the current episode but is not necessarily converted into reusable reasoning knowledge. Memory-augmented agents and skill libraries extend this idea by retaining previous trajectories or successful behaviors, yet knowledge accumulation is often an auxiliary consequence of execution rather than the explicit optimization objective. Another major paradigm is parameter-level adaptation. Reinforcement learning methods such as GRPO (Shao et al., 2024) optimize reasoning behavior by updating model parameters and have shown strong performance in structured reasoning domains. However, such approaches can require substantial training data and computation, while the acquired knowledge remains implicit in model weights and is difficult to interpret or transfer explicitly. EvoMol differs from these paradigms by optimizing an explicit reasoning trajectory while keeping the backbone model frozen. Instead of reducing failures to scalar rewards or transient critiques, EvoMol structures them into chemically grounded optimization signals and accumulates the resulting knowledge into an interpretable logic kernel. This enables reasoning strategies to evolve across instances while preserving data efficiency and transferability.

3 METHODOLOGY

EvoMol adapts a frozen LLM by evolving its reasoning context rather than updating its parameters. We represent this context as a structured **logic kernel** P_t, which encodes task decomposition strategies, molecular constraints, and verification rules at generation t. Starting from a seed kernel P_0, EvoMol iteratively diagnoses reasoning failures, converts them into semantic refinement signals, generates rule-level mutations, and selects robust candidates through Pareto consensus. As illustrated in Figure 1, each generation follows

$$P_t \xrightarrow{\text{Reason}} (\mathcal{R}_t, \widehat{\mathcal{Y}}_t) \xrightarrow{\text{Diagnose}} (\mathbf{E}_\tau^{(t)}, \mathcal{A}_t) \xrightarrow{\text{Evolve}} C_t \xrightarrow{\text{Consensus}} P_{t+1}, \tag{1}$$

where $\mathcal{R}_t$ and $\widehat{\mathcal{Y}}_t$ collect the reasoning trajectories and predictions, $\mathbf{E}_\tau^{(t)}$ is the error matrix for task τ, $\mathcal{A}_t$ is the resulting Actionable Side Information (ASI), and C_t is the verified candidate pool.

Notation. For task τ, let $\mathcal{D}_\tau = \{(x_i, y_i)\}_{i=1}^n$ denote its n training instances. We use $t \in \{0, \ldots, T-1\}$ to index evolution generations, d for the number of diagnostic criteria, q for the number of optimization objectives, and $H = n + q$ for the number of Pareto frontiers. The symbols m and h index candidate kernels and Pareto frontiers, respectively. The complete chemical knowledge base is denoted by $\mathcal{K}$.

3.1 Reasoning Logic Kernel

A logic kernel $P \in \mathcal{P}$ is a structured reasoning policy that specifies how the model decomposes a task, applies chemical constraints, and verifies its output. At generation t, the frozen backbone $f_{\theta_{\text{LLM}}}$ produces a reasoning trajectory $r_i^{(t)}$ and prediction $\hat{y}_i^{(t)}$:

$$\left(r_i^{(t)}, \hat{y}_i^{(t)}\right) = f_{\theta_{\text{LLM}}}(x_i; P_t), \qquad \nabla_{\theta_{\text{LLM}}} = 0. \tag{2}$$

Accordingly, $\mathcal{R}_t = \{r_i^{(t)}\}_{i=1}^n$ and $\widehat{\mathcal{Y}}_t = \{\hat{y}_i^{(t)}\}_{i=1}^n$. The utility of a logic kernel is defined as

$$U_\tau(P) = \frac{1}{n} \sum_{i=1}^{n} F_\tau(\hat{y}_i(P), y_i), \qquad P^\star = \arg\max_{P \in \mathcal{P}} U_\tau(P), \tag{3}$$

where $\hat{y}_i(P)$ is the prediction produced using kernel P, and F_τ is the task-specific evaluator. Since $\mathcal{P}$ is discrete and F_τ may involve non-differentiable chemical tools, EvoMol optimizes P through rule-level evolution rather than gradient descent.

3.2 Structured Diagnosis and Semantic Gradients

At generation t, the evaluation toolkit assesses each prediction along d diagnostic criteria, such as chemical validity, structural consistency, and target-property satisfaction. The normalized failure severities are collected into

$$\mathbf{E}_\tau^{(t)} = \left[\sigma_{j,i}^{(t)}\right]_{d \times n}, \qquad \sigma_{j,i}^{(t)} \in [0, 1], \tag{4}$$

where $j \in \{1, \ldots, d\}$ indexes diagnostic criteria and $\sigma_{j,i}^{(t)}$ measures the severity of criterion j on instance x_i. Larger values indicate more severe failures. The diagnostic evidence is distilled into

$$\text{ASI}_i^{(t)} = \left\langle e_i^{(t)}, \rho_i^{(t)}, a_i^{(t)}, \gamma_i^{(t)} \right\rangle, \tag{5}$$

where $e_i^{(t)}$ describes the observed error, $\rho_i^{(t)}$ identifies the underlying chemical or reasoning principle, $a_i^{(t)}$ provides an actionable correction, and $\gamma_i^{(t)} \in [0, 1]$ denotes diagnostic confidence. The complete ASI set,

$$\mathcal{A}_t = \left\{\text{ASI}_i^{(t)}\right\}_{i=1}^n, \tag{6}$$

serves as a **Semantic Gradient**: instead of providing only a scalar penalty, it specifies what failed, why it failed, and how the current logic kernel should be revised.

3.3 Knowledge-Guided Reflective Evolution

EvoMol maintains a chemical knowledge base

$$\mathcal{K} = \left\{\mathcal{K}^{\text{role}}, \mathcal{K}^{\text{principle}}, \mathcal{K}^{\text{pattern}}\right\}, \tag{7}$$

covering molecular roles, general chemical principles, and recurring reaction or failure patterns. Based on $\mathcal{A}_t$, EvoMol retrieves a relevant subset $\mathcal{K}_t \subseteq \mathcal{K}$. Given P_t, $\mathcal{A}_t$, and $\mathcal{K}_t$, the Reflective Proposer generates candidate mutations:

$$P'_{t,m} \sim \pi_{\text{evolve}}\left(P_t, \mathcal{A}_t, \mathcal{K}_t\right), \tag{8}$$

where m indexes the generated candidates. The proposer follows four operations: Analyze the diagnosed failures, Reflect on their chemical causes, Brainstorm possible corrections, and Mutate the relevant rules in P_t. Each mutation is first compared with P_t on a verification mini-batch $\mathcal{B}_t \subseteq \mathcal{D}_\tau$. The candidates that improve the estimated mini-batch utility form

$$C_t = \left\{P'_{t,m} \;\middle|\; \widehat{U}_{\mathcal{B}_t}(P'_{t,m}) > \widehat{U}_{\mathcal{B}_t}(P_t)\right\}, \tag{9}$$

where $\widehat{U}_{\mathcal{B}_t}$ denotes utility estimated on $\mathcal{B}_t$. If $C_t = \varnothing$, EvoMol retains the current kernel by setting $P_{t+1} = P_t$.

3.4 MULTI-FRONTIER PARETO CONSENSUS

Average utility may favor rules that improve only a subset of molecules or objectives. EvoMol therefore evaluates each verified candidate through a **Multi-Frontier Pareto Consensus** mechanism.

Let $g_o(P, x_i)$ denote the performance of kernel P on objective $o \in \{1, \ldots, q\}$ for instance x_i, with all metrics oriented such that higher values are better. For each instance and objective, we construct

$$\mathbf{z}_i^{\text{inst}}(P) = [g_1(P, x_i), \ldots, g_q(P, x_i)]^\top, \qquad \mathbf{z}_o^{\text{obj}}(P) = [g_o(P, x_1), \ldots, g_o(P, x_n)]^\top. \tag{10}$$

The n instance-level vectors and q objective-level vectors define $H = n+q$ complementary frontiers.

Let $\mathbf{z}_h(P)$ denote the score vector of candidate P on frontier h. Candidate P dominates Q, denoted $P \succ_h Q$, if it is no worse in every component of $\mathbf{z}_h$ and strictly better in at least one. The corresponding non-dominated set is

$$\text{ND}_h = \{P \in C_t \mid \nexists Q \in C_t \text{ such that } Q \succ_h P\}. \tag{11}$$

The consensus support of candidate P is

$$\omega_t(P) = \sum_{h=1}^{H} \mathbb{I}[P \in \text{ND}_h], \tag{12}$$

which counts the number of frontiers on which P is Pareto-optimal. We normalize this support into a sampling probability:

$$\pi_t(P) = \frac{\omega_t(P)}{\sum_{Q \in C_t} \omega_t(Q)}. \tag{13}$$

The next-generation kernel is then sampled as

$$P_{t+1} \sim \text{Categorical}\left(\{\pi_t(P)\}_{P \in C_t}\right). \tag{14}$$

This mechanism favors candidates that remain robust across molecules and objectives while retaining exploration among valid alternatives.

