From Static to Agentic Reasoning for Chemical Reaction Prediction

Anonymous ACL submission

Abstract

We present **Reaction-Mind**, a framework that evolves static chemical reaction reasoning into agentic reasoning through progressively enriched learning paradigms. It comprises three learning paradigms of increasing capability. The first paradigm combines supervised fine-tuning with reinforcement learning to improve static reasoning using reaction-aware reward modeling. The second paradigm introduces grounded reasoning by incorporating structured outputs from external chemical systems as intermediate evidence. The third paradigm extends to autonomous agentic reasoning, where the model learns to interact with tools and optimize full reasoning-action trajectories via reinforcement learning, enabling dynamic tool selection and verification-based self-reflection. On forward chemical reaction prediction task, **Reaction-Mind** outperforms general LLMs and many specialized reaction prediction systems. The results demonstrate that integrating LLM with tool-grounded interaction enables better and more reliable chemical reasoning.

1 Introduction

Chemical reaction prediction is a fundamental problem in computational chemistry, with applications in synthesis planning, drug discovery, and mechanistic analysis. The task typically includes forward reaction prediction, which infers products from given reactants, and retrosynthesis prediction, which identifies plausible reactant sets for a target molecule. Despite decades of research, accurate and generalizable prediction remains challenging due to the complexity and sparsity of the chemical reaction space (Yang and Liu, 2004).

Early approaches primarily rely on template-based or template-free learning methods. Template-based methods depend on predefined reaction rules (Chen and Jung, 2022; Sacha et al., 2021; Yan et al., 2022), while template-free approaches employ graph neural networks and sequence models to directly learn mappings between reactants and products (Schwaller et al., 2018; Irwin et al., 2022; Lin et al., 2022). Although these methods achieve strong predictive performance, they operate as largely black-box systems and lack explicit reasoning capabilities, limiting their ability to generalize across unseen reaction types.

Recently, large language models (LLMs) have demonstrated promising reasoning abilities in scientific domains, including chemistry (Narayanan et al., 2026). A growing body of work explores applying LLMs to reaction prediction through chain-of-thought prompting, supervised fine-tuning, and reinforcement learning with outcome-based supervision (Wang et al.; Zhao et al., 2024a,b). These methods improve interpretability and performance by introducing explicit reasoning traces. However, most existing approaches rely on static reasoning paradigms with fixed trajectories, where the model passively generates reasoning steps without interaction or adaptation during inference.

In contrast, human chemists perform reaction reasoning in an interactive and iterative manner. They actively retrieve external knowledge, evaluate multiple hypotheses, refine predictions based on intermediate evidence, and verify feasibility through iterative reasoning cycles (Smith, 2023; Levy, 2017). Such behavior requires not only domain knowledge but also the ability to dynamically select strategies, incorporate external feedback, and correct reasoning trajectories.

Motivated by this gap, we argue that chemical reasoning should evolve from static trajectory-based generation toward agentic reasoning with progressively stronger interaction and skill acquisition capabilities. However, existing chemistry-oriented LLM systems either focus on retrieval augmentation or static reasoning ability, and do not systematically study how reasoning behaviors can emerge through staged learning and reinforcement-driven interaction (Astaraki et al., 2026).

In this work, we propose **Reaction-Mind**, a unified framework that evolves static chemical reaction reasoning into agentic reasoning through progressively enriched learning paradigms. Specifically, we design a three-stage learning framework spanning static reasoning enhancement, grounded reasoning with external agent-generated evidence, and autonomous agentic reasoning via tool interaction. This progressive design enables the model to gradually transition from passive reasoning to interactive decision-making. The main contributions of our work can be concluded in the following parts.

- We propose a multi-stage learning framework that evolves reward-enhanced static reasoning into agentic reasoning for chemical reaction prediction, and demonstrate consistent performance improvements on forward chemical reaction prediction task.
- We introduce a grounded reasoning paradigm that integrates external chemical analysis systems as structured evidence for reasoning.
- We further develop an autonomous agentic learning framework that learns effective skills for flexible tool use and verification-based self-reflection through reasoning-action optimization with outcome-based rewards.
- The experiment results confirm that our method outperforms all compared LLMs and even exceeds traditional task-specific models, in accuracy (Exact Match), Validity, and fingerprint similarity performance (FTS).

2 Method

As illustrated in Figure 1, we systematically investigate the evolutionary paradigm shifting from static text-based reasoning to fully autonomous agentic reasoning for reaction prediction. We use Qwen3-32B (Yang et al., 2025) as the base model throughout all main experiments.

Given a chemical input x, the objective is to predict the target chemical outcome y. For forward prediction, x comprises the SMILES of reactants, solvent, and reagents. To bridge the gap between passive generation and dynamic exploration, we formulate three progressively advanced paradigms.

Paradigm 1: Static Reasoning Enhancement. This stage complies with the canonical chain-of-thought (CoT) formulation, where the model performs inference entirely within a self-contained, text-only reasoning trajectory:

$$y = f_\theta(x) \tag{1}$$

We enrich this by incorporating structural retrieval databases and optimizing the reasoning policy via Group Relative Policy Optimization (GRPO) with reaction-space-aware reward mechanism.

Paradigm 2: Grounded Agentic Reasoning. To mitigate hallucination in reasoning, we introduce an intermediate paradigm where the LLM conditions its predictions on multi-source observations o generated by external analytical systems $\mathcal{A}$:

$$y = f_\theta(x, o), \quad \text{where } o = \mathcal{A}(x) \tag{2}$$

Then o is fed into the LLM as textual context. It encapsulates dynamically generated structured information such as reasoning traces, candidate pools, and categorical constraints.

Paradigm 3: Autonomous Agentic Reasoning. We finally transition to a fully interactive policy where the model dynamically interleaves internal reasoning, tool invocation, and verification-based self-reflection during inference:

$$a_t \sim \pi_\theta(\cdot|s_t) \tag{3}$$

where s_t denotes the current execution state at step t, and a_t represents the LLM action (e.g., tool call or verification-based self-reflection) which yields a subsequent environment observation $o_t \sim \mathcal{P}(\cdot|s_t, a_t)$. Through outcome-supervised optimization over reasoning-action trajectories, the LLM gradually acquires reusable agent skills to flexibly navigate the complex chemical space.

2.1 Static Reasoning Enhancement

We mainly refer to existing research (Wang et al.). Additional details are provided in **Appendix A.2.1**, **A.2.2** and **A.2.3**.

2.1.1 Retrieval-Augmented Context

Reaction Type Classifier We train a reaction type classifier to provide LLM with coarse-grained chemical priors.

Similar-Case Retrieval Databases We construct a type-conditioned chemical exemplar library utilizing the training split of the Open Reaction Database (ORD) (Kearnes et al., 2021).

2.1.2 Reasoning Trajectory Construction

For forward reaction prediction, we adopt existing methods and open source datasets (Wang et al.).

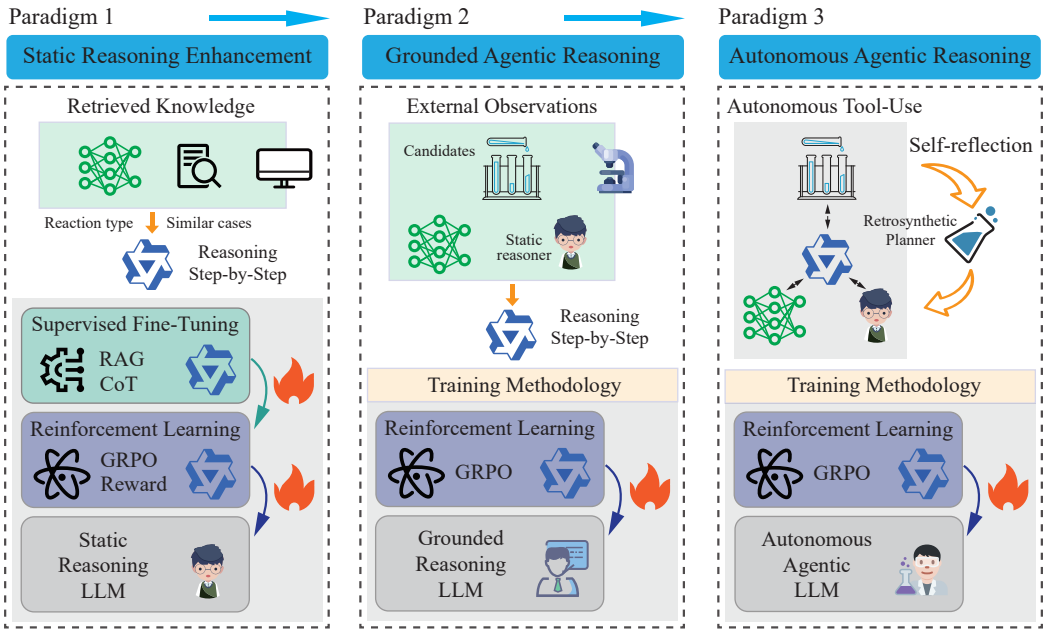

Figure 1: An overview of the proposed **Reaction-Mind**, where the reasoning paradigm transitions from static reasoning to fully autonomous agentic reasoning.

