SOLVENT: LONG-HORIZON REASONING FOR MULTI-STEP RETROSYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-step retrosynthesis requires recursively transforming a target molecule through successive reaction steps into purchasable building blocks, and recent work has begun to draw on large language model (LLM) reasoning to tackle the task. In its latest evolution, LLM reasoning has entered the long-horizon regime: models pursue objectives across interdependent steps, interact with environments, and adapt to feedback. This shift has markedly advanced LLM capabilities, as illustrated by breakthroughs in AI-driven mathematical research. At this frontier, we introduce **Solvent**, an 8B LLM that successfully extends long-horizon reasoning to multi-step retrosynthesis. Through a two-round post-training curriculum, Solvent learns to interleave iterative synthesis planning, external availability verification, and adaptive route revision. Experiments show that, following a $2.1\times$ performance boost from RL, Solvent scores 61.86% across four benchmarks, surpassing mid-scale language models and prior search-based retrosynthesis systems, and rivaling frontier trillion-parameter LLMs.

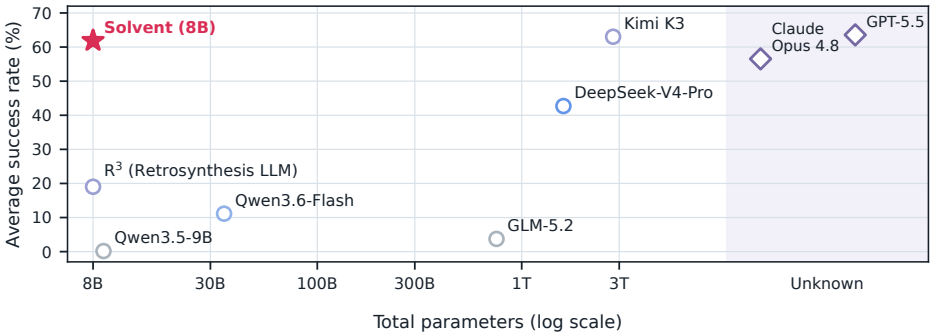


Figure 1: At 8B parameters, Solvent substantially outperforms mid-scale LLMs and rivals frontier trillion-parameter models. Proprietary models with undisclosed parameter counts are arbitrarily positioned in the shaded region.

1 INTRODUCTION

Retrosynthetic planning plays a central role in organic synthesis and drug discovery (Corey, 1991; Zheng et al., 2022) by working backward from a complex target molecule to identify simpler, readily accessible precursors from which the target can be synthesized. This task is studied at two levels of granularity. Single-step retrosynthesis predicts the immediate reactants capable of producing a specified product through one chemical transformation (Jiang et al., 2023). Multi-step planning further requires a sequence of interdependent decisions that recursively decomposes the target into commercially available building blocks (Zheng et al., 2022; Segler et al., 2018; Chen et al., 2020).

Traditional AI-based multi-step retrosynthesis casts route planning as an explicit search built on single-step reaction proposals. Progress has come from improving the local proposal model itself and from learning value functions that prioritize expansions in search algorithms (Segler et al., 2018; Chen et al., 2020; Kim et al., 2021; Hong et al., 2023). This decomposition has made large-scale synthesis planning practical, but it also ties route recovery to the coverage of the single-step model,

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.

incurs substantial search costs as routes deepen, and leaves both the global chemical analysis of the target and the rationale behind individual bond disconnections implicit in the search procedure.

Recently, large language models (LLMs) have evolved from single-turn chatbots into long-horizon agents, demonstrating remarkable capabilities in mathematics, coding, and scientific problem solving through persistent iteration across reasoning, verifying, and revising (Dong et al., 2026; OpenAI, 2026b; Alpöge & Furman, 2026). However, this paradigm remains underexplored in multi-step retrosynthesis, a task naturally suited to long-horizon reasoning, with route construction involving recursive subproblems, precursor checking, and feedback-guided revision. Existing approaches refine LLM-generated routes through external optimization or search (Wang et al., 2025; Song et al., 2026), guide planning over reaction candidates supplied by single-step models (Liu et al., 2025; Liao et al., 2026; Zhu et al., 2026), or generate a complete route directly in one turn (Wang et al., 2026). Across these designs, the LLM's reasoning capacity remains constrained.

We introduce **Solvent**, an 8B LLM that frames multi-step retrosynthesis as a long-horizon task, solving it through iterative planning, external verification, and adaptive revision. Solvent jointly reasons over global strategy and stepwise disconnections, checks precursor availability against a building-block inventory, and updates proposed routes accordingly, keeping all retrosynthetic decisions within the model. Achieving this requires retrosynthetic knowledge and the capacity for continual interaction with the environment, which we develop in Solvent through a two-round, multistage post-training curriculum. The first round develops a seed policy that incorporates retrosynthetic knowledge and exhibits a stable pattern of long-horizon reasoning; the second bootstraps a stronger starting policy, preparing the model for final RL that boosts its capabilities. The resulting model shows strategic chemical insight, devising protecting-group schemes and convergent routes where appropriate.

Across four benchmarks, Solvent achieves leading performance and demonstrates progressive self-improvement through long-horizon interaction. Among the evaluated LLMs, Solvent reaches an average success rate of 61.86%, rivaling the strongest frontier models while surpassing the similarly sized, retrosynthesis-specialized LLM R^3 by 42.83 percentage points. Compared with prior search-based retrosynthesis systems, Solvent reaches an average success rate of 93.22% at a search budget of 100, achieving the highest success rates across the budget range. As the interaction horizon expands, Solvent's average success rate rises from 18.75% to 61.86%, demonstrating the benefit that sustained feedback-based revision provides for Solvent.