4 EXPERIMENTS

Our experimental evaluation examines the effectiveness, interpretability, and transferability of EvoMol through the following research questions: **RQ1 (Low-resource Adaptation):** Can EvoMol effectively adapt frozen LLMs to diverse molecular reasoning tasks under limited supervision, and how does it compare with parameter-level optimization such as GRPO under matched evaluation budgets? **RQ2 (Rule Specificity):** Do the evolved logic kernels capture task-relevant molecular reasoning strategies rather than benefiting from generic instructions, irrelevant content, or superficial prompt augmentation? **RQ3 (Cross-task Transfer):** Can reasoning strategies evolved for one molecular task transfer to related tasks, and how does their transferability depend on the molecular operations shared across task families? **RQ4 (Competing Objectives):** Can EvoMol effectively evolve reasoning strategies for molecular optimization problems involving multiple and potentially competing property objectives?

4.1 EXPERIMENTAL SETUP

Benchmarks and Evaluation Metrics. We evaluate **EvoMol** across three complementary molecular reasoning benchmarks: (i) **ChemCoTBench** (Hao et al., 2026), which comprises 19 subtasks spanning molecular optimization, editing, understanding, and reaction prediction; (ii) **ChemLLMBench** (Guo et al., 2023), which includes 15 tasks covering molecular generation, reaction reasoning, and property prediction; and (iii) **MolInstruction** (Fang et al., 2024), which contains 17 instruction-following subtasks represented primarily using SELFIES. We employ task-specific metrics that reflect the distinct requirements of these tasks. Structural agreement is measured using Tanimoto similarity computed from Morgan fingerprints, while molecular property optimization is evaluated using Therapeutics Data Commons (TDC) (Huang et al., 2021; 2022) oracle scores, including QED, DRD2, and GSK3B. For molecular counting tasks, such as `fg_count` and `ring_count`, we

Table 1: Performance comparison across three molecular reasoning benchmarks. The best result in each model–task setting is highlighted in light blue and shown in **bold**. Lower values are better for counting tasks (↓); higher values are better otherwise (↑).

Task	Qwen2.5-3B			Qwen2.5-7B			Qwen2.5-14B		
	Direct	GRPO	**EvoMol**	Direct	GRPO	**EvoMol**	Direct	GRPO	**EvoMol**
ChemCoTBench									
Add	0.067	**0.150**	0.117	0.083	**0.150**	0.133	0.200	**0.350**	0.333
Delete	0.133	0.200	**0.217**	0.367	0.250	**0.433**	**0.600**	0.550	0.583
Substitute	0.017	0.050	**0.072**	0.117	**0.133**	0.122	0.272	0.283	**0.361**
DRD2	−0.034	−0.011	**0.011**	−0.021	−0.013	**0.036**	−0.040	−0.017	**0.001**
GSK3B	−0.028	−0.005	**0.003**	−0.032	−0.010	**0.001**	−0.019	**0.008**	−0.006
Murcko scaffold	0.082	**0.485**	0.484	0.088	0.097	**0.114**	0.068	0.131	**0.146**
FG count ↓	1.680	1.690	**0.900**	0.723	0.970	**0.497**	1.055	1.089	**0.591**
Ring count ↓	1.450	1.050	**0.467**	1.050	1.050	**0.450**	1.050	1.050	**0.400**
RCR	0.007	0.010	**0.077**	0.071	**0.092**	0.083	0.069	0.069	**0.113**
ChemLLMBench									
Captioning	0.023	0.028	**0.088**	0.026	0.037	**0.073**	0.000	0.034	**0.063**
Design	0.005	0.001	**0.014**	0.033	0.064	**0.070**	0.101	0.090	**0.108**
IUPAC-to-SMILES	0.004	0.007	**0.036**	0.014	0.025	**0.061**	0.074	**0.080**	0.064
Retrosynthesis	0.051	0.072	**0.087**	0.038	0.071	**0.142**	0.000	0.134	**0.166**
SMILES-to-IUPAC	0.225	0.233	**0.283**	0.267	0.255	**0.274**	0.271	0.289	**0.305**
MolInstruction									
Captioning	0.011	0.017	**0.027**	0.005	0.021	**0.342**	0.009	0.013	**0.369**
Disease extraction	0.011	0.013	**0.262**	0.010	0.011	**0.343**	0.009	0.014	**0.336**
Chemical NER	0.007	0.035	**0.332**	0.167	0.200	**0.382**	0.406	**0.453**	0.411
Molecular design	0.001	0.001	**0.352**	0.071	0.024	**0.438**	0.124	0.058	**0.304**
Forward reaction	0.149	0.384	**0.613**	0.638	0.503	**0.641**	**0.705**	0.528	0.684
Retrosynthesis	0.183	0.146	**0.790**	0.741	0.529	**0.774**	0.758	0.647	**0.796**

Note: Direct denotes zero-shot inference without task-specific adaptation. Results are averaged over three random seeds.

report Mean Absolute Error (MAE). Molecular editing is assessed using RDKit-based correctness rates that verify structural validity and task-specific constraints. For property prediction, we report classification accuracy or relative regression error, depending on the task. Detailed task definitions and metric formulations are provided in Appendix B.

Data Partitioning and Baselines. To assess low-resource adaptation, we partition each task into 15 training instances and 85 held-out evaluation instances. The training set is used exclusively for strategy evolution or parameter adaptation, while the held-out set is reserved for final evaluation. This deliberately data-scarce setting tests whether a model can extract task-relevant and reusable reasoning strategies from minimal supervision. We compare **EvoMol** with two representative baselines: (1) **Direct Prompting**, which performs zero-shot inference without task-specific adaptation; and (2) **GRPO** (Guo et al., 2025), a parameter-level reinforcement learning baseline trained on the same 15 instances. For GRPO, we optimize task-specific LoRA adapters with rank 32 under 4-bit quantization for two epochs.

Budget-Matched Comparison. To compare optimization efficiency under a controlled low-resource setting, we constrain both methods to a comparable task-level feedback budget of approximately $O(10^2)$ evaluator calls per task. For **EvoMol**, the total budget is 100 calls, consisting of 75 evaluations across five generations and 15 training instances, together with 25 additional calls for candidate verification ($5 \times 15 + 25 = 100$). For **GRPO**, we use 120 reward-evaluated rollouts, obtained from 15 training instances over two epochs with four responses sampled per instance ($15 \times 2 \times 4 = 120$). This setup enables a controlled comparison between explicit reasoning-trajectory evolution and parameter-level reinforcement learning under comparable amounts of task-specific evaluative feedback.

Implementation Details. We evolve **EvoMol** for $T = 5$ generations. The **Reflective Proposer** is instantiated using DeepSeek-V4 via API with a temperature of 0.7, generating rule-level mutations conditioned on the actionable triples in Eq. **??** and the chemical knowledge base $\mathcal{K}$. Local model inference is served through Ollama on a workstation equipped with an Intel Core i9-14900K processor, 128 GB of RAM, and two NVIDIA GeForce RTX 4090D GPUs with 24 GB of VRAM each. The GRPO baseline is implemented using Unsloth[1] under 4-bit quantization. Specifically, we optimize task-specific LoRA adapters with rank $r = 32$ and scaling factor $\alpha = 64$, targeting all linear layers. Training is performed for two epochs with a batch size of 4, an initial learning rate of 5×10^{-5}, and a cosine learning-rate scheduler. During optimization, four responses are sampled per prompt for reward computation. All experiments are repeated with three random seeds, and we report the mean performance. Additional software and hardware configurations are in Appendix C.

4.2 Main Results (RQ1)

Table 1 summarizes the low-resource adaptation results across three benchmarks. EvoMol achieves the best performance in 49 of the 60 reported model–task settings, with consistent advantages across all three model scales. The largest gains occur on tasks requiring explicit structural reasoning, molecular transformation, or precise instruction following.

On **ChemCoTBench**, EvoMol effectively corrects recurrent optimization and structural errors. For example, the DRD2 improvement score on Qwen2.5-14B increases from -0.040 with direct prompting to $+0.001$, reversing an otherwise detrimental optimization trajectory. On Qwen2.5-3B, EvoMol reduces the ring-counting MAE from 1.450 to 0.467, compared with 1.050 for GRPO. On **ChemLLMBench**, improvements are generally more moderate: for instance, the SMILES-to-IUPAC score on Qwen2.5-14B increases from 0.271 to 0.305. This smaller margin suggests that reasoning-strategy evolution offers less additional benefit when performance primarily depends on knowledge already encoded in the model parameters. The most pronounced gains are observed on **MolInstruction**. On Qwen2.5-7B, the molecule-captioning score increases from 0.005 to 0.342, while EvoMol achieves 0.790 on Qwen2.5-3B retrosynthesis, compared with 0.183 for direct prompting and 0.146 for GRPO. Overall, these results answer **RQ1** affirmatively: under comparable evaluation budgets, EvoMol provides effective low-resource adaptation, particularly when recurring failures can be converted into explicit reasoning and verification strategies.