2.1.3 Reaction-Space-Aware Optimization

Algorithmic Framework Building upon the SFT initialization, we deploy Group Relative Policy Optimization (GRPO) to optimize reasoning policy π_θ with LoRA (Hu et al., 2022). Details about GRPO are provided in **Appendix A.2.4**.

Multi-Objective Reward Formulation Here we formulate a composite reward function to enforce chemical rigor and structural validity:

- **Format Reward** (R_{fmt})**:** Evaluates adherence to the predefined text template, awarding 0.1 for strict compliance, and 0 otherwise.
- **Length Reward** (R_{len})**:** Encourages concise reasoning by awarding 0.1 if the CoT falls within 1000 to 3000 tokens, and 0 otherwise.
- **Validity Reward** (R_{val})**:** Validates whether the generated output can be successfully parsed into valid chemical structures by RDKit, awarding 0.1 for success, and 0 otherwise.
- **Accuracy Reward** (R_{acc})**:** Compares the canonicalized predicted SMILES with the ground truth, awarding a dominant value of 2.0 for an exact match, and 0 otherwise.
- **Feasibility Reward** (R_{feas})**:** Verifies the chemical viability via a retrosynthesis planning model Retro-star (Chen et al., 2020). Award 1.0 for chemically plausible pathways.

Batch-Adaptive Reward Scheduling Existing work (Wang et al.) shows that chemical reaction outcomes may not be unique, multiple distinct products can be equally valid under different mechanistic pathways. This necessitates a reward design that guides the LLM beyond memorizing a single ground-truth answer. R_{feas} prioritizes exploring the reaction space by rewarding chemically plausible products. R_{acc} instead prioritizes exploitation, steering the LLM toward the single expected answer to achieve high scores on standard metrics.

Balancing the two rewards is important for reasoning optimization. Initially, we implement a heuristic two-phase schedule split at a manual milestone step K. During the early exploration phase (step $\leq K$), a higher static weight is assigned to R_{feas} to encourage diverse chemical path exploration, even if the outputs deviate from the ground truth. In the subsequent convergence phase (step $> K$), this weight is manually throttled to force the model toward precise prediction.

However, manually tuning K and static weights requires extensive trial-and-error. Furthermore, applying a uniform weight across samples fails to accommodate the varying chemical difficulty within each training batch.

To overcome the challenge, we propose a fully dynamic, batch-adaptive reward weighting scheme. Its schematic diagram is shown in Figure 2. Additional details are provided in **Appendix A.2.5**.

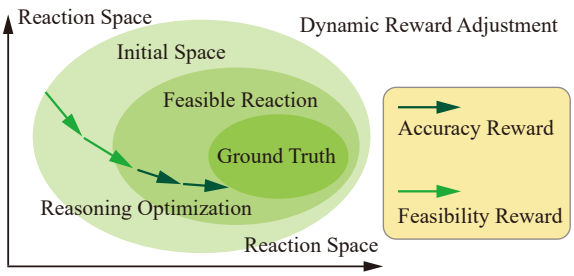

Figure 2: Schematic diagram of the dynamic reward adjustment scheme.

2.2 Grounded Agentic Reasoning

2.2.1 Agent-Grounded Context Construction

To bridge static reasoning and fully autonomous agentic reasoning, we introduce a grounded reasoning paradigm in which the LLM learns from structured observations generated by multiple external systems, denoted as $o = \mathcal{A}(x)$. The grounded observations are produced by three components:

- **Static Reasoning LLM** (o^{reason})**:** A trained model that generates both initial reasoning trajectories and preliminary reaction predictions, providing explanatory chemical analysis and candidate outcomes. Its training process is detailed in Section 2.1 **Static Reasoning Enhancement**.

- **Chemistry-Specific Predictor** (o^{cand})**:** A task-specific model providing the Top-5 high-confidence candidates. In this work, we use ReactionT5-V2 (Sagawa and Kojima, 2023).

- **Reaction Type Classifier** (o^{type})**:** A classifier providing the Top-1 most probable reaction mechanism classes, with its architecture and training detailed in Section 2.1.1 **Retrieval-Augmented Context**.

These feedbacks are serialized into a unified observation context $o = \{o^{reason}, o^{cand}, o^{type}\}$. The target LLM then learns a grounded reasoning policy $y = f_\theta(x, o)$, optimizing its internal decision-making by cross-examining its internal weights against external evidence.

2.2.2 Grounded Reasoning Optimization

We optimize this grounded policy using GRPO. In contrast to the multi-component reward design in the static stage, this phase adopts a simplified outcome-based reward R_{acc}, encouraging the model to learn effective utilization of external evidence under sparse supervision. This training objective induces two complementary behaviors.

- **Evidence Selection:** Evaluating and selecting the correct product from the external candidate pool o^{cand}, conditioned on consistency with the overall reasoning context.

- **Evidence Verification:** Critically assessing the reliability of heterogeneous external signals o^{reason}, and explicitly verifying the prediction produced by the Static Reasoning LLM. The model incorporates this verification process into its intermediate reasoning trajectory, where correctness judgments are incorporated as a process reward signal for policy optimization in GRPO.

This optimization framework provides empirical evidence for the effectiveness of incorporating heterogeneous external signals into LLM reasoning. The results demonstrate that the model can reliably perform evidence selection and verification under structured grounding. This motivates the transition to the more challenging setting of autonomous agentic reasoning with direct tool interaction.

2.3 Autonomous Agentic Reasoning

2.3.1 Action Space and Structured Skills

Building upon the methodological insights gained from the grounded reasoning analysis, which confirmed that LLMs can effectively synthesize and rectify external analytical signals, we now advance to the final evolution stage. Here, **Reaction-Mind** transitions into a fully autonomous agent that dynamically interacts with the chemical environment.

We formulate the interaction as a sequential decision process where the agent samples actions from its policy $a_t \sim \pi_\theta(\cdot|s_t)$ over an extended horizon. The available tool set incorporates four key actions: (1) Call Static Reasoner; (2) Query Chemistry-Specific Predictor; (3) Identify Reaction Type; (4) Invoke Verification.

Here we introduce a new tool **Retrosynthetic Planner** (o^{retro}) to execute invoke verification. We use Retro-star (Chen et al., 2020) to verify whether the predicted product corresponds to chemically plausible pathways, thereby ensuring the synthetic feasibility of the proposed reactions. Specifically, during self-reflection, the LLM evaluates the predicted product by checking whether a retrosynthetic route exists given the reaction inputs. If

found, the prediction is deemed chemically reasonable and thus highly reliable.

To improve the interaction efficiency of the agents, we initialize the policy by introducing two prompt-guided thinking paths, representing expert-designed skills (shown in Figure 3). Skill-A (**Candidate-Driven**): Top-5 candidates + static reasoning → reaction type filtering → candidate-reasoning-type triple consistency. Skill-B (**Type-Guided**): Reaction type + static reasoning → Top-5 candidates → type-mechanism consistency.

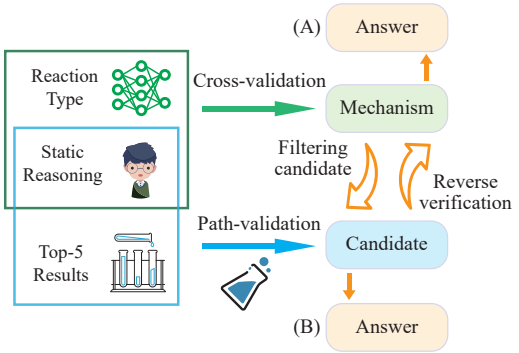

Figure 3: Illustration of the two prompt-guided expert skills for autonomous agentic reasoning.

2.3.2 Emergent Trajectory Optimization

Beyond fixed expert workflows, we enable unconstrained action execution during reinforcement learning. The agent operates in a flat action space and autonomously decides when to invoke external tools, how to incorporate environmental feedback $o_t = \mathcal{T}_{a_t}(x, s_t)$, and whether to trigger reflection-based trajectory revision.