Our main contributions are:

- **Framing Multi-Step Retrosynthesis as a Long-Horizon Task.** Solvent addresses the problem through iterative planning, external verification, and adaptive revision, keeping all retrosynthetic decisions inside the model.
- **Developing Long-Horizon Reasoning through Post-Training.** Our two-round, multi-stage curriculum builds Solvent's domain knowledge and capacity to reason across turns, interpret environmental feedback, and make adaptive revisions.
- **Establishing Strong Performance at 8B Parameters.** Solvent rivals frontier LLMs, substantially surpasses similarly sized LLMs, and consistently outperforms prior search-based retrosynthesis systems across the search-budget range.

2 Related Work

2.1 Long-Horizon Tasks and Agents

Long-horizon tasks are characterized by dependencies among decisions: reaching a goal requires multiple steps, with earlier results shaping later choices (Dong et al., 2026). Recent work examines such tasks in software engineering (Kwa et al., 2025), web navigation (Erdogan et al., 2025), and realistic workplace assistance (Xu et al., 2025).

Long-horizon agents are designed to solve such tasks through sustained interaction with an environment (Dong et al., 2026). Recent work has studied this capability across code generation, tool-assisted reasoning, and software engineering, using execution or task feedback to guide subsequent steps (Gehring et al., 2025; Feng et al., 2026; OpenAI, 2025a).

2.2 Large Language Models for Multi-Step Retrosynthesis

Recent studies have begun to harness LLMs for multi-step retrosynthetic planning. One line of work takes a search-based approach. These methods either integrate LLM-generated routes or disconnections into external search procedures, or use LLMs to coordinate search over precursor candidates supplied by external single-step models. LLM-Syn-Planner improves a population of complete routes through evolutionary search (Wang et al., 2025). AOT* maps complete pathways proposed by an LLM into an AND–OR tree and applies explicit tree search to select unsolved intermediates (Song et al., 2026). DeepRetro combines an external recursive search procedure driven by a template-based or MCTS-based retrosynthesis solver with an LLM that proposes and validates single-step disconnections (Sathyanarayana et al., 2026). Retro-R1 uses an LLM to select among precursor sets proposed by an external single-step model (Liu et al., 2025); RetrOrchestrator uses an LLM to choose the molecule to expand, the single-step model, and the requested number of predictions (Liao et al., 2026); and RetroAgent uses an LLM to select graph expansions while an external environment maintains the AND–OR graph and supplies single-step reaction candidates (Zhu et al., 2026). In these methods, route exploration or precursor generation is handled outside the LLM.

A separate approach lets the LLM generate a complete route directly. R^3 produces a complete retrosynthetic route in a single open-loop turn (Wang et al., 2026).

Solvent instead frames multi-step retrosynthesis as a long-horizon task in which the LLM carries out global planning, stepwise decomposition, inventory checking, and feedback-guided revision over multiple turns. More related work is discussed in Appendix A.

3 Preliminaries

Multi-step retrosynthesis. Let $\mathcal{M}$ denote the molecular space and $\mathcal{B} \subset \mathcal{M}$ the set of available building blocks. Given a target molecule $x \in \mathcal{M} \setminus \mathcal{B}$, multi-step retrosynthesis seeks a rooted, acyclic reaction tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ whose root is x (Chen et al., 2020; Liu et al., 2025; Wang et al., 2026). Each internal molecule $m \in \mathcal{V}$ is decomposed by a reaction $e : m \Rightarrow \{m_1, \ldots, m_k\}$, and the leaves $\mathcal{L}(\mathcal{T})$ are the terminal precursors. For simplicity, we call a route *stock-complete* if

$$\mathrm{root}(\mathcal{T}) = x, \qquad \mathcal{L}(\mathcal{T}) \subseteq \mathcal{B}. \tag{1}$$

A stock-complete solution is then further assessed for the plausibility of each reaction. Further preliminaries on agentic reinforcement learning is provided in Appendix B.

4 Methodology

We frame multi-step retrosynthesis as a long-horizon task. Over multiple turns, Solvent iteratively plans a route, checks precursor availability against a building-block inventory, and revises the route in response to inventory feedback. Unlocking the model's full potential through agentic RL requires a strong starting point with clear, stable learning signals, motivating our two-round training pipeline: (1) a first round that develops a seed policy carrying retrosynthetic knowledge and a stable pattern of long-horizon reasoning, and (2) a second round that uses rejection-sampling fine-tuning to bootstrap a stronger starting policy and finally applies RL to boost long-horizon reasoning.

4.1 Developing a Seed Policy for Long-Horizon Reasoning

We first equip the backbone with retrosynthetic knowledge, then cold-start long-horizon reasoning over multiple turns, and finally stabilize this behavior with RL.

Building retrosynthetic intuition. We use a teacher model to distill retrosynthetic knowledge from reference routes into structured analyses, covering target-level strategy and every reaction step. We retain analyses that faithfully preserve the original reaction routes. Throughout post-training, we apply a decontamination procedure to training corpora to remove overlap with evaluation benchmarks. The retained data remain structurally diverse and well separated from benchmark targets at both fingerprint and scaffold levels (Appendix C.1). We fine-tune the retrosynthesis-pretrained backbone on the corpus, yielding the warmed-up policy π_{warmed} with retrosynthetic intuition.