4.3 Rule Specificity and Cross-task Transfer (RQ2 & RQ3)

To address **RQ2** and **RQ3**, we compare the full EvoMol kernel with four variants: Seed, Nonsense, Generic, and Cross-task. These controls examine whether the observed gains arise from task-specific reasoning strategies rather than additional prompt content alone, and whether the evolved strategies transfer across related tasks. Detailed definitions of the variants are provided in Appendix D.4.

For **RQ2**, the Full kernel consistently outperforms the Seed, Generic, and None variants across the settings shown in Fig. 2. On Qwen2.5-3B forward reaction prediction, Full achieves 0.6125, compared with 0.1787 for Generic and 0.1128 for None; similarly, on Qwen2.5-7B molecule captioning, Full reaches 0.3420, whereas None (0.0052) remains close to Seed (0.0049). These results indicate that the improvements depend on task-specific strategies induced from diagnostic feedback rather than generic chemical instructions or the initial reasoning context alone. The qualitative case study in Appendix D.6 further illustrates this process: on a representative molecular design example, the kernel progressively acquires strategies for SELFIES validity, structural feature extraction, chemical-class discrimination, and scaffold recovery. Accordingly, molecular similarity increases from 0.462 at P_2 to 0.917 at P_5 and reaches 1.000 at P_{10}, while validation fitness improves from 0.0011 to 0.6346. Together, the quantitative and qualitative results suggest that EvoMol converts recurring failures into increasingly specific and effective molecular reasoning strategies.

For **RQ3**, the Cross-task variant shows that transfer is selective and depends on the compatibility between source and target tasks. For Qwen2.5-14B retrosynthesis, the Cross-task kernel achieves 0.7614, approaching the task-specific Full kernel at 0.7955. On Qwen2.5-3B reagent prediction, transferred rules even outperform the Full kernel (0.2956 versus 0.2073), suggesting substantial overlap between forward-reaction and reagent-reasoning strategies. In contrast, transfer is limited

[1] https://unsloth.ai/

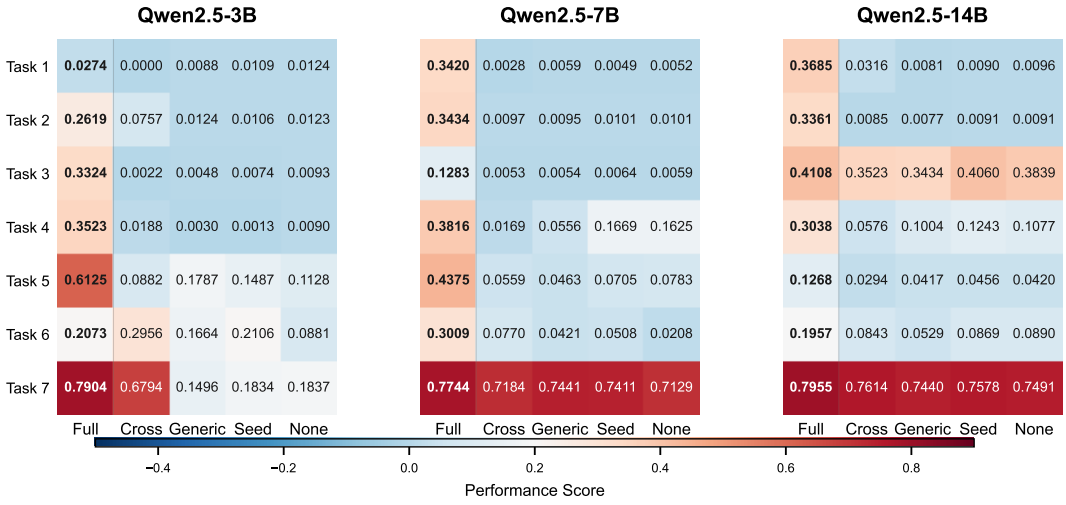


Figure 2: Ablation of evolved logic kernels on MolInstruction. **Full** uses task-specific evolved rules; **Cross** transfers rules from a related task; **Generic** uses general chemical principles; **Seed** denotes the initial logic kernel; and **None** uses nonsense context. Higher scores indicate better performance. Task-ID mappings are provided in Appendix A.2.

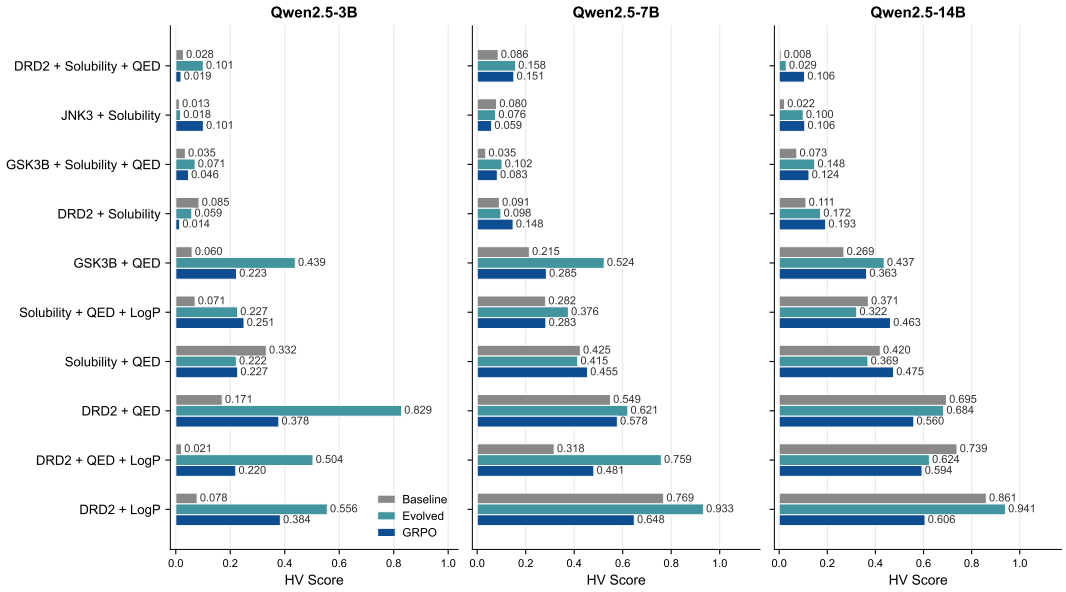


Figure 3: Multi-objective optimization performance (HV scores) across Qwen2.5 scales for Baseline, EvoMol, and GRPO across ten complex tasks.

for tasks such as molecule captioning, whose requirements differ substantially from those of the source tasks. Together, these findings answer **RQ2** and **RQ3**: EvoMol learns task-relevant reasoning strategies beyond superficial prompt artifacts, while their transferability is strongest between molecular tasks sharing compatible reasoning structures.

4.4 Multi-Objective Molecular Optimization (RQ4)

To address **RQ4**, we evaluate EvoMol on ten multi-objective molecular optimization tasks, including seven bi-objective and three tri-objective settings. Performance is measured using Hypervolume (HV), which jointly captures solution quality and coverage of the objective space (Fig. 3; see Appendix D.7 for details). EvoMol achieves the highest HV scores on 7 of the 10 objective com-

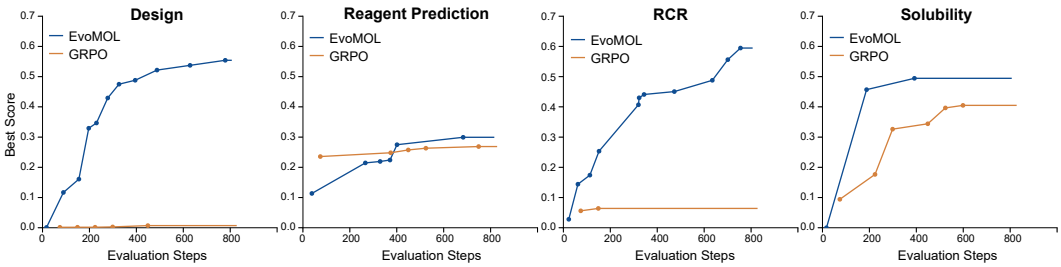


Figure 4: Best-score trajectories of EvoMol (blue) and GRPO (orange) over approximately 800 evaluation steps. EvoMol exhibits gradual improvement on Molecular Design, stepwise gains on Reagent Prediction and RCR, and rapid early convergence on Solubility. Higher scores are better.

binations, with particularly clear gains on the smaller 3B and 7B backbones. For GSK3B + QED, it consistently outperforms both direct prompting and GRPO across all model scales; on Qwen2.5-7B, its HV reaches 0.5236, compared with 0.2151 for direct prompting. EvoMol also improves the Qwen2.5-3B score on the tri-objective DRD2 + QED + LogP task from 0.0209 to 0.5037. The gains are larger when objectives impose competing structural demands and more modest when properties are comparatively aligned, such as Solubility + QED. This pattern suggests that semantic gradients help convert property conflicts into actionable strategies, while Pareto-consensus favors kernels that remain effective across objectives. Overall, EvoMol provides effective low-resource adaptation for complex multi-property optimization, answering **RQ4** affirmatively.