Here, we adopt GRPO to optimize long-horizon skill composition. A complete interaction trajectory is defined as $\tau = (s_1, a_1, o_1, \ldots, s_T, a_T, o_T)$, and optimization is performed over trajectory-level advantages $A(\tau)$. The total reward is defined as:

- **Outcome Reward** ($R_{outcome}$)**:** Accuracy Reward, measures the final prediction accuracy.
- **Process Reward** ($R_{process}$)**:** Encourages interaction behaviors, including verification-based self-reflection steps.
- **Efficiency Penalty** (R_{eff})**:** Penalizes redundant tool calls and cyclic execution reasoning.

The final trajectory reward is computed as:

$$R(\tau) = R_{outcome} + R_{process} + R_{eff}$$

This framework enables the emergence of complex behaviors such as tool reliability estimation and uncertainty-aware reasoning under incomplete feedback, without explicitly specifying these behaviors in the policy design.

3 Experiments

In this section, we first present a comprehensive comparative evaluation between our method and existing baselines. Next, we examine the transition from static to agentic reasoning. Furthermore, we conduct ablation studies for each paradigm to discuss the contributions of various components. Finally, we provide case study (in **Appendix A.2.6**) to investigate different reasoning trajectories.

3.1 Experimental Setting

3.1.1 Dataset and Metrics

The raw Open Reaction Database (ORD) has been preprocessed by ORDerly (Wigh et al., 2024) and split into 832K for training and 86K for testing. It is worth noting that approximately 65% of the USPTO-MIT test samples already appear in the ORD training set. So we do not use USPTO-MIT test set in order to ensure a fair comparison.

We employ various metrics including Validity (whether the output SMILES can be processed by RDKit), Exact Match (after canonicalization) and molecular fingerprint similarity (FTS, including MORGAN, RDK, and AVALON fingerprints). We deliberately avoid relying on text-based similarity metrics (e.g., BLEU and Levenshtein), since they poorly reflect actual molecular differences, even a single alteration in a SMILES string can correspond to a substantial change in the chemical structure.

3.1.2 Baselines

To comprehensively evaluate the performance, we compare our model with: (1) general LLMs, including GPT-4o, DeepSeek-R1 (Guo et al., 2025), DeepSeek-R1-Distill-Llama-70B, DeepSeek-R1-Distill-Qwen series (32B/14B/7B), Qwen3-32B (Yang et al., 2025), and Qwen2.5-72B-Instruct; (2) chemical LLMs, including ChemDFM (OpenDFM Team, 2024), Text-Chem-T5 (Christofidellis et al., 2023), Mol-Instructions (Fang et al., 2023), BioT5+ (Pei et al., 2024); and (3) traditional task-specific models including Chemformer (Irwin et al., 2022), which is reported to achieve the best Top-1 accuracy on USPTO-MIT (Chen and Jung, 2022), Molecular Transformer (Schwaller et al., 2019), and ReactionT5-V2 (Sagawa and Kojima, 2023).

For general LLMs, we use the same prompt as our method, and for chemical LLMs, we adopt the training prompts specified in the relevant papers.

3.2 Main Result

As shown in Table 1, our method outperforms all compared LLMs and traditional task-specific models, achieving the highest Exact Match, Validity, and fingerprint similarity (FTS).

3.3 Progressive Paradigm Comparison

To evaluate our evolutionary reasoning framework, we compare three independently trained paradigms that conceptually progress from static to agentic reasoning: **Static Reasoning Enhancement** (I), **Grounded Agentic Reasoning** (II), and **Autonomous Agentic Reasoning** (III). As shown in Table 2, the results demonstrate the methodological advantages of our evolving framework.

3.4 Ablation Studies for Static Reasoning

3.4.1 Scaling High-Quality Reasoning Data

We initialize our RL foundation via SFT on existing CoT data, during which additional high-quality reasoning trajectories are sampled. As is shown in Table 3, re-training the model on the expanded dataset significantly boosts accuracy. This indicates that iteratively scaling high-quality reasoning data can strengthen LLM reasoning capabilities.

3.4.2 Integrating RAG with CoT Reasoning

We assess the effect of retrieving similar cases on reasoning, with results summarized in Table 4. The findings show that the integration of RAG substantially enhances reaction prediction, as it allows the model to combine analogical reasoning from similar cases with deductive steps, effectively enabling the LLM to reason in a chemist-like manner.

3.4.3 Effect of Reward Scheduling

To evaluate the effectiveness of the proposed batch-adaptive reward scheduling strategy, we compare two different optimization schemes for balancing the feasibility reward R_{feas} and the accuracy reward R_{acc}: (1) manually designed two-stage scheduling, and (2) batch-adaptive scheduling.

Results in Table 5 and Table 6 show that fixed weighting leads to unstable optimization, often requiring extensive trial-and-error to avoid convergence to suboptimal solutions. The two-stage schedule improves stability but remains sensitive to the manually selected transition step, which also demands considerable task-specific tuning. In contrast, the proposed adaptive scheduler achieves the best performance in a more efficient manner.

3.5 Ablation Studies for Grounded Reasoning

3.5.1 Grounding Source Ablation

To analyze the contribution of different grounding signals, we conduct an ablation study over the three external observation sources o^{reason} (Static Reasoning LLM), o^{cand} (Chemistry-Specific Predictor), and o^{type} (Reaction Type Classifier).

Specifically, we evaluate the model under different grounding configurations by incrementally removing each observation source from the grounded context. The results in Tabel 7 show that all grounding components contribute positively to final performance, while combining all sources achieves the best overall accuracy.

Among the individual components, o^{cand} provides the largest performance improvement, indicating that high-confidence external candidates offer strong guidance for reaction prediction. The reasoning traces o^{reason} further improve performance by supplying intermediate chemical analysis that helps the model verify and refine candidate selection. And the o^{type} mainly provides coarse-grained mechanistic priors and show comparatively smaller but still consistent gains.

3.5.2 Noisy Evidence Robustness

We further evaluate grounded reasoning under noisy external observations by separately injecting incorrect candidate predictions, misleading reasoning traces, and wrong reaction type labels. For each noise class, we construct 1K corrupted test samples to analyze their individual impact on robustness.

Results in Tabel 8 show that incorrect candidate predictions cause the largest performance drop, indicating that candidate pools strongly influence the final decision process. However, the grounded model still outperforms variants without candidate predictions in Tabel 7, so we argue that the LLM can often leverage reasoning traces to recover correct predictions from noisy candidates.

In contrast, misleading reasoning traces and incorrect reaction type labels only produce minor performance degradation. These findings suggest that different grounding signals contribute unequally to model decisions, while diverse and complementary evidence helps the model filter noisy information and make more robust predictions.

Table 1: Comparison of our method with various baselines on the task of forward reaction prediction. The results for Molecular Transformer are directly taken from existing work (Wigh et al., 2024). Due to API cost constraints, the results for GPT-4o (marked †) are based on a randomly sampled subset of 500 instances from the ORD test set. The top results are marked in **bold**.

Model	Model Type	Validity (%)	Exact Match (%)	FTS (%)		
				MORGAN	RDK	AVALON
Chemformer	Task-Specific Model	98.57	88.13	92.40	94.35	95.12
Molecular Transformer		99.66	85.84	–	–	–
ReationT5-V2		92.80	**91.20**	–	–	–
GPT-4o	General LLM	83.05	**28.26**	64.93	72.13	72.09
DeepSeek-R1		68.54	11.68	55.71	64.09	64.98
Qwen2.5-72B-Instruct		35.27	0.54	37.46	45.95	46.01
DeepSeek-R1-Distill-Llama-70B		66.42	7.20	49.06	59.17	59.67
DeepSeek-R1-Distill-Qwen-32B		57.58	6.52	50.50	61.05	61.05
Qwen3-32B		68.17	6.04	50.34	60.88	60.93
DeepSeek-R1-Distill-Qwen-14B		43.72	1.69	32.72	40.92	40.95
DeepSeek-R1-Distill-Qwen-7B		47.74	1.21	43.50	54.58	53.77
ChemDFM-13B	Chemical LLM	98.29	**52.41**	77.27	82.03	82.15
ChemDFM-8B		97.80	48.02	74.69	79.85	80.02
Text-Chem-T5		95.67	47.88	76.45	81.81	81.81
BioT5+		97.62	16.09	59.47	67.79	67.82
Mol-Instructions		98.15	1.83	44.01	52.53	52.67
Reaction-Mind (Best Result)	–	**99.70**	**91.98**	**95.54**	**96.71**	**97.79**

Table 2: Exact Match (%) performance across three independent reasoning paradigms.