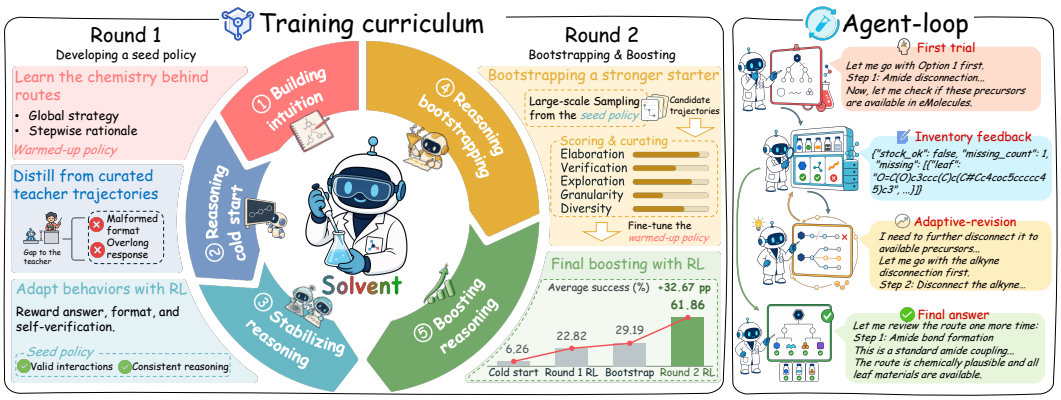


Figure 2: **Overview of Solvent. Left:** Two rounds of post-training prepare Solvent for effective RL. Round 1 develops a seed policy with retrosynthetic knowledge and stable reasoning. Round 2 bootstraps a stronger starting policy using the seed policy, enabling final-stage RL to substantially boost performance. **Right:** Solvent performs long-horizon reasoning through iterative planning, external verification, and adaptive revision.

Teacher-guided cold start for long-horizon reasoning. We formulate multi-step retrosynthesis as a long-horizon reasoning task. Given a target molecule, the model develops a global route strategy, makes stepwise disconnection decisions, checks proposed precursors against a building-block inventory, interprets availability feedback, and adapts subsequent decisions over multiple turns. For a target molecule x, let $q(x)$ denote the task prompt and $h_t = ((r_i, a_i, o_i))_{i<t}$ the history of reasoning r_i, actions a_i, and inventory observations o_i from preceding turns. At turn t, the policy samples:

$$(r_t, a_t) \sim \pi_\theta(\cdot \mid q(x), h_t)\,, \tag{2}$$

where r_t denotes the reasoning content and a_t either queries the inventory with a proposed route or returns the final route. The observation o_i reports the availability of the submitted route's terminal precursors. We sample multi-turn trajectories from a teacher model, retaining only those that yield valid solutions, remain consistent with the recorded inventory interactions, and pass the chemical-validity assessment. After strictly removing overlaps with the evaluation benchmarks, we fine-tune π_{warmed} on the curated trajectories, establishing an initial policy for iterative route construction and revision; data-construction and training details appear in Appendices D.2 and E.2.

Consolidating long-horizon reasoning behavior with RL. Teacher-guided fine-tuning establishes a basis for long-horizon reasoning. However, the capability gap between the teacher model and the student policy can lead to unstable behavior, including truncated responses and malformed inventory queries. To close this gap, we optimize the policy with Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025), consolidating its own successful behaviors into stable reasoning patterns. We draw structurally challenging targets with broad molecular diversity from USPTO-full (Dai et al., 2019), exclude all instances that overlap with the evaluation benchmarks, and apply the following trajectory-level reward:

$$R_i = 0.6r_i^{\text{sol}} + 0.2r_i^{\text{verify}} + 0.2r_i^{\text{format}}. \tag{3}$$

The solution term r_i^{sol} is the binary output of the fixed route verifier used consistently to define task success throughout training and evaluation. It accepts only nontrivial routes that are structurally valid, stock-complete, and satisfy the chemical-validity criteria. The other terms reward inventory verification and format adherence. We denote the resulting policy by π_{seed} and use it to generate trajectories for second-round bootstrapping; details of target selection and RL training are provided in Appendices D.3 and E.3.

4.2 Bootstrapping and Boosting Long-Horizon Reasoning

We then bootstrap a stronger starting policy through rejection-sampling fine-tuning (Yuan et al., 2023), and finally apply RL to elicit a more refined mode of long-horizon reasoning.

4.2.1 REASONING BOOTSTRAPPING

Effective RL benefits from an initialization that balances task performance with exploratory capacity (Li et al., 2026), and we obtain such a policy through reasoning bootstrapping (Zelikman et al., 2022): we sample trajectories from π_{seed} and fine-tune π_{warmed} on a curated subset of high-quality, diverse trajectories.

Sampling candidate trajectories. We assemble a structurally diverse set of challenging targets from USPTO-full (Dai et al., 2019). After strictly removing overlaps with the evaluation benchmarks, we use π_{seed} to sample multiple trajectories per target, forming the candidate pool.

Curating for reasoning quality and diversity. We first retain only those trajectories that yield a stock-complete route, remain fully consistent with the recorded inventory interactions, and pass the chemical-validity assessment. Following LIMO (Ye et al., 2025), we then score the remaining trajectories along four complementary axes: reasoning elaboration, self-verification, exploration, and adaptive granularity. These individual scores are further combined with a reasoning-style diversity term and an n-gram repetition penalty, and we retain only the single highest-scoring trajectory for each canonical target. Full filtering and scoring details are provided in Appendix D.4.

Balancing performance and exploration. We fine-tune π_{warmed} on the curated trajectories, carefully tracking validation success and held-out entropy throughout training in order to detect any premature loss of exploration. We find that the first-epoch checkpoint offers a favorable exploration–exploitation trade-off: it substantially improves retrosynthesis performance over π_{seed} while still avoiding the sharp entropy decline observed with further training (Appendix F.1.2). The resulting policy, π_{boot}, subsequently serves as the initialization for the final stage of RL.