4.5 Evolutionary Dynamics and Convergence Patterns

In a separate extended-budget analysis, we track the best performance achieved by EvoMol and GRPO over approximately 800 evaluations across four representative tasks (Fig. 4). EvoMol generally improves faster and reaches higher performance plateaus, with particularly large gains on RCR (0.595 vs. 0.064) and Molecular Design (0.554 vs. 0.007). The trajectories reveal three task-dependent convergence patterns: (i) **fast alignment** on Solubility, where performance stabilizes within approximately 200 evaluations; (ii) **stepwise discovery** on RCR and Reagent Prediction, characterized by abrupt improvements consistent with the acquisition of discrete reasoning strategies; and (iii) **gradual accumulation** on Molecular Design, where performance improves steadily as complementary heuristics are incorporated. In contrast, GRPO exhibits flatter trajectories on the more structurally constrained tasks, suggesting that scalar rewards provide limited directional guidance under the same evaluation budget. Overall, these results show that semantic-gradient evolution supports adaptive search across heterogeneous molecular reasoning landscapes. Additional analyses are provided in Appendix D.8.

5 Conclusion

We introduced EvoMol, a semantic gradient evolution framework that converts recurring molecular failures into explicit reasoning rules for adapting frozen LLMs without parameter updates. EvoMol combines structured failure diagnosis, knowledge-guided rule mutation, and Pareto consensus selection to progressively refine task-specific logic kernels from limited supervision. Across three molecular benchmarks, EvoMol achieves effective low-resource adaptation, transfers learned strategies between mechanistically related tasks, and improves optimization under competing molecular objectives. Controlled analyses further show that these gains arise from task-relevant chemical strategies rather than generic instructions or increased prompt length. A key direction for future work is to consolidate the evolved strategies into a curated and reusable molecular rule library. Each rule will document its applicable tasks, chemical constraints, empirical support, confidence, and relationships to other rules. Such a resource would allow reasoning strategies to be retrieved, combined, verified, and updated across tasks, extending EvoMol from task-specific adaptation toward continual molecular knowledge evolution. We plan to release this library to support transparent knowledge reuse and reproducible research in molecular reasoning and design.

AI USE STATEMENT

Generative AI was used in this work in two distinct roles. First, DeepSeek-V4 served as the Reflective Proposer in EvoMol, where it analyzed diagnostic feedback and proposed rule-level mutations of the logic kernel. This usage constitutes an experimental component of the proposed method and is described in the methodology and implementation details.

Second, large language model assistants, including ChatGPT/Codex, were used to support conceptual brainstorming, refinement of the methodology and experimental presentation, code drafting and debugging, organization and interpretation of empirical results, literature discovery, language editing, and the formatting of figures and tables. Generative AI was not used to create benchmark data or labels, formulate mathematical proofs, fabricate or modify experimental measurements, or independently determine the conclusions of the study.

All AI-assisted outputs were reviewed by the authors. Generated code was inspected and tested against the task definitions, numerical results were independently checked, and citations, chemical claims, figures, and manuscript text were manually verified. All final decisions concerning the research design, methodology, analysis, and presentation were made by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

This study is entirely computational and does not involve human participants, personal data, clinical interventions, or animal experiments. All benchmarks and molecular evaluation tools are publicly available and used in accordance with their respective licenses. Although EvoMol may support molecular design and optimization, its outputs are intended solely for research purposes and should not be interpreted as experimentally or clinically validated compounds. The authors acknowledge the potential dual-use risks of generative molecular systems and encourage appropriate expert review and experimental validation before any real-world application.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the complete implementation of EvoMol and all experimental scripts as anonymous supplementary material. We also release representative intermediate reasoning trajectories, including failure diagnoses, actionable side information, logic-kernel mutations, and selection outcomes. The main text specifies the proposed algorithm and evaluation protocol, while the appendix documents the datasets, task partitions, evaluation metrics, prompts, hyperparameters, and computational environment. All reported experiments are repeated over three random seeds using the configurations described in the paper.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

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

Joseph M Cavanagh, Kunyang Sun, Andrew Gritsevskiy, Dorian Bagni, Yingze Wang, Thomas D Bannister, and Teresa Head-Gordon. Smileyllama: Modifying large language models for directed chemical space exploration. *Nature Computational Science*, pp. 1–14, 2026.

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

Wenhan Gao, Xiran Fan, Chin-Chia Michael Yeh, Jiarui Sun, Yuzhong Chen, Menghai Pan, Mahashweta Das, and Yi Liu. Feedback to reasoning: Llm-assisted molecular optimization with domain feedback and historical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12735–12754, 2026.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

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

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 34133–34156, 2024.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in neural information processing systems*, 36:59662–59688, 2023.

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

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Artificial intelligence foundation for therapeutic science. *Nature Chemical Biology*, 2022.

Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

Yunhui Jang, Jaehyung Kim, and Sungsoo Ahn. Structural reasoning improves molecular understanding of llm. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21016–21036, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Saiful Haq, Ashutosh Sharma, Thomas Joshi, Hanna Moazam, Heather Miller, et al. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, volume 2024, pp. 54928–54958, 2024.

Jiawei Li, Minzhou Li, Qi Yang, and Sanzhong Luo. Reactionseek: Llm-powered literature data mining and knowledge discovery in organic synthesis. *Nature Communications*, 17(1):3356, 2026a.

Xuan Li, Zongze Li, Jiangchao Yao, Yu Rong, Lu Zhang, Bo Han, et al. Reference-guided policy optimization for molecular optimization via llm reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 65266–65297, 2026b.

Tengfei Ma, Xuan Lin, Tianle Li, Chaoyi Li, Long Chen, Peng Zhou, Xibao Cai, Xinyu Yang, Daojian Zeng, Dongsheng Cao, et al. Y-mol: A multiscale biomedical knowledge-guided large language model for drug development. *arXiv preprint arXiv:2410.11550*, 2024.

Tengfei Ma, Yuqin He, Zhonghao Ren, Bosheng Song, Qian Li, and Xiangxiang Zeng. Expanding knowledge boundaries via llm-grounded alignment for drug combination recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 11633–11644, 2026.

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. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Siru Ouyang, Zhuosheng Zhang, Bing Yan, Xuan Liu, Yejin Choi, Jiawei Han, and Lianhui Qin. Structured chemistry reasoning with large language models. *arXiv preprint arXiv:2311.09656*, 2023.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

Zhili Pu, Lantian Zhang, Hao Duan, Zhixing Zhang, Keyun Zhu, Yongqi Fan, Ruihui Hou, Tong Ruan, and Yun Tang. Mcle-mol: Empowering llm with molecular comprehension and low-cost continual evolution for interpretable property prediction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5305–5333, 2026.

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, et al. A systematic survey of automatic prompt optimization techniques. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33066–33098, 2025.

Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *Chemical science*, 16(6):2514–2572, 2025.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Riya Singh, Aryan Amit Barsainyan, Rida Irfan, Connor Joseph Amorin, Stewart He, Tony Davis, Arun Thiagarajan, Shiva Sankaran, Seyone Chithrananda, Walid Ahmad, et al. Chemberta-3: an open source training framework for chemical foundation models. *Digital Discovery*, 5(2): 662–685, 2026.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Ziqing Wang and Kaize Ding. Remol: Llm-guided molecular optimization with reinforcement learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (PhD Consortium)*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Shen Yuan, Shukai Gong, and Hongteng Xu. Uspto-llm: A large language model-assisted information-enriched chemical reaction dataset. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 817–820, 2025.

Daojian Zeng, Tianle Li, Jiahao Yang, Jiacai Yi, Xieping Gao, Lincheng Jiang, Tengfei Ma, and Xiangxiang Zeng. Expert-inspired multi-agent coordination for multi-objective molecular optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34575–34583, 2026.

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.

Qiang Zhang, Keyan Ding, Tianwen Lv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, et al. Scientific large language models: A survey on biological & chemical domains. *ACM Computing Surveys*, 57(6):1–38, 2025.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

A Appendix

A.1 EvoMol Algorithm and Detailed Summary

Self-Evolution Algorithm. Algorithm 1 formalizes the iterative trajectory update process of EvoMol. Throughout the optimization process, the foundation model parameters θ_{LLM} remain completely untouched ($\nabla_{\theta_{\text{LLM}}} = 0$), ensuring that all reasoning improvements are derived strictly from the evolutionary refinement of the logic kernel in the discrete strategy space.