Paradigm	Exact Match (%) (Best Result)
I	90.16
II	91.31
III	**91.98**

Table 3: The Exact Match (%) performance after SFT and GRPO on raw and scaled reasoning datasets.

	Raw Data	Scaling Data
After SFT	61.04	68.75
After GRPO	69.35	**76.52**

Table 4: Exact Match (%) before and after RL for models trained with and without retrieving similar cases.

	w/o RAG	w/ RAG
Exact Match (%)	68.75 / 76.52	84.71 / **89.33**

Table 5: Exact Match (%) performance (EM) of the selected optimal combinations of R_{feas} and step K during RL, for forward prediction. Where R_{feas} pre / post denote the **Feasibility Reward** applied before and after step K, respectively.

R_{feas} pre	K	R_{feas} post	EM (%)
1.0	300	0.5	87.61
0.5	300	0.1	**89.96**
0.2	500	0.1	88.02

3.6 Ablation Studies for Autonomous Agent

3.6.1 Tool Allocation and Skill Initialization

In this section, we investigate the impact of prompt-based skill initialization on the final performance of LLM, focusing on two distinct human-engineered operational paradigms: **Type-guided skill** and **Candidate-driven skill**. Note that we do not retrieve similar cases in the following sections due to the high computational cost. As summarized in Table 9, the candidate-driven approach significantly outperforms the type-guided counterpart, achieving a superior final accuracy of 91.98%.

The superiority of the candidate-driven paradigm stems from search space collapsing. Retrieving a Top-5 candidate pool collapses the generative space into a localized decision space, allowing the LLM to validate and re-rank candidates rather than generate structures from scratch during GRPO training, substantially reducing prediction difficulty.

3.6.2 Handling Tool Failures

During inference, we found that intensive use of Retrosynthetic Planner causes queuing and time-

Table 6: Exact Match (%) performance after RL for models trained with manually designed two-stage reward scheduling (select the best result) versus dynamic reward adjustment.

	Manually	Dynamic
Exact Match (%)	89.96	**90.16**

Table 7: Exact Match (%) performance before and after RL under different combinations of observation sources. The first row indicates the baseline without any source.

Observation Sources	Exact Match (%)
None	6.04 / –
o^{cand}	**78.82** / 90.23
o^{cand} + o^{type}	76.46 / 90.29
o^{cand} + o^{reason}	73.07 / 90.59
o^{reason} + o^{type}	67.22 / 75.83
o^{cand} + o^{reason} + o^{type}	73.40 / **90.71**

Table 8: Exact Match (%) performance (EM) under noisy observations of different classes. Each row shows the result in inference when the corresponding observation source is corrupted.

Noisy Evidence	EM (%)	Drop (%)
o^{cand}	78.95	11.76
o^{reason}	90.30	0.41
o^{type}	90.43	0.28

Table 9: Exact Match (%) performance under different skill initialization paradigms after RL.

Skill	Exact Match (%)
Type-guided	90.52
Candidate-driven	**91.98**
Emergent	90.74

outs. The LLM responds by repeatedly invoking failed tools, rapidly exhausting the context window.

To mitigate this, we employ a two-fold approach. First, we instruct the model by System Prompt to give up on failed tools and instead rely on the evidence it already has. Second, we optimize the deployment infrastructure of Retro-star to improve concurrent throughput, significantly reducing timeout rates. While these guardrails successfully eliminate tool loops in standard evaluations, we strategically retain the efficiency penalty in GRPO reward design as a defensive safeguard against redundant actions in unforeseen scenarios.

3.6.3 Analysis of Emergent Strategies

Beyond predefined workflows, we further investigate whether the LLM can autonomously develop effective reasoning strategies under unconstrained settings. Specifically, we allow the model to freely decide when to invoke external tools during inference. To avoid pathological tool-use loops, the System Prompt limits Retrosynthetic Planner to at most three invocations and all other tools to one invocation. Violating these constraints receives a R_{eff} of -1.0, while valid planner usage receives a $R_{process}$ of 0.1.

As shown in Table 9, the Emergent strategy achieves 90.74% Exact Match accuracy, outperforming the Type-guided policy (90.52%) while remaining slightly below the Candidate-driven policy (91.98%). Further analysis of tool invocation frequencies before and after GRPO reveals clear behavioral evolution. The usage rate of the Chemistry-Specific Predictor increases from 91% to 99%, while the average number of Retrosynthetic Planner calls rises from 0.6 to 1.8 per trajectory, whereas other tools remain relatively stable (around 90%).

These observations suggest that the LLM gradually learns to identify and prioritize the most informative signals, particularly high-confidence candidate predictions. Moreover, comparing the results in Table 7 and Table 9, we observe that autonomous tool selection consistently outperforms directly providing all external information to the LLM. We attribute this improvement to the LLM ability to actively invoke the verification model during reasoning, enabling reflection-based validation.

4 Conclusion

We present **Reaction-Mind**, a progressive framework that evolves chemical reaction prediction from static reasoning to agentic reasoning through three stages: static reasoning enhancement, grounded reasoning with external evidence, and tool-interactive skill learning via reinforcement learning. Experiments show that **Reaction-Mind** consistently outperforms all compared LLMs and traditional task-specific models. Ablation studies validate the benefits of adaptive reward scheduling, multi-source grounding, and emergent tool-use strategies. The results demonstrate that interactive reasoning significantly improves performance in chemical reaction prediction.

Limitations

Our method has several limitations. Currently, experiments are limited to forward reaction prediction; future work will extend to retrosynthesis and other tasks. The tool set is relatively small and will be expanded to support more diverse reasoning. Additionally, the implementation ignores stereochemistry and reaction conditions beyond SMILES, potentially affecting stereoselective or condition-sensitive reactions. Addressing these issues remains the focus of future work.

AI Assistant Usage Statement

We used ChatGPT and Copilot for minor language and code assistance. No AI was used for generating core ideas or results. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Mahdi Astaraki, Mohammad Arshi Saloot, Ali Shiraee Kasmaee, Hamidreza Mahyar, and Soheila Samiee. 2026. When iterative rag beats ideal evidence: A diagnostic study in scientific multi-hop question answering. *arXiv preprint arXiv:2601.19827*.

Andres M Bran, Theo A Neukomm, Daniel Armstrong, Zlatko Jončev, and Philippe Schwaller. 2026. Chemical reasoning in llms unlocks strategy-aware synthesis planning and reaction mechanism elucidation. *Matter*.

Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. 2020. Retro*: learning retrosynthetic planning with neural guided a* search. In *International conference on machine learning*, pages 1608–1616. PMLR.

Shuan Chen and Yousung Jung. 2022. A generalized-template-based graph neural network for accurate organic reactivity prediction. *Nature Machine Intelligence*, 4(9):772–780.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR.

DeepSeek-AI. 2025. Deepseek-r1-distill-qwen-7b. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B. Distilled reasoning model based on Qwen2.5-7B.

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

Peter Gedeck, Bernhard Rohde, and Christian Bartels. 2006. Qsar- how good is it in practice? comparison of descriptor sets on an unbiased cross section of corporate data sets. *Journal of chemical information and modeling*, 46(5):1924–1936.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. 2022. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022.

Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. 2021. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826.

Hanbum Ko, Chanhui Lee, Ye Rin Kim, Rodrigo Hormazabal, Sehui Han, Sungbin Lim, and Sungwoong Kim. Reactionreasoner: Towards reasoning llm for chemical reaction prediction. In *NeurIPS 2025 AI for Science Workshop*.

Hanbum Ko, Chanhui Lee, Ye Rin Kim, Rodrigo Hormazabal, Sehui Han, Sungbin Lim, and Sungwoong Kim. 2026. Retroreasoner: A reasoning llm for strategic retrosynthesis prediction. *arXiv preprint arXiv:2603.12666*.

HKAIR Lab. 2024. Hk-o1aw models: Leveraging o1 slow thinking in the development of hong kong legal large language models. https://github.com/HKAIR-Lab/HK-O1aw.

Greg Landrum. 2016. Rdkit: Open-source cheminformatics software. https://www.rdkit.org. Version as of your usage date.

Daniel E Levy. 2017. *Arrow-pushing in organic chemistry: an easy approach to understanding reaction mechanisms*. John Wiley & Sons.

Min Htoo Lin, Zhengkai Tu, and Connor W Coley. 2022. Improving the performance of models for one-step retrosynthesis through re-ranking. *Journal of cheminformatics*, 14(1):15.