4.2.2 BOOSTING LONG-HORIZON REASONING WITH RL

Finally, we optimize π_{boot} with GSPO (Zheng et al., 2025) to further enhance the policy's long-horizon reasoning. For final-stage RL, we curate a more challenging and structurally diverse target set from USPTO-full (Dai et al., 2019), with all overlaps with benchmarks removed. To prioritize route success, we gate r_i^{verify} and r_i^{format} on r_i^{sol}:

$$\begin{aligned} \bar{r}_i^k &= \begin{cases} r_i^k, & r_i^{\text{sol}} = 1, \\ 0, & \text{otherwise}, \end{cases} \qquad k \in \{\text{verify}, \text{format}\}, \\ R_i^{\text{final}} &= 0.8 r_i^{\text{sol}} + 0.1 \bar{r}_i^{\text{verify}} + 0.1 \bar{r}_i^{\text{format}}. \end{aligned} \tag{4}$$

The reward components retain the definitions from Eq. 3, and $\bar{r}_i^k$ denotes the gated form of the two auxiliary terms. By scoring complete trajectories, the objective assigns credit to route construction, inventory interaction, and feedback-guided revision when they jointly produce a successful route. The complete RL configuration is detailed in Appendix E.5.

5 EXPERIMENTS

Solvent achieves leading results across four widely used multi-step retrosynthesis benchmarks. Among the evaluated LLMs, Solvent rivals frontier models while substantially outperforming similarly-sized ones. Against prior search-based retrosynthesis systems, Solvent records the highest success rates across the search-budget range. Stage-wise analyses trace how post-training progressively unlocks Solvent's long-horizon reasoning. Evaluation settings, additional ablations, and representative cases appear in Appendices C, F.1, and G.

5.1 EXPERIMENTAL SETUP

Benchmarks. We use four widely adopted multi-step retrosynthesis benchmarks. USPTO-190 contains 190 patent-derived targets (Chen et al., 2020). ChEMBL-1000 comprises 1,000 structurally complex compounds sampled from the ChEMBL bioactivity database (Liu et al., 2023; Gaulton et al., 2012). Pistachio Reachable contains 150 targets with reference routes averaging 5.5 reactions

Table 1: **At 8B parameters, Solvent remains competitive with frontier LLMs across four multi-step retrosynthesis benchmarks.** We report parameter counts, benchmark-specific success rates, and their average. To enable cost-controlled, target-matched evaluation, selected models are evaluated on a shared 400-target subset and marked with †. Em dashes denote undisclosed parameter counts. Gray shading and boldface mark the best frontier and mid-scale results, respectively.

Method	Params.	Pistachio Hard	Pistachio Reachable	USPTO-190	ChEMBL-1000	Average (%)
Frontier LLMs (proprietary)						
Claude Opus 4.8†	—	38.57	77.50	54.44	55.63	56.54
Gemini 3.5 Flash†	—	42.86	72.50	52.22	65.63	58.30
Grok 4.5†	—	52.86	71.25	50.00	62.50	59.15
GPT-5.5†	—	58.57	82.50	54.44	58.75	63.57
Frontier LLMs (open-weight)						
GLM-5.2†	753 B	1.43	6.25	2.22	5.00	3.73
Kimi K2.5†	1 T	27.14	36.25	25.56	19.38	27.08
DeepSeek-V4-Pro	1.6 T	38.00	54.00	36.32	42.50	42.70
Kimi K3†	2.8 T	54.29	80.00	52.22	65.63	63.03
Mid-scale LLMs						
Qwen3-8B	8 B	0.00	0.00	0.53	0.10	0.16
Qwen3.5-9B	9 B	0.00	0.67	0.00	0.00	0.17
Qwen3.6-Flash†	35 B	10.00	22.50	8.89	3.13	11.13
R^3	8 B	19.00	26.67	13.16	17.30	19.03
Solvent	8 B	**62.00**	**78.00**	**48.95**	**58.50**	**61.86**

and a mean depth of 5.4, whereas Pistachio Hard contains 100 targets with longer routes averaging 7.5 reactions and a mean depth of 7.2 (Yu et al., 2024). Following standard practice, we use the building-block set from eMolecules as the inventory of commercially available compounds for route evaluation and precursor-availability feedback (Chen et al., 2020; Yu et al., 2024).

Evaluation. We evaluate Solvent under the route-level evaluation protocols established in prior work (Chen et al., 2020; Yu et al., 2024; Wang et al., 2026; Song et al., 2026). Success requires a valid route tree rooted at the target, all leaves in stock, and every reaction to pass the chemical-validity check combining plausibility with atom-source accounting (Appendix C.1).

Baselines. We first compare Solvent with proprietary and open-weight LLMs, including frontier models such as Claude Opus 4.8 (Anthropic, 2026) and Kimi K3 (Kimi Team, 2026b) and mid-scale models such as Qwen3.6-Flash (Qwen Team, 2026b) and the 8B retrosynthesis model R^3 (Wang et al., 2026). We further compare Solvent with prior search-based retrosynthesis systems on USPTO-190, ChEMBL-1000, and the Pistachio benchmarks (Chen et al., 2020; Kim et al., 2021; Hong et al., 2023; Liu et al., 2023; 2025; Wang et al., 2025; Song et al., 2026; Zhong et al., 2022; Segler et al., 2018). Details are provided in Appendix C.2.

Inference protocol. We select Solvent's inference configuration on a held-out set before test evaluation (Appendices F.2.1 and F.2.2). For comparison with LLMs, all models use a common 32k-token output limit and up to five interaction turns, with each model evaluated under its optimal inference configuration. Comparisons with prior search-based retrosynthesis systems use aligned search budgets, with the protocol detailed in Appendix C.2.

5.2 Main Results

We compare Solvent with both LLMs and prior search-based retrosynthesis systems, demonstrating Solvent's strong performance on multi-step retrosynthesis. Among the evaluated LLMs, Solvent performs on par with frontier models and substantially outperforms similarly sized models across the

Table 2: **Comparison with prior search-based retrosynthesis systems across search budgets.** Here, N tracks how much explicit reaction-level construction each method performs before finding a successful route, with the cross-method accounting defined in Appendix C.2. Bold and underlined values denote the best and second-best results, respectively.