Algorithm 1 EvoMol Self-Evolution Procedure

Require: Training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, seed kernel P_0, chemical knowledge base $\mathcal{K}$, evaluation trials T, objective criteria count O.

Initialize candidate pool $C_0 \leftarrow \{P_0\}$, initialize support weight $w(P_0) \leftarrow 1$
for $t = 0, \ldots, T-1$ **do**
  Parent Sampling: Sample parent $P_{\text{parent}} \sim \text{Categorical}\left(\left\{\frac{w(P)}{\sum_{P' \in C_t} w(P')}\right\}_{P \in C_t}\right)$ (Eq. 14)
  Diagnosis: Profile failures into $\mathbf{E}_k^{(t)}$ (Eq. 4) and derive semantic gradients $\text{ASI}_i^{(t)} = \langle e_i^{(t)}, r_i^{(t)}, a_i^{(t)} \rangle$ (Eq. **??**)
  Reflective Mutation: Propose single candidate $P' \leftarrow \pi_{\text{evolve}}(P_{\text{parent}}, \{\text{ASI}_i^{(t)}\}, \mathcal{K})$
  Mini-Batch Verification: Evaluate P' against P_{parent} on diagnostic subset $\mathcal{D}_{\text{batch}} \subset \mathcal{D}$
  if $f(P', \mathcal{D}_{\text{batch}}) > f(P_{\text{parent}}, \mathcal{D}_{\text{batch}})$ **then** ▷ Passes admission filter
    $C_{t+1} \leftarrow C_t \cup \{P'\}$ ▷ Admit into candidate pool
    Update non-dominated sets $\mathcal{ND}_k$ across $K = N + O$ frontiers over C_{t+1} (Eq. 11)
    Recompute support weights $w(P) = \sum_{k=1}^{K} \mathbb{I}[P \in \mathcal{ND}_k]$ for all $P \in C_{t+1}$ (Eq. 12)
  else
    $C_{t+1} \leftarrow C_t$ ▷ Early rejection to save evaluation budget
  end if
end for
return Optimal logic kernel $P^* = \arg\max_{P \in C_T} w(P)$

Conceptual Responsibility Matrix. The EvoMol framework cleanly separates responsibilities across the entire self-evolution pipeline:

- **Structured Diagnosis:** Identifies and profiles systematic failure patterns in the error matrix $\mathbf{E}_k^{(t)}$;
- **Semantic Gradient (ASI):** Formalizes directional feedback into actionable triples $\langle e, r, a \rangle$ (error, chemical rule, and correction action);
- **Chemical Knowledge Base ($\mathcal{K}$):** Supplies domain principles, mechanistic rules, and common reaction pitfalls as inductive search constraints;
- **Reflective Proposer (π_{evolve}):** Executes four-step meta-cognitive mutations ($\text{Analyze} \rightarrow \text{Reflect} \rightarrow \text{Brainstorm} \rightarrow \text{Mutate}$) to propose candidate rules;

- **Mini-Batch Gating:** Protects population quality and evaluation efficiency via early rejection of unverified mutations;
- **Multi-Frontier Pareto Filtering:** Preserves diverse non-dominated trade-offs across N instances and O conflicting objectives ($\mathcal{ND}_k$);
- **Consensus Sampling:** Balances exploitation of broadly robust rules with exploratory stochasticity via support-weighted categorical selection.

A.2 Task ID Mapping for Benchmarks

To ensure the interpretability of the results presented in the main text and supplementary figures, we provide a definitive mapping of Task IDs (Task 1–7) to their respective subtasks across all model scales in Table 2.

Table 2: Mapping of Task IDs to specific molecular reasoning subtasks across three model scales for the scientific validation and mechanistic studies.

ID	Qwen2.5-3B	Qwen2.5-7B	Qwen2.5-14B
Task 1	molecule captioning	molecule captioning	molecule captioning
Task 2	chem_disease_extract	chem_disease_extract	chem_disease_extract
Task 3	chemical_ner	chem_protein	chemical_ner
Task 4	molecular design	chemical_ner	molecular design
Task 5	forward reaction	molecular design	open question answering
Task 6	reagent prediction	reagent prediction	reagent prediction
Task 7	retrosynthesis	retrosynthesis	retrosynthesis

B Experimental Setup Details

B.1 Subtask Taxonomy and Metrics

EvoMol is evaluated on a total of 51 subtasks. The metrics used for each task are defined as follows:

- **Tanimoto Similarity:** For molecular and reaction generation, we calculate $T(a,b) = \frac{N_c}{N_a+N_b-N_c}$ based on Morgan fingerprints (radius 2, 2048 bits).
- **TDC Oracles:** Specifically, QED measures drug-likeness, DRD2 targets dopamine receptor D2, and gsk targets glycogen synthase kinase 3.
- **Structural Correctness:** In molecular editing and understanding, we use RDKit to verify if the output SMILES/SELFIES are valid and meet task-specific structural criteria.
- **Text-based Metrics:** For captioning and text extraction, we employ BLEU-4 and Soft-score (normalized Levenshtein similarity).

C Implementation and Hyperparameters

C.1 GRPO Training Configurations

The GRPO baseline is fine-tuned using the Unsloth library with 4-bit quantization.

- **LoRA Parameters:** Rank $r = 32$, Alpha $\alpha = 64$, targeting all linear layers.
- **Optimization:** Learning rate 5×10^{-5} with a cosine scheduler; 2 epochs; Batch size 4.
- **Sampling:** 4 responses sampled per prompt for reward calculation during optimization.

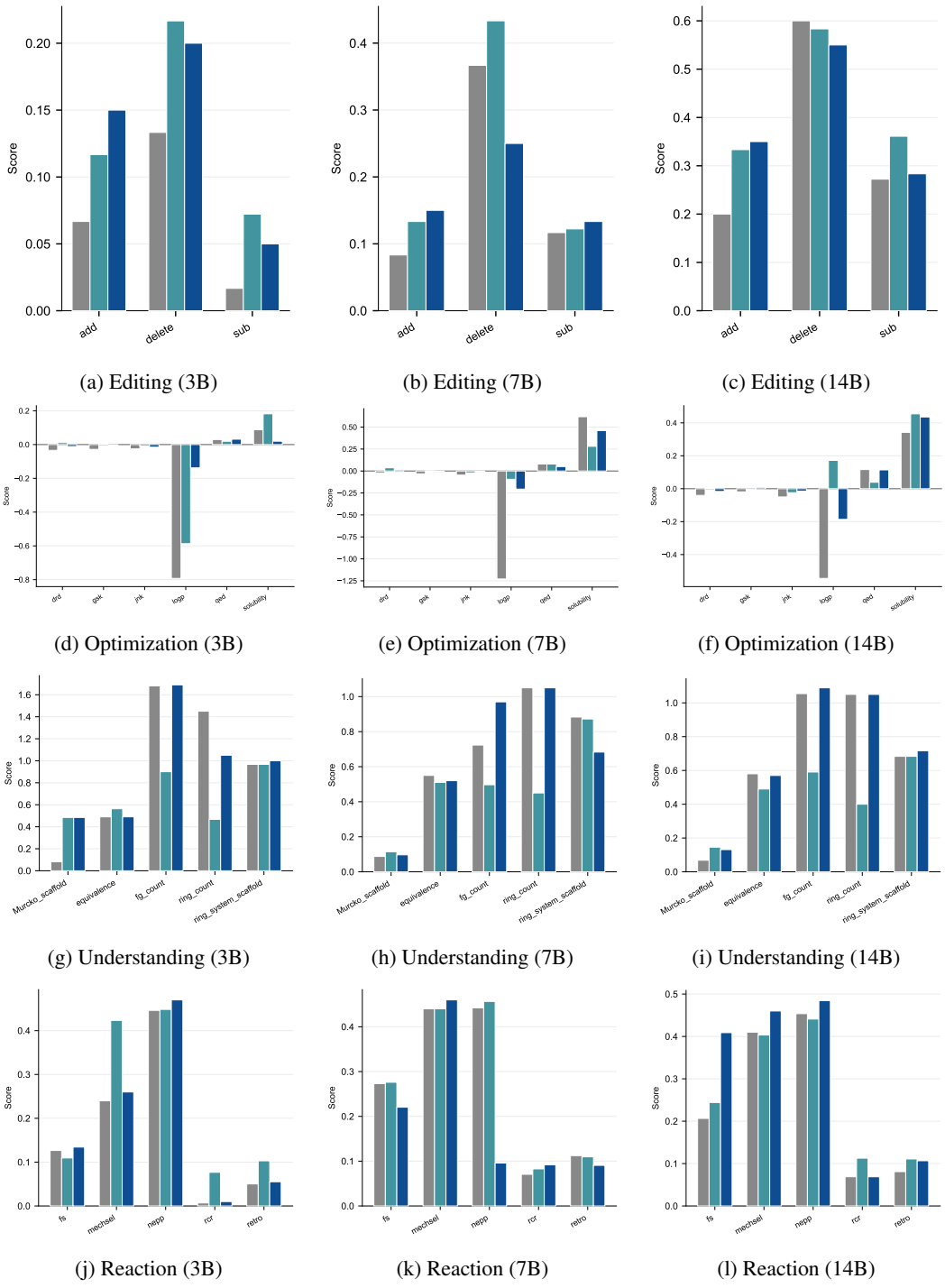
C.2 Compute Infrastructure

All local model inferences and fine-tuning were performed on a workstation equipped with an Intel Core i9-14900K, 128GB RAM, and a single NVIDIA RTX 4090D GPU (24GB VRAM). DeepSeek-V4 was accessed via API with a temperature of 0.7.