Xuan Lin, Qingrui Liu, Hongxin Xiang, Daojian Zeng, and Xiangxiang Zeng. 2025. Enhancing chemical reaction and retrosynthesis prediction with large language model and dual-task learning. *arXiv preprint arXiv:2505.02639*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, and 1 others. 2025. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034.

Siddharth Narayanan, James Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Michael Pieler, Sam Rodriques, and Andrew White. 2026. Training a scientific reasoning model for chemistry. *Advances in Neural Information Processing Systems*, 38:157671–157710.

OpenDFM Team. 2024. Chemdfm-13b/8b. https://github.com/OpenDFM/ChemDFM. Open-source chemical LLM models.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. 2024. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. *arXiv preprint arXiv:2402.17810*.

Qwen Team. 2024. Qwen2.5-72b-instruct. https://huggingface.co/Qwen/Qwen2.5-72B-Instruct. Instruction-tuned Qwen2.5 model.

RDKit-Book. 2025a. RDKit: Open-source cheminformatics. https://www.rdkit.org/docs/RDKit_Book.html. Accessed: 2025-08-01.

RDKit-Book. 2025b. RDKit: Open-source cheminformatics. https://www.rdkit.org/docs/RDKit_Book.html. Accessed: 2025-08-01.

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

Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. 2021. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284.

Tatsuya Sagawa and Ryosuke Kojima. 2023. Reactiont5: a large-scale pre-trained model towards application of limited reaction data. *arXiv preprint arXiv:2311.06708*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Nadine Schneider, Daniel M Lowe, Roger A Sayle, and Gregory A Landrum. 2015a. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of chemical information and modeling*, 55(1):39–53.

Nadine Schneider, Roger A Sayle, and Gregory A Landrum. 2015b. Get your atoms in order-an open-source implementation of a novel and robust molecular canonicalization algorithm. *Journal of chemical information and modeling*, 55(10):2111–2120.

Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. 2018. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical science*, 9(28):6091–6098.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. 2019. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583.

David K Smith. 2023. Priority and selectivity rules to help students predict organic reaction mechanisms. *Journal of Chemical Education*, 100(3):1164–1178.

Qingyu Wang, Xinyuan Cai, Xiang Cheng, YuZhe Gao, and Bo Xu. Towards knowledge-and-data-driven organic reaction prediction: Rag-enhanced and reasoning-powered hybrid system with llms. In *The Fourteenth International Conference on Learning Representations*.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, and 1 others. 2026. Agentic reasoning for large language models. *arXiv preprint arXiv:2601.12538*.

Daniel S Wigh, Joe Arrowsmith, Alexander Pomberger, Kobi C Felton, and Alexei A Lapkin. 2024. Orderly: data sets and benchmarks for chemical reaction data. *Journal of Chemical Information and Modeling*, 64(9):3790–3798.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28489–28503.

Renjun Xu and Yang Yan. 2026. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*.

Chaochao Yan, Peilin Zhao, Chan Lu, Yang Yu, and Junzhou Huang. 2022. Retrocomposer: composing templates for template-based retrosynthesis prediction. *Biomolecules*, 12(9):1325.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Xueming Yang and Kopin Liu. 2004. Modern trends in chemical reaction dynamics: Experiment and theory.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Barry Zhang, Keith Lazuka, and Mahesh Murag. 2025. Equipping agents for the real world with agent skills. *Anthropic Engineering Blog*.

Bingxi Zhao, Lin Geng Foo, Ping Hu, Christian Theobalt, Hossein Rahmani, and Jun Liu. 2025. Llm-based agentic reasoning frameworks: A survey from methods to scenarios. *arXiv preprint arXiv:2508.17692*.

Zihan Zhao, Bo Chen, Jingpiao Li, Lu Chen, Liyang Wen, Pengyu Wang, Zichen Zhu, Danyang Zhang, Yansi Li, Zhongyang Dai, and 1 others. 2024a. Chemdfm-x: towards large multimodal model for chemistry. *Science China Information Sciences*, 67(12):1–2.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, and 1 others. 2024b. Chemdfm: Dialogue foundation model for chemistry. *arXiv e-prints*, pages arXiv–2401.

Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2025. Knowagent: Knowledge-augmented planning for llm-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3709–3732.

A Appendix

A.1 Related work

A.1.1 LLM Agentic Reasoning

Agentic reasoning extends LLMs from static chain-of-thought prompting to autonomous agents that plan, act, and adapt via tool use, memory, and environmental feedback. Recent surveys (Wei et al., 2026; Zhao et al., 2025) provide unified taxonomies, typically organizing systems into single-agent, tool-augmented, and multi-agent paradigms, and further distinguishing in-context orchestration from post-training optimization. Foundational methods such as ReAct (Yao et al., 2022) interleave reasoning and action to ground generation in external feedback, while Toolformer (Schick et al., 2023) learns tool invocation decisions during training. Subsequent approaches extend these ideas via reflection and memory, lifelong embodied learning, and structured multi-agent collaboration (Zhu et al., 2025; Wu et al., 2025). Overall, these works shift LLMs toward environment-interacting agents that tightly couple reasoning, execution, and iterative improvement in open-ended tasks.

A.1.2 LLM Agent Skills

Agent skills introduce a modular abstraction layer for equipping LLM agents with reusable procedural capabilities, extending beyond monolithic prompts and atomic tool calls. Recent work formalizes skills as portable, composable artifacts for on-demand capability extension without retraining. Anthropic Agent Skills (Zhang et al., 2025) standard popularizes this paradigm. Some other works (Xu and Yan, 2026) systematize it across architecture, acquisition, deployment, and security, highlighting both autonomous skill generation and vulnerability risks in community-contributed skills. Empirical studies (Wei et al., 2026) further show that skill effectiveness depends on routing and task–skill alignment, with non-trivial robustness and application challenges.

A.1.3 Reasoning LLMs in Chemistry

Recent work has explored LLMs for computational chemistry, including forward reaction prediction, retrosynthesis, and molecular design.

For reaction prediction, ChemDual (Lin et al., 2025) improves bidirectional consistency via joint forward–backward modeling, while ReactionReasoner (Ko et al.) enhances interpretability through synthetic step-by-step reasoning traces. In ret-

rosynthesis, LLMs are increasingly integrated into search-based frameworks. They enable strategy-aware planning (Bran et al., 2026) and model full synthesis pathways with improved efficiency (Bran et al., 2026). RetroReasoner (Ko et al., 2026) incorporates chemist-inspired or chain-of-reaction reasoning for multi-step planning. Tool-augmented and agentic systems such as ChemCrow (M. Bran et al., 2024) further extend these capabilities.

Beyond synthesis planning, LLMs support molecular design by coupling retrosynthesis with property optimization, while benchmarks such as ChemBench (Mirza et al., 2025) systematically evaluate chemical reasoning across tasks. Despite these advances, challenges remain in multi-step reasoning reliability, chemical validity, and hallucination, motivating hybrid LLM–search–tool paradigms as an emerging direction.

A.2 Supplementary Material

A.2.1 Reaction Type Classifier

Given the SMILES of chemical inputs, we utilize RDKit (Landrum, 2016) to extract a comprehensive suite of molecular fingerprints. To capture multi-scale structural and topological features, we concatenate five distinct fingerprint modalities: RDK (Schneider et al., 2015b), LAYERED (RDKit-Book, 2025a), PATTERN (RDKit-Book, 2025b), AVALON (Gedeck et al., 2006), and Morgan fingerprints (Rogers and Hahn, 2010).

The concatenated fingerprint vector $Mol\text{-}FP$ is passed through the MLP. We train this classifier on the Schneider-50K dataset (Schneider et al., 2015a), which categorizes reactions into 50 representative chemical classes. After optimization, the second-to-last layer is extracted as a low-dimensional molecular embedding ($Rea\text{-}Embed$), while the final layer outputs the category probability ($Class\text{-}Out$):

$$\begin{aligned} Rea\text{-}Embed &= \boldsymbol{Layer1\&2}\,(Mol\text{-}FP) \\ Class\text{-}Out &= \boldsymbol{Layer3}\,(Rea\text{-}Embed) \end{aligned} \tag{4}$$

A.2.2 Similar-Case Retrieval Databases

For each reaction in the library, we compute its $Rea\text{-}Embed$ and predicted type via the pre-trained classifier. During inference, given a query input, we first identify its reaction type and then retrieve top-k nearest neighbors within that specific category based on the Euclidean distance (L_2 norm) of their embeddings. The full SMILES representations of these retrieved exemplars are appended to the prompt as in-context chemical references.