Performance on USPTO-190

Method	$N = 50$	$N = 100$	$N = 200$	$N = 300$	$N = 400$	$N = 500$
Greedy DFS	19.47	19.47	19.47	19.47	19.47	19.47
retro*	38.95	50.00	68.95	74.74	77.89	79.47
retro*-0	27.37	37.89	55.79	62.63	72.11	74.21
retro*+	55.79	68.42	79.47	83.16	84.21	84.21
retro*+-0	47.89	61.05	76.84	82.11	85.79	86.32
EG-MCTS	58.42	64.74	69.47	75.26	78.42	81.05
PDVN	64.74	72.11	<u>82.63</u>	<u>87.89</u>	<u>91.58</u>	<u>92.11</u>
RETRO-R1	<u>73.21</u>	<u>77.21</u>	82.58	84.47	85.89	86.95
Solvent	**82.11**	**87.36**	**88.95**	**94.21**	**96.84**	**97.37**

Performance on ChEMBL-1000

Method	$N = 50$	$N = 100$	$N = 200$	$N = 300$	$N = 400$	$N = 500$
Greedy DFS	38.10	38.10	38.10	38.10	38.10	38.10
retro*	65.60	69.00	71.60	73.20	74.00	74.40
retro*-0	64.60	67.20	70.30	71.80	72.70	73.40
retro*+	70.50	73.90	76.20	78.00	78.80	79.40
retro*+-0	69.40	73.60	75.60	76.70	77.60	78.50
EG-MCTS	71.20	74.20	76.90	78.30	79.00	79.30
PDVN	73.90	76.40	78.10	79.60	80.30	80.60
RETRO-R1	<u>77.88</u>	<u>80.21</u>	<u>81.79</u>	<u>82.62</u>	<u>83.28</u>	<u>83.72</u>
Solvent	**87.10**	**93.20**	**95.90**	**96.70**	**97.30**	**98.20**

Performance on Pistachio Reachable & Hard

Method	Pistachio Reachable			Pistachio Hard		
	$N = 100$	$N = 300$	$N = 500$	$N = 100$	$N = 300$	$N = 500$
LLM (MCTS)	12.70	17.30	20.70	0.00	4.00	5.00
LLM (Retro*)	14.70	19.30	13.30	0.00	2.00	5.00
LLM-Syn-Planner	93.30	98.00	98.00	72.00	86.00	87.00
AOT*	98.70	<u>99.30</u>	<u>99.30</u>	<u>86.00</u>	<u>89.00</u>	<u>93.00</u>
RootAligned (MCTS)	<u>99.30</u>	<u>99.30</u>	<u>99.30</u>	83.00	85.00	85.00
RootAligned (Retro*)	98.70	98.70	98.70	78.00	82.00	82.00
Solvent	**99.33**	**100.00**	**100.00**	**93.00**	**98.00**	**98.00**

four benchmarks. Against prior search-based retrosynthesis systems, Solvent achieves the highest success rates across the search-budget range.

At 8B parameters, Solvent performs competitively with frontier LLMs on multi-step retrosynthesis. Table 1 compares Solvent with leading proprietary and open-weight LLMs. Given limited evaluation budgets, we follow established practice by evaluating selected models on a common subset (OpenAI, 2025b; Anthropic, 2025), with details provided in Appendix C.2. Solvent averages 61.86% on the full test sets, competing with frontier LLMs while substantially outperforming similarly sized open-weight models. For a direct comparison, we also report the results for all evaluated LLMs on the shared subset (Appendix C.2). Appendix G.1 shows how Solvent revises routes using inventory feedback, while Appendix F.2.4 reports further assessments of route quality.

Solvent leads prior search-based retrosynthesis systems across search budgets. Solvent achieves the highest success rates on all four benchmarks throughout the budget range (Table 2). We leverage larger search budgets through parallel sampling, enabling Solvent to explore a broader range of retrosynthetic strategies; Appendix G.2 compares routes produced in independent trials.

Table 3: **Stage-wise performance across the two-round post-training curriculum.** We report performance for four training stages described in Sec. 4.1 and Sec. 4.2. Average success improves across the reported stages, with the largest gain occurring after final-stage RL.

Training stage		Pistachio Hard	Pistachio Reachable	USPTO-190	ChEMBL-1000	Average
Initial Backbone		0.00	0.00	0.53	0.20	0.18
Round 1	Teacher-guided cold start	2.00	12.67	3.16	7.20	6.26
	First-round RL	19.00	40.00	9.47	22.80	22.82
Round 2	Reasoning bootstrapping	23.00	41.33	22.63	29.80	29.19
	Final-stage RL	**62.00**	**78.00**	**48.95**	**58.50**	**61.86**

5.3 Ablation Study

Successive post-training stages progressively unlock Solvent's long-horizon reasoning, with final-stage RL accounting for the largest gain. Table 3 compares the initial backbone with cumulative stage-wise checkpoints across the four benchmarks. Average success increases at every stage, from 0.18% to 61.86%.

The first training round develops a seed policy that exhibits a stable pattern of long-horizon reasoning. After teacher-guided cold start, average success increases from 0.18% to 6.26%, while truncated responses and malformed inventory queries remain frequent. Round 1 RL substantially stabilizes the model's response patterns, raising success to 22.82%. Appendix G.3.1 compares the model's reasoning on the same target before and after Round 1 RL.