D Complete Experimental Results

We provide the complete performance breakdown for all benchmarks and model scales. In the following figures, we compare **EvoMol** against the **Baseline** (Direct Prompting) and **GRPO**.

D.1 ChemCoTBench: Comprehensive Results



(a) Editing (3B) (b) Editing (7B) (c) Editing (14B)

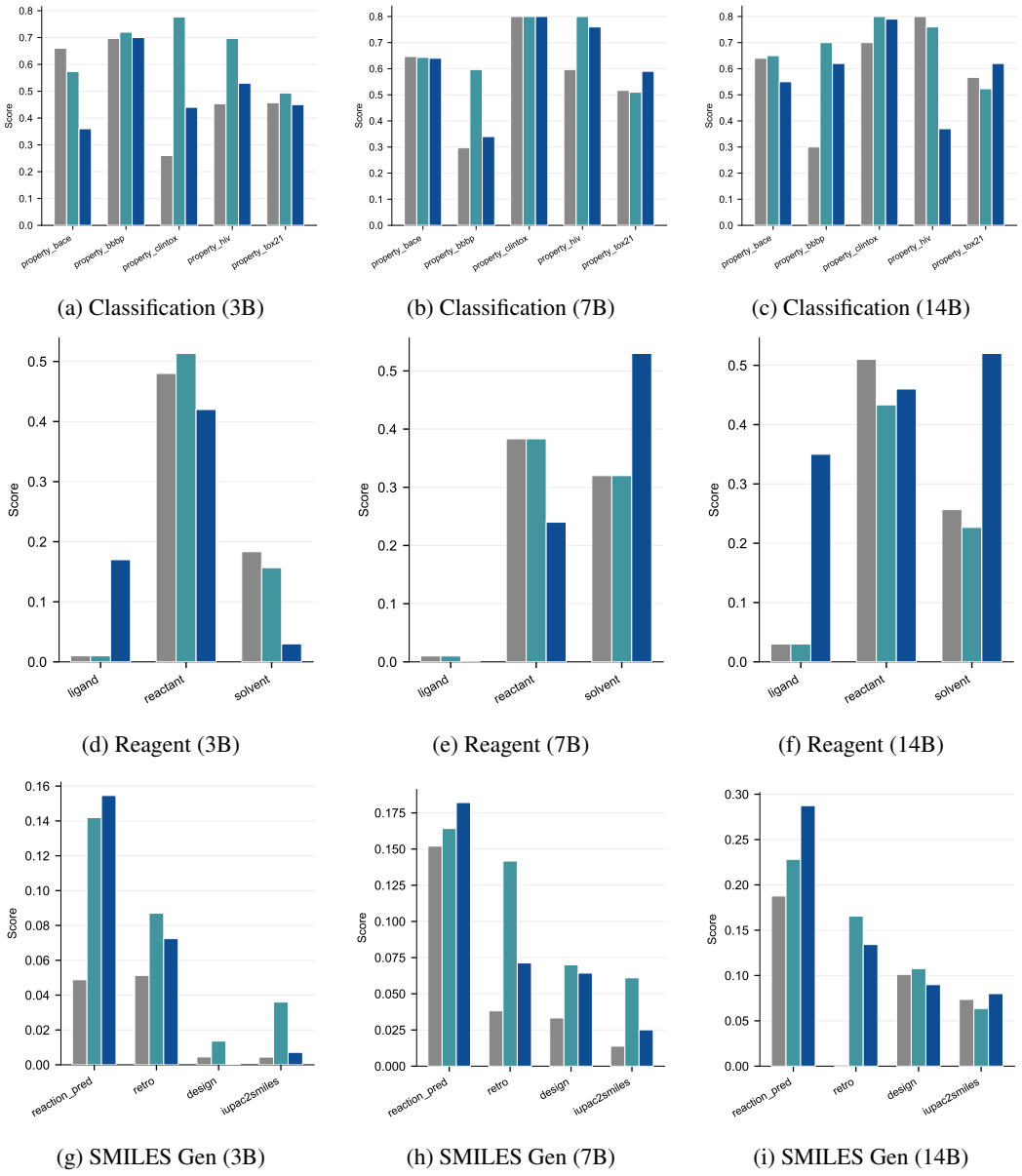
(d) Optimization (3B) (e) Optimization (7B) (f) Optimization (14B)

(g) Understanding (3B) (h) Understanding (7B) (i) Understanding (14B)

(j) Reaction (3B) (k) Reaction (7B) (l) Reaction (14B)

Figure 5: Complete results on **ChemCoTBench** across four task categories and three model scales.

D.2 ChemLLMBench: Comprehensive Results



(a) Classification (3B) (b) Classification (7B) (c) Classification (14B)

(d) Reagent (3B) (e) Reagent (7B) (f) Reagent (14B)

(g) SMILES Gen (3B) (h) SMILES Gen (7B) (i) SMILES Gen (14B)

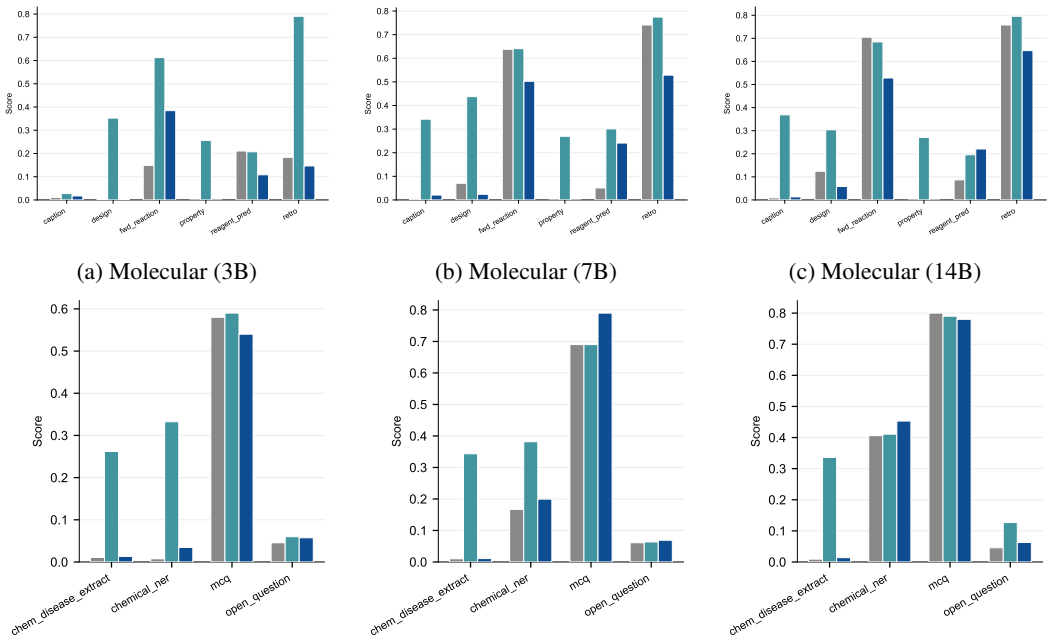
Figure 6: Complete results on **ChemLLMBench** (Selected Categories). EvoMol consistently shows superior performance in property classification and reagent selection.

D.3 MolInstruction: Comprehensive Results

D.4 Scientific Validation: Comprehensive Ablation Analysis

To isolate the semantic utility of the logic kernel and ensure that performance gains are not driven by superficial artifacts, we evaluate five logic kernel variants across all model scales and benchmarks:

- **Seed**: The baseline performance using only the original task instruction without any evolved rules.