A.2.3 Reasoning Trajectory Construction

Step 1: Preliminary Construction Following the prompt-based distillation paradigm (Lab, 2024), we extract a subset from the USPTO-MIT training set to construct high-quality rationale data. We leverage Qwen2.5-72B-Instruct (Qwen Team, 2024) to reverse-engineer the reaction mechanisms via deductive reasoning. Even though the complete reaction SMILES is provided during this generation phase, the model is required to step-by-step reconstruct the underlying transformation pathways. Rigorous heuristic filters (e.g., format compliance and chemical entity matching) are applied to yield 50K validated CoT data samples.

Step 2: Distillation and Validation Using the constructed 50K CoT dataset, we first perform supervised fine-tuning (SFT) on DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025). To further align the static reasoning with exact targets, we execute a validation-driven RL phase on the ORD training set via GRPO. During this distillation phase, we exclusively harvest the trajectories that culminate in the exact ground-truth product. This stringent filter yields a final distilled dataset of 100K high-fidelity validated CoT trajectories.

A.2.4 Group Relative Policy Optimization

Given an input query $q \sim P(Q)$, GRPO samples a group of G responses $\{y_1, y_2, \ldots, y_G\}$ from the current policy $\pi_{\theta_{\text{old}}}$. The core idea is to update the policy π_θ by maximizing an objective function that encourages responses with higher-than-average rewards within their group. The GRPO objective function is defined as follows:

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_{q\sim P(Q),\, \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \\ \min\Big(c_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(c_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right) \\ \hat{A}_{i,t}\Big) - \beta\,\mathbb{D}_{KL}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right]\Bigg] \end{aligned} \tag{5}$$

where ϵ is the clipping ratio, β is the coefficient for KL-divergence loss, and π_{ref} is the reference policy. $c_{i,t}(\theta)$ is the importance sampling ratio for token $y_{i,t}$ (the t-th token of the i-th response y_i):

$$c_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|q, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|q, y_{i,<t})} \tag{6}$$

$\hat{A}_{i,t}$ is advantage estimate for all tokens in response y_i and is calculated by normalizing the re-

wards $\{r_1, r_2, \ldots, r_G\}$ using the group mean and standard deviation:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})} \quad (7)$$

A.2.5 Dynamic Reward Adjustment

Let w denote the dynamic weight scaled for R_{feas}. For each rollout with n candidate trajectories, we compute the number of successfully verified, chemically feasible paths m. The dynamic weight w is formalized as:

$$w = \frac{n - m}{n} \quad (8)$$

When a generation batch exhibits low chemical compliance ($m \to 0$), w automatically scales up toward 1.0 to actively incentivize the policy to explore alternative, chemically plausible spaces. Conversely, as the policy converges and more candidates pass environmental verification ($m \to n$), w gracefully degrades to 0, systematically shifting the optimization focus exclusively toward the exact match target R_{acc}. This adaptive scheduling successfully eliminates manual parameter searching by aligning the reward landscape directly with the real-time learning state of the model.

A.2.6 Case Study

As is illustrated from Figure 4 to Figure 8, for the LLM trained with the **Emergent Skill**, we selected several reasoning trajectories and analyzed its tool invocation (call success and result correctness) alongside the final predictions. Note that verified correct calls or answers are highlighted in green, while incorrect ones are marked in red. Each round of conversation (including the tool invocation request and the returned result) is highlighted within a gray box.

Across different cases, the LLM demonstrates robust multi-tool coordination under incomplete feedback, unsuccessful tool invocations, and missing intermediate predictions, while still maintaining reliable reasoning and accurate final prediction.

A.2.7 Prompt Setting

We design distinct system prompts for the autonomous agentic reasoning. Each prompt enforces a different reasoning strategy. All prompts share the same tool interface. We present only the **Tool Descriptions** and **Emergent Skill** prompt in the anonymous link. The complete prompts for **Candidate-Driven Skill** and **Type-Guided Skill** will be made publicly available upon the acceptance of this paper.

A.3 Anonymous Link

Here, the anonymous link[1] has been provided to open-source some of the code (both for training and inference, along with the corresponding parameters) and prompts referenced in this work. The complete code, data and prompts will be made publicly available upon the acceptance of this paper.

[1] https://github.com/hajimi618/Chem-Reason-Agent

User
Reaction input: CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\n\nPredict the possible product(s).

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n",
"output": "<tool_call>\n{\"name\": \"predict_reaction_type\", \"arguments\":
{\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\"}}\n</tool_call>

User
<tool_response>\nPredicted reaction type: Bromination\nConfidence: 0.0210\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"predict_reaction_products\", \"arguments\": {\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\", \"top_k\": 5}}\n</tool_call>

User
<tool_response>\nTop-5 reaction predictions for reactants: CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1
1. CN(C)Cc1ccc(-c2cncc(Br)c2)nc1\n 2. CN(C)Cc1cncc(-c2cncc(Br)c2)c1\n 3. CN(C)Cc1ccc(-c2cncc(Br)c2)cn1\n
4. CN(C)Cc1csc(-c2cncc(Br)c2)c1\n 5. CN(C)Cc1ccc(-c2cncc(Br)c2)cc1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"query_reasoning_agent\", \"arguments\":
{\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\",
\"prompt\": \"This is a bromination reaction\"}}\n</tool_call>

User
<tool_response>\n=== Reasoning Agent Chain-of-Thought ===\nOkay, I will follow a structured approach to determine the reaction product.
Here is a step-by-step analysis. (...reasoning process omitted...) </think>\n\nUnder the stated conditions,
the expected products of the reaction are\n===\n Reasoning Agent Prediction: SMILES: CN(C)Cc1ccc(-c2cncc(Br)c2)nc1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"validate_reaction_pathway\", \"arguments\": {\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\",\"product\": \"CN(C)Cc1ccc(-c2cncc(Br)c2)nc1\"}}\n</tool_call>

User
<tool_response>\nAgent call failed: attempt 3/3: ReadTimeout

Assistant (Reaction-Mind)
<think>\n\n</think>\n\nGiven the absence of a confirmed retrosynthetic pathway and the low confidence in the predicted reaction type, I will proceed with a detailed analysis of the reaction components to make a reasoned prediction.\n\n### Step-by-Step Analysis:\n1. **Reactants and Reagents**:\n - **CN(C)Cc1ccc(I)nc1**: An iodinated aromatic compound with a pyridine-like structure.\n - **OB(O)c1cncc(Br)c1**: A brominated aromatic compound with a hydroxyl group.\n - **O=C([O-])[O-]**: A peroxycarboxylic acid, likely acting as an oxidizing agent.\n - **[Na+].[Na+]**: Sodium ions, possibly as a counterion or to assist in deprotonation.\n - **COCCOC**: Diethyl ether, a common solvent.\n - **CCO**: Ethanol, a solvent or potential nucleophile.\n - **Palladium complex**: A Pd(0) complex with phosphine ligands, likely acting as a catalyst for cross-coupling.\n\n2. **Reaction Type**:\n - The presence of a Pd(0) complex and an iodinated aromatic compound suggests a **Suzuki-Miyaura coupling** or a **Stille coupling**.\n - The brominated aromatic compound is a common partner in such cross-coupling reactions.\n - The peroxycarboxylic acid may act as an oxidizing agent to regenerate the Pd(0) catalyst.\n\n3. **Predicted Product**:\n - A cross-coupling reaction between the iodinated and brominated aromatic compounds would form a new carbon-carbon bond.\n - The product would retain the functional groups from both reactants, with the iodine and bromine atoms likely being replaced by the coupling partner.\n\n4. **Chemical Reasoning**:\n - The iodinated compound (CN(C)Cc1ccc(I)nc1) would act as the aryl halide in the coupling.\n - The brominated compound (OB(O)c1cncc(Br)c1) would act as the aryl halide partner.\n - The Pd(0) complex would facilitate the coupling, forming a new aromatic ring system.\n\n### Final Prediction:\nThe most likely product is a **biphenyl-like structure** where the iodinated and brominated aromatic rings are coupled via a carbon-carbon bond.\n\nSMILES:
CN(C)Cc1ccc(-c2cncc(Br)c2)nc1"

Figure 4: Failure-tolerant reasoning despite unsuccessful verification-based self-reflection and missing Static Reasoner prediction. The LLM integrates mechanistic analysis, reaction type guidance, and candidate proposals to recover the correct product.