Bootstrapping consolidates reasoning patterns and preserves exploration, setting the stage for capability emergence in final-stage RL. Round 2 bootstrapping produces a stronger starting point for final-stage RL, reaching 29.19% average success after fine-tuning on curated trajectories. Appendix F.1.3 further examines the role of bootstrapping. Final-stage RL yields the largest improvement, raising success from 29.19% to 61.86%.

6 Analysis

We analyze the emergence of Solvent's long-horizon capabilities throughout post-training. We first examine how post-training enhances Solvent's ability to revise routes adaptively. We then examine how final-stage RL progressively improves Solvent's long-horizon reasoning.

6.1 The Capacity for Adaptive Revision Emerges across Post-Training

Solvent progressively learns to use feedback for revision during post-training. To examine this, we evaluate Solvent across post-training stages with turn limits of two to five on all four benchmarks (Figure 3(a)). Figure 3(b) reports the success gain from an additional turn at each post-training stage as the ordinary least-squares slope $S(t)$, in percentage points per turn (pp/turn). The results show that, over the course of post-training, Solvent learns to improve its proposals through environmental feedback. Appendix G.1.1 provides a case study of self-revision.

6.2 How Final-Stage RL Improves Long-Horizon Route Construction

Final-stage RL progressively improves chemical validity and stock completion during long-horizon route construction. Validation success rises throughout final-stage RL (Figure 4(a)). Chemical validity improves first, followed by gains in stock completion (Figure 4(b)), indicating that final-stage RL first improves individual disconnection decisions and then integrates them into stock-complete routes. The mean response length remains stable (Figure 4(c)). After final-stage RL, Solvent exhibits the capacity for strategic synthesis planning, including protecting-group use and convergent route design (Appendix G.1). Appendix G.3.2 presents a case study comparing chemical strategies before and after final-stage RL. Further training dynamics appear in Appendix F.2.5.

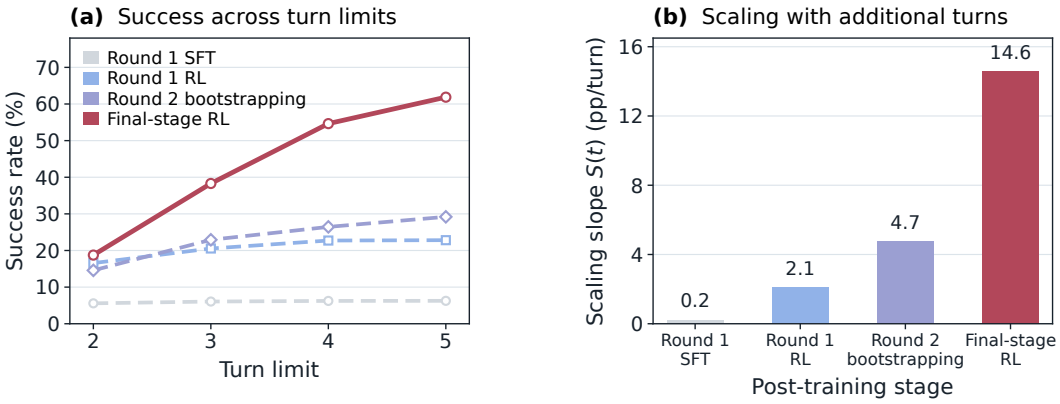


Figure 3: **The benefit of additional interaction turns increases throughout post-training.** (a) Average success rate under turn limits of 2–5. (b) Success gain per additional turn across stages.

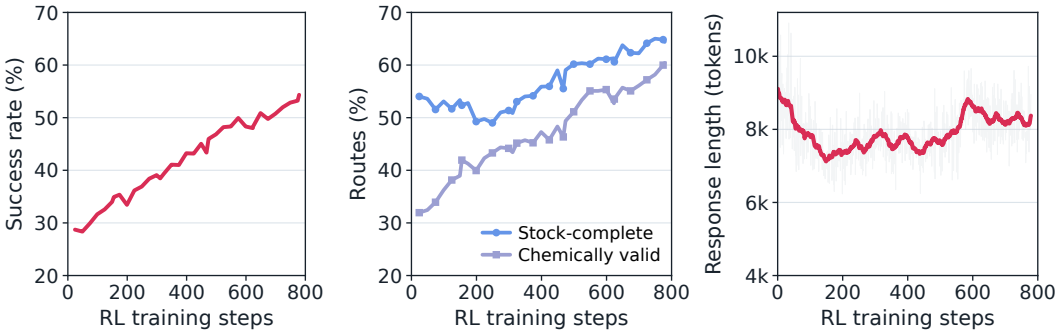


Figure 4: **Final-stage RL successively improves chemical validity and stock completion.** (a) Performance on validation set during final-stage RL. (b) Evolution of chemical validity and stock completion. (c) Mean response length during training.

7 Limitations and Future Directions

Despite the promising results, reliable route verification remains an open challenge in multi-step retrosynthesis. In silico checks can serve as proxies for reaction feasibility, but cannot establish experimental success, yield, selectivity, reproducibility, or operational safety.

Future work could connect long-horizon LLM agents to automated wet-lab platforms, allowing proposed routes and reaction conditions to be evaluated against measured outcomes and practical laboratory constraints. Experimental results could then guide planning and revision over successive synthesis cycles, allowing long-horizon agents to base future decisions on real experimental results rather than on in silico assessment alone. Further discussion appears in Appendix H.