(a) Molecular (3B) (b) Molecular (7B) (c) Molecular (14B)

(d) Text & QA (3B) (e) Text & QA (7B) (f) Text & QA (14B)

Figure 7: Complete results on **MolInstruction**. EvoMol exhibits high gains in complex structured reasoning and QA tasks.

- **+Generic rules**: Augments the seed with four universal chemistry principles (e.g., "Obey the octet rule," "Conserve mass in reactions"). These are scientifically correct but not task-specific.
- **+Nonsense rules**: Augments the seed with four semantically well-formed but chemically irrelevant sentences (e.g., "The sky is blue"). This serves as a control for prompt length, formatting, and attention-triggering effects.
- **+Cross-task rules**: Augments the seed with rules extracted from the evolved logic kernels of other tasks within the same benchmark family.
- **Full EvoMol**: Uses the task-specific best evolved logic kernel, representing the performance upper bound of our framework.

D.5 Detailed Results and Heatmaps

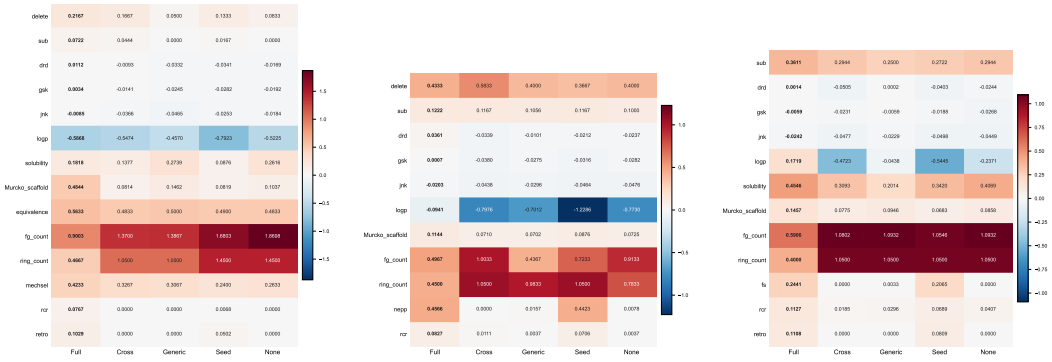
The comprehensive results for all 15 subtasks across three benchmarks are visualized in Figures 8, 9, and 10.

D.5.1 ChemCoTBench Analysis

As shown in Figure 8, EvoMol provides decisive improvements in structural counting and property optimization.

- **Counting Accuracy:** In tasks like *fg_count* and *ring_count*, the Full kernel significantly reduces MAE across all scales. For instance, on Qwen2.5-3B, the *ring_count* MAE drops from 1.450 (Seed) to 0.467 (Full). Notably, the **+Nonsense** variant results in a performance collapse back to 1.450, proving that counting logic is tied to semantic content rather than prompt length.
- **Property Recovery:** In *logp* and *drd* optimization, the Seed often results in extreme negative scores (e.g., 7B *logp*: -1.229). The Full kernel successfully "corrects" the optimization

trajectory, bringing the score toward zero or positive values (-0.094), a feat not achieved by the **+Generic** variant.



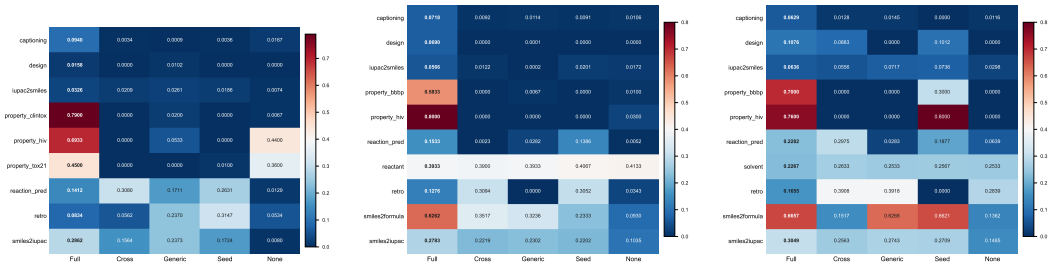
(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Qwen2.5-14B

Figure 8: Rule ablation on ChemCoTBench. Higher values indicate better performance (except for counting tasks where lower is better).

D.5.2 ChemLLMBench Analysis

Figure 9 highlights the emergence of "all-or-nothing" reasoning capabilities in property prediction.

- **Emergent Reasoning:** On tasks such as *property_hiv* and *property_clintox*, the Seed and +Generic variants often yield scores of 0.000. The Full kernel enables the model to bypass this failure, achieving scores up to 0.790 (3B) and 0.800 (7B/14B). This suggests that task-specific heuristics are the primary drivers of success in these binary classification landscapes.
- **Scaling Independence:** Interestingly, the Qwen2.5-3B model with a Full kernel (0.693 on *property_hiv*) significantly outperforms the Qwen2.5-14B Seed baseline (0.800 is high but see *property_tox21* where 3B Full 0.450 vs 14B Seed 0.010). This confirms that explicit logic evolution can compensate for smaller parameter scales.



(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Qwen2.5-14B

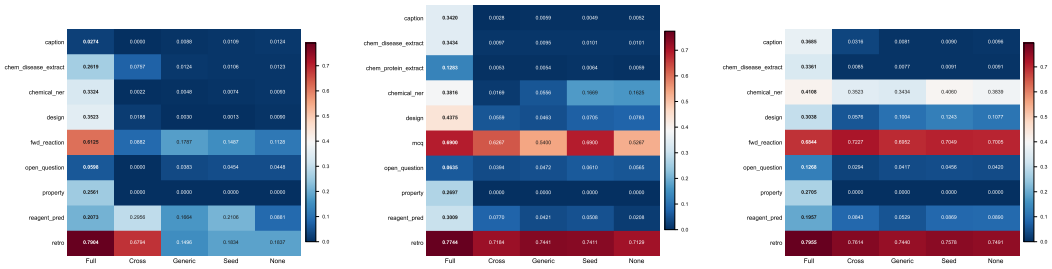
Figure 9: Rule ablation on ChemLLMBench. Note the significant leap in property prediction accuracy with the Full kernel.

D.5.3 MolInstruction Analysis

The results in Figure 10 validate the transferability of reasoning primitives.

- **Primitive Sharing:** In the forward reaction (*fwd_reaction*) and retrosynthesis (*retro*) tasks, the **+Cross-task** variant consistently approaches the performance of the Full kernel. For example, on 3B *retro*, +Cross achieves 0.679 compared to Full's 0.790, both far exceeding the Seed's 0.183. This indicates that EvoMol distills underlying mechanistic logic that is shared between symmetrical chemical tasks.

- **Generative Robustness:** In complex generative tasks like *design* and *caption*, the Full kernel provides a leap in performance (e.g., 7B *design*: 0.070 to 0.438). The failure of the **+Nonsense** and **+Generic** variants to match this gain reinforces that task-specific strategic logic is essential for high-fidelity molecular generation.



(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Qwen2.5-14B

Figure 10: Rule ablation on MolInstruction. Transferability between reaction-related tasks is highly evident in the +Cross-task variant.

D.6 Mechanistic Analysis of Logic Induction Trajectory

This section provide a granular examination of the optimization trajectory of the **logic kernel** within EvoMol. By tracing the evolution of a representative Molecular Design task, we demonstrate how the **Semantic Gradient** (∇_{semantic}) iteratively contracts the model's failure space. Figure 11 visualizes this progression across four pivotal generations.

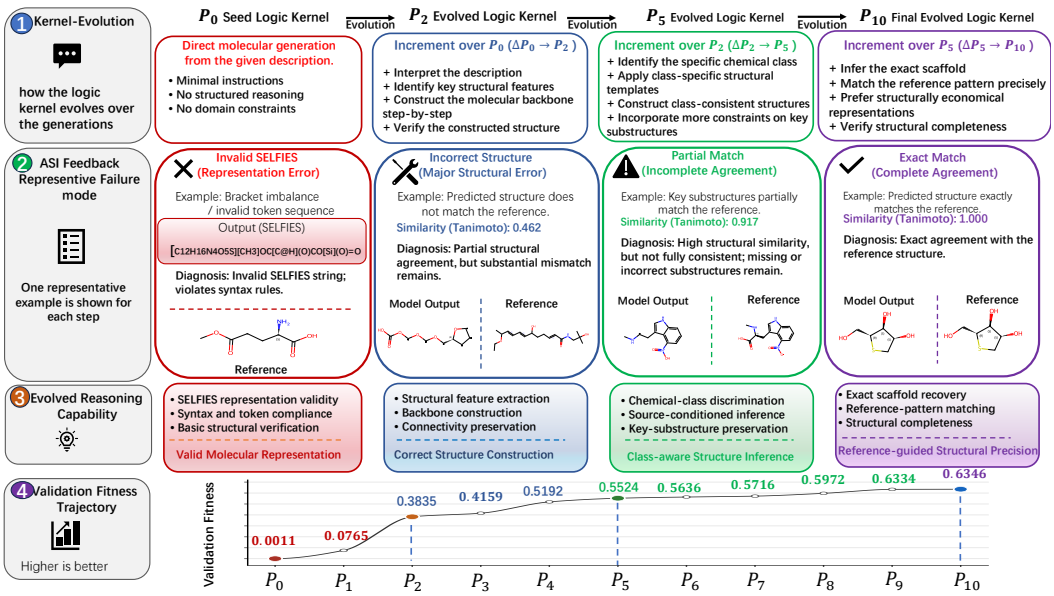


Figure 11: Longitudinal case study of logic kernel evolution. The figure tracks the co-evolution of reasoning prompts and behavior across four tracks: ① Kernel-Evolution (ΔP); ② ASI Feedback (representative failures); ③ Evolved Reasoning Capability; and ④ Validation Fitness Trajectory.