User
Reaction input: CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\n\nPredict the possible product(s).\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n", "output": "<tool_call>\n{\"name\": \"predict_reaction_type\", \"arguments\": {\"reactants\": \"CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\"}}\n</tool_call>

User
<tool_response>\nPredicted reaction type: Chloro N-alkylation\nConfidence: 0.9627\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"predict_reaction_products\", \"arguments\": {\"reactants\": \"CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\", \"top_k\": 5}}\n</tool_call>

User
<tool_response>\nTop-5 reaction predictions for reactants: CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\n 1. CC1(N)CN(Cc2ccccc2)C(=O)C12CC2\n 2. CC(C)(C)OC(=O)NC1(C)CN(Cc2ccccc2)C(=O)C12CC2 \n 3. CC1(NC(=O)OC(C)(C)C)CN(Cc2ccccc2)C(=O)C12CC2\n 4. CC1(N2Cc3ccccc3C2=O)CN(Cc2ccccc2)C(=O)C12CC2\n 5. CC1(C)CN(Cc2ccccc2)C(=O)C12CC2\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"validate_reaction_pathway\", \"arguments\": {\"reactants\": \"CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\", \"product\": \"CC1(N)CN(Cc2ccccc2)C(=O)C12CC2\"}}\n</tool_call>

User
<tool_response>\nFound 1 retrosynthetic pathway(s) for reactants → CC1(N)CN(Cc2ccccc2)C(=O)C12CC2:\n\nPathway 1 (score: 0.9997):\n Step 1: retrosynthesis_path → CC1(N)CN(Cc2ccccc2)C(=O)C12CC2>0.9997>CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.ClCc1ccccc1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"query_reasoning_agent\", \"arguments\": {\"reactants\": \"CC(C)(C)OC(=O)NC1(C)CNC(=O)C12CC2.CC(C)(C)O[K].ClCc1ccccc1.O.CC(=O)N(C)C\", \"prompt\": \"focus on the chloro N-alkylation reaction pathway\"}}\n</tool_call>

User
<tool_response>\nReasoning agent call failed: attempt 3/3: \n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\nBased on the available evidence, the most likely product of this chloro N-alkylation reaction is the N-alkylated compound formed by the reaction of the amine group with the chloroarene.\n\nThe top-ranked prediction from the model is:\nCC1(N)CN(Cc2ccccc2)C(=O)C12CC2\n\nThis product features the expected alkylation of the nitrogen atom with the chlorobenzene moiety.\n\nSMILES: CC1(N)CN(Cc2ccccc2)C(=O)C12CC2"

Figure 5: Reasoning with unsuccessful Static Reasoner invocation. Reaction type prediction and candidate proposals are provided, the correct candidate is included, and verification-based self-reflection confirms the final correct prediction.

User
Reaction input: CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)[P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\n\nPredict the possible product(s).

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n", "output": "<tool_call>\n{\"name\": \"predict_reaction_type\", \"arguments\": {\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\"}}\n</tool_call>

User
<tool_response>\nPredicted reaction type: Bromination\nConfidence: 0.0210\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"predict_reaction_products\", \"arguments\": {\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\", \"top_k\": 5}}\n</tool_call>

User
<tool_response>\nTop-5 reaction predictions for reactants: CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\n 1. CN(C)Cc1ccc(-c2cncc(Br)c2)nc1\n 2. CN(C)Cc1cncc(-c2cncc(Br)c2)c1\n 3. CN(C)Cc1ccc(-c2cncc(Br)c2)cn1\n 4. CN(C)Cc1csc(-c2cncc(Br)c2)c1\n 5. CN(C)Cc1ccc(-c2cncc(Br)c2)cc1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"query_reasoning_agent\", \"arguments\": {\"reactants\": \"CN(C)Cc1ccc(I)nc1.OB(O)c1cncc(Br)c1.O=C([O-])[O-].[Na+].[Na+].COCCOC.CCO.c1ccc([P](c2ccccc2)(c2ccccc2)[Pd]([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)([P](c2ccccc2)(c2ccccc2)c2ccccc2)cc1\", \"prompt\": \"This is a bromination reaction\"}}\n</tool_call>

User
<tool_response>\n=== Reasoning Agent Chain-of-Thought ===\nOkay, I will follow a structured approach to determine the reaction product. Here is a step-by-step analysis. {Analyzing the given reactants and reagents} The reactants and reagents include O=Cc1ccccc1 (benzaldehyde), Br[Mg] The reasoning process confirms that the major product is Cc1ccccc1CBr, where the bromine atom is attached to the aromatic ring. The reaction conditions and the presence of the bromine atom suggest that the bromine atom is the most likely site of reduction or substitution. The aromatic ring is stable and less likely to undergo reduction under these conditions. The bromine atom is a good leaving group and can be displaced by a nucleophile, but in this case , the reaction conditions suggest a reduction rather than a substitution. The reduction of the bromine atom to a hydrogen atom is a common reaction in organic chemistry, especially in the presence of a reducing agent. The aromatic ring remains intact, and the bromine atom is the most likely site of ... (truncated)...of the bromine atom is reduced to a hydrogen atom. 4. The final product is Cc1ccccc1CBr,where the bromine atom is attached to the aromatic ring. The aromatic ring is stable and less likely to undergo reduction under these conditions. The bromine atom is the most likely site of reduction. The aromatic ring is stable and less likely to undergo reduction under these conditions. 2. The bromine atom is not involved in the reaction, as it is not present in the reactants. 3. The cyclic ether (C1CCOC1) is likely a solvent or a ligand to stabilize the nickel catalyst. 4. The reaction conditions (e.g., temperature, pressure, and the reaction. 5. The reaction is a typical ester, and the [Na+] ions is a nucleophilic attack step, The [H-] ion is involved in the deprotonates the alcohol, and the [Na+] ion is involved in the carbonyl carbon of the carboxylic acid, forming the ester. The final product is O=C(O)C1CCc2ccccc21.\n===nReasoning agent did not produce a valid SMILES prediction.\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\nBased on the available information, I will analyze the reaction and make a prediction.\n\nThe reaction type was predicted as bromination with low confidence (0.0210). The top product candidates from the prediction model include:\n1. CN(C)Cc1ccc(-c2cncc(Br)c2)nc1 (score: 0.0000)\n2. CN(C)Cc1cncc(-c2cncc(Br)c2)c1 (score: -0.9756)\n3. CN(C)Cc1ccc(-c2cncc(Br)c2)cn1 (score: -1.2200)\n4. CN(C)Cc1csc(-c2cncc(Br)c2)c1 (score: -1.2601)\n5. CN(C)Cc1ccc(-c2cncc(Br)c2)cc1 (score: -1.3399)\n\nThe highest scoring candidate is CN(C)Cc1ccc(-c2cncc(Br)c2)nc1, which shows a brominated product where the bromine atom has been added to the aromatic ring. This is consistent with a bromination reaction mechanism.\n\nSMILES: CN(C)Cc1ccc(-c2cncc(Br)c2)nc1"

Figure 6: Reasoning despite unsuccessful Retro verification and missing Static Reasoner prediction. Mechanistic analysis, reaction type prediction, and candidate proposals are provided, and the final prediction is correct.

User
Reaction input: CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-].[C-]#[N+]CS(=O)(=O)c1ccc(C)cc1.O=C([O-])[O-].[K+].[K+].CO\n\n
Predict the possible product(s) of this reaction.