8 Conclusion

In this paper, we present Solvent, an 8B LLM that frames multi-step retrosynthesis as a long-horizon task. Solvent iteratively plans routes, checks precursor availability, and revises its decisions based on inventory feedback. A two-round post-training curriculum develops this capability. Across four benchmarks, Solvent rivals frontier LLMs and consistently outperforms prior search-based retrosynthesis systems. Case studies further show Solvent exhibiting strategic chemical reasoning, including protecting-group schemes and convergent routes where appropriate, highlighting Solvent's chemical insight and the interpretability of its proposals. Together, Solvent offers a promising long-horizon paradigm for multi-step retrosynthesis and a path to further scaling.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic training data, assisting with coding, and polishing the manuscript. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results; tasks involving mathematical proofs are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research develops an LLM-based method for multi-step retrosynthesis and does not involve human subjects or sensitive personal data. All experiments use established chemistry datasets and benchmarks widely adopted in retrosynthesis research. The proposed method does not involve demographic attributes or decisions about individuals and therefore raises no direct concerns related to discrimination or human-subject bias. The authors declare no conflicts of interest or competing financial interests related to this work. The research follows established practices in machine learning and computational chemistry.

REPRODUCIBILITY STATEMENT

Appendix C documents the benchmarks, inventory processing, evaluation criteria, baseline configurations, subset construction, and planner-budget accounting. Appendix D details the training data sources, prompts, filtering procedures, and target curricula, while Appendix E provides the complete post-training configurations. Additional ablations and analyses are reported in Appendix F, and representative trajectories and their chemical review are provided in Appendix G. We will include training and evaluation scripts and configuration files in the supplementary material and, upon acceptance, publicly release the model weights together with the complete training and evaluation codebase.

REFERENCES

Levent Alpöge and Ralph Furman. More than two thirds of the zeta zeros are simple and on the critical line. arXiv preprint arXiv:2608.13637, August 2026.

Anthropic. Claude 3.7 sonnet system card. Anthropic System Card, 2025.

Anthropic. System Card: Claude Opus 4.8. Anthropic System Card, May 2026.

Lei Bai et al. Intern-S1: A scientific multimodal foundation model, 2025.

John Bradshaw, Anji Zhang, Babak Mahjour, David E. Graff, Marwin H. S. Segler, and Connor W. Coley. Challenging reaction prediction models to generalize to novel chemistry. *ACS Central Science*, 11(4):539–549, 2025. doi: 10.1021/acscentsci.5c00055.

Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided a* search. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1608–1616. PMLR, 2020.

Shuan Chen and Yousung Jung. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au*, 1(10):1612–1620, 2021. doi: 10.1021/jacsau.1c00246.

Jonghwi Choe, Shuan Chen, and Yousung Jung. Assessing the extrapolation capability of template-free retrosynthesis models. *Digital Discovery*, 5(6):2591–2599, 2026. doi: 10.1039/D6DD00072J.

Junseok Choe, Hajung Kim, Yan Ting Chok, Mogan Gim, and Jaewoo Kang. Retrosynthetic crosstalk between single-step reaction and multi-step planning. *Journal of Cheminformatics*, 17 (1):130, August 2025. ISSN 1758-2946. doi: 10.1186/s13321-025-01088-z.

Elias James Corey. The logic of chemical synthesis: Multistep synthesis of complex carbogenic molecules (nobel lecture). *Angewandte Chemie International Edition in English*, 30(5):455–465, May 1991. ISSN 0570-0833. doi: 10.1002/anie.199104553.

Hanjun Dai, Chengtao Li, Connor W. Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. arXiv preprint arXiv:2606.19348, 2026.

Guanting Dong, Xiaoshuai Song, Yuyang Hu, Jiajie Jin, Chenghao Zhang, Yifei Chen, Xiaoxi Li, Huaying Yuan, Xinyu Yang, Tongyu Wen, Jiejun Tan, Hongjin Qian, Shijue Huang, Junting Lu, Zhenyu Li, Wanjun Zhong, Yutao Zhu, Tat-Seng Chua, Zhicheng Dou, and Ji-Rong Wen. Towards long-horizon agents: A survey. *Preprints*, July 2026. doi: 10.20944/preprints202607.1328.v1.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15419–15462. PMLR, 13–19 Jul 2025.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, June 2009. ISSN 1758-2946. doi: 10.1186/1758-2946-1-8.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2012. doi: 10.1093/nar/gkr777.

Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Taco Cohen, and Gabriel Synnaeve. RLEF: Grounding code LLMs in execution feedback with reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19034–19055. PMLR, 2025.

Samuel Genheden and Jason D. Shields. A simple similarity metric for comparing synthetic routes. *Digital Discovery*, 4(1):46–53, 2025. ISSN 2635-098X. doi: 10.1039/d4dd00292j.

Google DeepMind. Gemini 3.5 flash model card. Google DeepMind Model Card, 2026.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14852–14882, 2023. doi: 10.18653/v1/2023.acl-long.830.

Siqi Hong, Hankz Hankui Zhuo, Kebing Jin, Guang Shao, and Zhanwen Zhou. Retrosynthetic planning with experience-guided monte carlo tree search. *Communications Chemistry*, 6(1):120, 2023. doi: 10.1038/s42004-023-00911-8.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3 (1):015022, January 2022. ISSN 2632-2153. doi: 10.1088/2632-2153/ac3ffb.

Yinjie Jiang, Yemin Yu, Ming Kong, Yu Mei, Luotian Yuan, Zhengxing Huang, Kun Kuang, Zhihua Wang, Huaxiu Yao, James Zou, Connor W. Coley, and Ying Wei. Artificial intelligence for retrosynthesis prediction. *Engineering*, 25:32–50, June 2023. ISSN 2095-8099. doi: 10.1016/j.eng.2022.04.021.

Jeffrey S. Johnson. Counting steps. *Nature Synthesis*, 2:6–8, 2023. doi: 10.1038/s44160-022-00204-3.

Junsu Kim, Sungsoo Ahn, Hankook Lee, and Jinwoo Shin. Self-improved retrosynthetic planning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5486–5495. PMLR, 2021.

Kimi Team. Kimi K2.5: Visual agentic intelligence, 2026a.