Stage 0 to 1: From Syntax Alignment to Structured Reasoning. Initially, the seed kernel (P_0) lacks domain constraints (Track ①), resulting in near-zero scores and frequent representation errors (Track ②). By **Stage 1** (P_2), the kernel transitions to a **structured reasoning** protocol. The model learns to interpret descriptions and construct the backbone step-by-step (Track ③), leading to a performance leap to 0.3835 (Track ④).

Stage 2: Domain Specialization and Topological Awareness. In **Stage 2 (P_5)**, the logic kernel addresses "topological blindness" through class-specific templates (Track ①). Track ② identifies failures in generic scaffolds. In response, EvoMol induces rules for chemical-class discrimination (Track ③), pushing fitness to 0.5524 (Track ④).

Stage 3: Reference-aware Convergence and Structural Precision. The final transition to **Stage 3 (P_{10})** represents mature **diagnostic induction**. The kernel internalizes reference-aware heuristics (Track ①). By using the **Pareto-consensus mechanism** as an Occam's razor, EvoMol filters out unnecessary substituents. As evidenced in Track ②, the model eventually achieves an **Exact Match** with the reference (Similarity = 1.000).

D.7 Multi-Objective Molecular Optimization

Task Composition. We construct ten multi-objective tasks by combining objectives from Chem-CoTBench: DRD inhibition, gsk inhibition, JNK3 inhibition, QED, LogP (target range 1.5–3.5), and aqueous solubility. Each task utilizes 50 source molecules.

Evolutionary Configuration and Metrics. EvoMol executes 8 generations per task to accommodate the expanded search space. The fitness function is the arithmetic mean of normalized property scores $[0, 1]$. For LogP, we apply target-distance normalization where the optimal range scores 1.0, decaying linearly with distance. Performance is quantified using Hypervolume (HV) with the origin $(0, \ldots, 0)$ as the reference point, capturing the model's ability to navigate the Pareto frontier.

D.8 Detailed Analysis of Evolutionary Trajectories

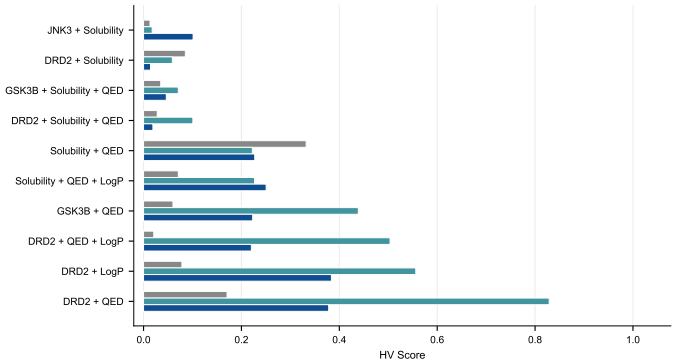
As visualized in Fig. 4, the evolutionary trajectories of EvoMol and GRPO reveal fundamental differences in how they navigate complex chemical strategy spaces under identical evaluation budgets:

Molecular Design (Generative Landscape). This task exhibits the most substantial performance gap. GRPO's best score remains near zero throughout the 800 evaluations. This is because molecular design requires satisfying a complex intersection of combinatorial constraints: structural validity, scaffold preservation, and target property improvement simultaneously. Parameter-level updates in GRPO struggle to discover the discrete "AND" logic required to satisfy these multi-faceted demands. In contrast, EvoMol demonstrates a steady upward trajectory, indicating that the Reflective Proposer successfully distills and accumulates additive heuristics (e.g., "preserve the aromatic core while iteratively modifying peripheral side chains").

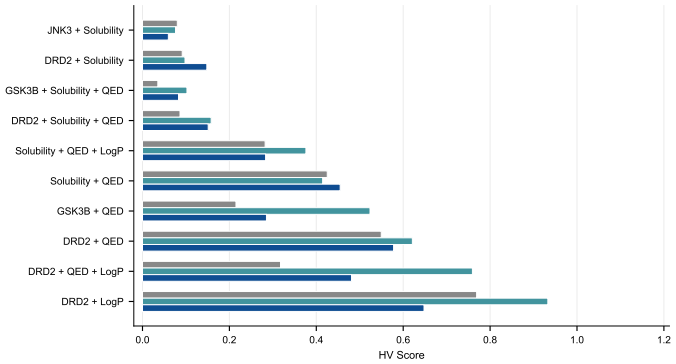
RCR and Reagent Prediction (Knowledge-Intensive Discovery). These tasks follow the distinctive "stair-step" pattern mentioned in Sec. 4.5. For instance, in Catalyst Prediction (RCR), EvoMol remains on a plateau from index 400 to 600 before a decisive jump. This period represents the "Reflective" phase where the model identifies persistent failure patterns in the error matrix $\mathbf{E}_k^{(t)}$—such as recurring valence errors or incorrect functional group compatibility—but requires multiple evolutionary trials to search for and induce a precise **reasoning primitive**. Once this primitive (e.g., a specific catalytic-mechanistic rule) is discovered and integrated into the logic kernel, it generalizes across the held-out set, causing an abrupt performance leap.

Solubility (Format and Constraint Alignment). Solubility optimization represents a relatively simpler landscape where the primary bottleneck is valid SMILES output and basic property correlation. While both methods show improvement, EvoMol reaches a significantly higher plateau much faster ($\sim$200 evaluations). Analysis of the intermediate kernels shows that EvoMol explicitly internalizes formatting rules (e.g., "ensure all ring closures in SMILES are numerically balanced") as discrete instructions within the first two generations. GRPO, however, requires significantly more samples to implicitly align its output probability distribution with the required chemical syntax.

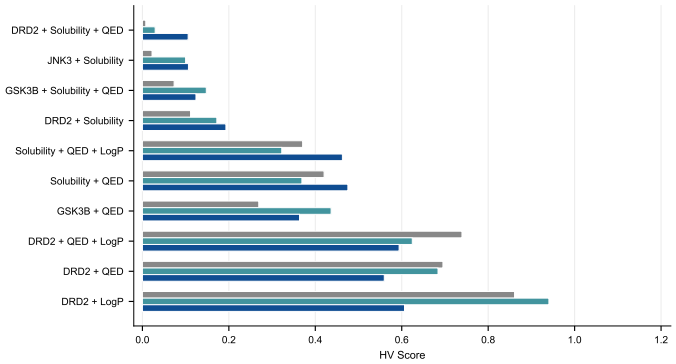
Efficiency and Mechanistic Summary. Across all task families, EvoMol's "Best Score" at index 200 often exceeds GRPO's score at index 800. This $4\times$ sample efficiency advantage stems from the fundamental difference between policy gradients and semantic evolution. The **Semantic Gradient**



(a) Qwen2.5-3B



(b) Qwen2.5-7B



(c) Qwen2.5-14B

Figure 12: Multi-objective molecular optimization performance across scales.

allows the system to ”explain” errors through chemical reasoning rather than just ”penalizing” them via scalar rewards. This explanatory power enables EvoMol to bypass the high-variance stochastic search characteristic of parameter-level RL in discrete, non-differentiable strategy spaces.

E COMPARED WITH AUTONOMOUS RESEARCH AGENTS

Recent systems such as AutoResearch and The AI Scientist automate broader research workflows by allowing language-model agents to propose experiments, modify executable code, evaluate outcomes and iteratively retain successful changes. These systems primarily optimize research artifacts, including training programs, experimental configurations and manuscripts. EvoMol addresses a different level of adaptation: it keeps both the experimental environment and backbone LLM fixed and evolves the reasoning logic used to solve molecular tasks. Rather than relying solely on scalar experimental outcomes, EvoMol structures recurring molecular failures into actionable semantic gradients and accumulates the resulting corrections in an explicit logic kernel. Thus, autonomous research agents and EvoMol are complementary: the former automate the research process, whereas the latter provides a domain-specific mechanism for adapting scientific reasoning within that process.