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n", "output": "<tool_call>\n{\"name\": \"predict_reaction_type\", \"arguments\": {\"reactants\": \"CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-].[C-]#[N+]CS(=O)(=O)c1ccc(C)cc1.O=C([O-])[O-].[K+].[K+].CO\"}}\n</tool_call>

User
<tool_response>\nPredicted reaction type: Nitro to amino\nConfidence: 0.9707\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"predict_reaction_products\", \"arguments\": {\"reactants\": \"CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-].[C-]#[N+]CS(=O)(=O)c1ccc(C)cc1.O=C([O-])[O-].[K+].[K+].CO\", \"top_k\": 5}}\n</tool_call>

User
<tool_response>\nTop-5 reaction predictions for reactants: CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-].[C-]#[N+]CS(=O)(=O)c1ccc(C)cc1.O=C([O-])[O-].[K+].[K+].CO\n 1. CCOC(=O)N(Cc1ccc(C)nc1)c1cc(-c2cnco2)nc(N)c1[N+](=O)[O-] \n 2. CCOC(=O)N(Cc1ccc(C)nc1)c1cc(-c2cnc[nH]2)nc(N)c1[N+](=O)[O-] \n 3. CCOC(=O)N(Cc1ccc(OC)nc1)c1cc(-c2cnco2)nc(N)c1[N+](=O)[O-] \n 4. CCOC(=O)N(Cc1ccc(C)nc1)c1cc(-c2cnco2)nc2nc[nH]c12\n 5. Cc1ccc(CNc2cc(-c3cnco3)nc(N)c2[N+](=O)[O-])cn1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"query_reasoning_agent\", \"arguments\": {\"reactants\": \"CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-].[C-]#[N+]CS(=O)(=O)c1ccc(C)cc1.O=C([O-])[O-].[K+].[K+].CO\", \"prompt\": \"This is a nitro to amino reduction reaction\"}}\n</tool_call>

User
<tool_response>\n=== Reasoning Agent Chain-of-Thought ===\nOkay, I will follow a structured approach to determine the reaction products. Here is a step-by-step analysis. {Analyzing the given reactants and reagents} The reactants and reagents include CCOC(=O)N(Cc1ccc(C)nc1)c1cc(C=O)nc(N)c1[N+](=O)[O-] (a substituted aromatic amide with a nitroso group), [C-]#[N+]CS(=O)(=O)c1ccc(C)cc1 (a thiocyanate reagent), O=C([O-])[O-] (peroxide), and [K+].[K+] (potassium ions). The solvent is CO (carbon monoxide). {Identifying the key functional groups and reagents} The key functional groups in the reactant are the nitroso group (-N=O) and the amide group (-CONH-). The reagents include a thiocyanate reagent (SCN-), which is a common reagent for converting nitroso groups to thioureas. The reaction is likely to proceed via a nucleophilic attack of the thiocyanate on the nitroso group, followed by the formation of a thiourea derivative. {Predicting the possible reaction mechanism} The reaction mechanism can be broken down into the following st...(truncated)...he intermediate then loses a water molecule to form the thiourea product. The amide group (-CONH-) remains unchanged, and the following steps: 1. Nucleophilic attack: The thiocyanate ion (SCN-) attacks the nitroso group, forming a tetrah intermediate. 2. Proton transfer: A proton is transferred from the nitrogen to the oxygen, forming a thiourea intermediate. 3. Elimination of water: The intermediate loses a water molecule, forming the final thiourea product. {Considering alternative possibilities} 1. The thiocyanate ion could attack the carbonyl group of the amide, but this is less likely due to the lower reactivity of the amide group compared to the nitroso group. 2. The thiocyanate ion could attack the aromatic ring, but this is highly unlikely under the given conditions. 3. The thiocyanate ion could attack the nitrogen of the amide, but this is less likely due to the higher reactivity of the nitroso group.\n===\nReasoning agent did not produce a valid SMILES prediction.\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\nBased on the reasoning agent's analysis, the most likely reaction pathway involves the thiocyanate ion (SCN-) attacking the nitroso group (-N=O) in the aromatic amide. This leads to the formation of a thiourea derivative. The amide group remains intact, and the nitroso group is converted to a thiourea group.\n\nThe key steps in the reaction mechanism are:\n1. Nucleophilic attack of the thiocyanate ion on the nitroso group\n2. Proton transfer forming a thiourea intermediate\n3. Elimination of water to form the final thiourea product\n\nGiven these mechanistic insights, the most reasonable product is:\n\nSMILES: CCOC(=O)N(Cc1ccc(C)nc1)c1cc(-c2cnc[nH]2)nc(N)c1[N+](=O)[O-]

Figure 7: Reasoning without verification-based self-reflection. Mechanistic analysis, reaction type prediction, and candidate proposals are provided, the correct candidate is included, and the final prediction is correct.

User
Reaction input: [BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\n\nPredict the possible product(s) of this reaction.

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n", "output": "<tool_call>\n{\"name\": \"predict_reaction_type\", \"arguments\": {\"reactants\": \"[BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\"}}\n</tool_call>

User
<tool_response>\nPredicted reaction type: CO2H-Me deprotection\nConfidence: 0.9988\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"predict_reaction_products\", \"arguments\": {\"reactants\": \"[BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\", \"top_k\": 5}}\n</tool_call>

User
<tool_response>\nTop-5 reaction predictions for reactants: [BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\n 1. COC(=O)C1=C(C)Nc2ccnn2C1c1ccccc1C(F)(F)F \n 2. COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F \n 3. COC(=O)C1=C(c2ccccc2C(F)(F)F)N2N=CC=C2NC1C \n 4. COC(=O)C1=C(c2ccccc2C(F)(F)F)N2N=CC2=NC1C \n 5. COC(=O)C1=C(c2ccccc2C(F)(F)F)N2N=CCC2NC1C \n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"validate_reaction_pathway\", \"arguments\": {\"reactants\": \"[BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\", \"product\": \"COC(=O)C1=C(C)Nc2ccnn2C1c1ccccc1C(F)(F)F\"}}\n</tool_call>

User
<tool_response>\nFound 1 retrosynthetic pathway(s) for reactants → COC(=O)C1=C(C)Nc2ccnn2C1c1ccccc1C(F)(F)F:\n\nPathway 1 (score: 1.0000):\n Step 1: retrosynthesis_path → COC(=O)C1=C(C)Nc2ccnn2C1c1ccccc1C(F)(F)F>1.0000>COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"query_reasoning_agent\", \"arguments\": {\"reactants\": \"[BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\", \"prompt\": \"This is a CO2H-Me deprotection reaction\"}}\n</tool_call>

User
<tool_response>\n=== Reasoning Agent Chain-of-Thought ===\nOkay, I will follow a structured approach to determine the reaction product. Here is a step-by-step analysis. {Analyzing the given reactants and products} The reactants include [Li+], a lithium ion, [OH-], a hydroxide ion, and C1CCOC1, a cyclic ether (tetrahydrofuran). {Identifying the reaction type} The presence of [OH-] and [Li+] suggests a base-catalyzed reaction. The ester group (COC(=O)) in the reactant is a common functional group that can undergo hydrolysis in the presence of a base, leading to the formation of a carboxylic acid.{Hypothesizing the possible reaction mechanism} The reaction is likely a base-catalyzed ester hydrolysis. The hydroxide ion ([OH-]) will attack the carbonyl carbon of the ester, leading to the formation of a tetrahedral intermediate. This intermediate will then lose a proton to f...(truncated)... carboxylic acid due to the presence of the hydroxide ion. {Rechecking the reaction pathway} To recheck, let's consider the energy profile of the reaction. The formation of the tetrahedral intermediate is energetically favorable due to the strong nucleophilicity of the hydroxide ion. The subsequent proton transfer and elimination of the leaving group (chloride ion) are also energetically favorable, leading to the formation of the carboxylic acid. The presence of the lithium ion helps to stabilize the negatively charged intermediate, making the reaction more favorable. {Final reasoning result} Based on the above analysis, the reaction product is O=C(O)c1ccc(C(F)(F)F)c(-c2ccccc2C(F)(F)F)c1. The ester group (COC(=O)) has been hydrolyzed to form a carboxylic acid group (C(=O)O).\n</think>\n\nUnder the stated conditions, the expected products of the reaction are <chem>O=C(O)c1ccc(C(F)(F)F)c(-c2ccccc2C(F)(F)F)c1</chem>\n===\nReasoning Agent Prediction: SMILES: O=C(O)c1ccc(C(F)(F)F)c(-c2ccccc2C(F)(F)F)c1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\n<tool_call>\n{\"name\": \"validate_reaction_pathway\", \"arguments\": {\"reactants\": \"[BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\", \"product\": \"O=C(O)c1ccc(C(F)(F)F)c(-c2ccccc2C(F)(F)F)c1\"}}\n</tool_call>

User
<tool_response>\nNo retrosynthetic pathways found connecting\n reactants: [BH4-].[Na+].COC(=O)c1c(C)nc2ccnn2c1-c1ccccc1C(F)(F)F.O.CO\n product: O=C(O)c1ccc(C(F)(F)F)c(-c2ccccc2C(F)(F)F)c1\n</tool_response>\n

Assistant (Reaction-Mind)
<think>\n\n</think>\n\nBased on the available evidence, the most chemically reasonable product is the one validated by the retrosynthetic pathway and supported by the reaction type prediction. The product SMILES is:\n\nSMILES: COC(=O)C1=C(C)Nc2ccnn2C1c1ccccc1C(F)(F)F

Figure 8: Full multi-tool collaboration with two-stage self-reflection. The agent sequentially verifies predictions from both Specific Predictor and Static Reasoner before confirming the final product.