Kimi Team. Kimi K3: Open frontier intelligence, 2026b.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring ai ability to complete long software tasks. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 92213–92266. Curran Associates, Inc., 2025. doi: 10.52202/085713-3086.

Xinran Li, Guangda Huzhang, Siqi Shen, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Jun Zhang. Getting your LLMs ready for reinforcement learning with lightweight SFT. In *The Fourteenth International Conference on Learning Representations*, 2026.

Chang Liao, Luotian Yuan, Yiping Ke, and Ying Wei. RetrOrchestrator: A multi-step retrosynthesis agent dynamically orchestrating single-step transition models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Guoqing Liu, Di Xue, Shufang Xie, Yingce Xia, Austin Tripp, Krzysztof Maziarz, Marwin Segler, Tao Qin, Zongzhang Zhang, and Tie-Yan Liu. Retrosynthetic planning with dual value networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22266–22276. PMLR, 2023.

Wei Liu, Jiangtao Feng, Hongli Yu, Yuxuan Song, Yuqiang Li, Shufei Zhang, Lei Bai, Wei-Ying Ma, and Hao Zhou. Retro-r1: Llm-based agentic retrosynthesis. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 70709–70737. Curran Associates, Inc., 2025. doi: 10.52202/085713-2378.

OpenAI. Introducing codex. OpenAI, May 2025a.

OpenAI. Introducing gpt-5. OpenAI, 2025b.

OpenAI. Introducing GPT-5.5. OpenAI Model Release, 2026a.

OpenAI. Ten advances in mathematics and theoretical computer science. Typeset manuscript, August 2026b.

Sebastian Pagel, Michael Jirasek, and Leroy Cronin. Verification and execution of the scientific literature via chemputation augmented by large language models. *Communications Chemistry*, 9 (1):191, 2026. doi: 10.1038/s42004-026-01993-w.

Qwen Team. Qwen3.5-9B Model Card. Hugging Face Model Card, 2026a.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Official model release blog, April 2026b.

Jihye Roh, Joonyoung F. Joung, Kevin Yu, Zhengkai Tu, G. Logan Bartholomew, Omar A. Santiago-Reyes, Mun Hong Fong, Richmond Sarpong, Sarah E. Reisman, and Connor W. Coley. Higher-level strategies for computer-aided retrosynthesis. *ACS Central Science*, 12(3):345–357, 2026. doi: 10.1021/acscentsci.5c02014.

Shreyas Vinaya Sathyanarayana, Sharanabasava D. Hiremath, Shah Rahil Kirankumar, Rishikesh Panda, Rahul Jana, Riya Singh, Rida Irfan, Ashwin Murali, and Bharath Ramsundar. Deepretro discovers retrosynthetic pathways through iterative large language model reasoning. *Scientific Reports*, 16(1):8448, 2026. doi: 10.1038/s41598-026-38821-z.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018. doi: 10.1038/nature25978.

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

Yu Shee, Anton Morgunov, Haote Li, and Victor S. Batista. Directmultistep: Direct route generation for multistep retrosynthesis. *Journal of Chemical Information and Modeling*, 65(8):3903–3914, 2025. doi: 10.1021/acs.jcim.4c01982.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Xiaozhuang Song, Xuanhao Pan, Xinjian Zhao, Hangting Ye, Shufei Zhang, Jian Tang, and Tianshu Yu. AOT*: Efficient synthesis planning via LLM-empowered AND-OR tree search. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34727–34758. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.findings-acl.1734.

SpaceXAI. Introducing grok 4.5. Official model release, 2026.

Zhengkai Tu, Sourabh J. Choure, Mun Hong Fong, Jihye Roh, Itai Levin, Kevin Yu, Joonyoung F. Joung, Nathan Morgan, Shih-Cheng Li, Xiaoqi Sun, Huiqian Lin, Mark Murnin, Jordan P. Liles, Thomas J. Struble, Michael E. Fortunato, Mengjie Liu, William H. Green, Klavs F. Jensen, and Connor W. Coley. ASKCOS: Open-source, data-driven synthesis planning. *Accounts of Chemical Research*, 58(11):1764–1775, 2025. doi: 10.1021/acs.accounts.5c00155.

Haorui Wang, Jeff Guo, Lingkai Kong, Rampi Ramprasad, Philippe Schwaller, Yuanqi Du, and Chao Zhang. LLM-augmented chemical synthesis and design decision programs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62980–63001. PMLR, 2025.

YiFei Wang, Qizhi Pei, Jiangtao Feng, Yuntian Shi, Yi Duan, Lihao Wang, Lei Bai, Lijun Wu, Wei-Ying Ma, and Hao Zhou. r^3: End-to-end reasoning-based planning for multi-step retrosynthesis via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37618–37632. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.acl-long.1745.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Frank (Fangzheng) Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. Theagentcompany: Benchmarking llm agents on consequential real world tasks. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen

(eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-0315.

An Yang et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *The Second Conference on Language Modeling*, 2025.

Kevin Yu, Jihye Roh, Ziang Li, Wenhao Gao, Runzhong Wang, and Connor W. Coley. Double-ended synthesis planning with goal-constrained bidirectional search. In Amir Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 112919–112949. Curran Associates, Inc., 2024. doi: 10.52202/079017-3588.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023.

Bogdan Zagribelnyy, Ivan Ilin, Maksim Kuznetsov, Nikita Bondarev, Mathieu Reymond, Roman Schutski, Thomas MacDougall, Rim Shayakhmetov, Zulfat Miftakhutdinov, Mikolaj Mizera, Vladimir Aladinskiy, Alex Aliper, and Alex Zhavoronkov. When single answer is not enough: Rethinking single-step retrosynthesis benchmarks for LLMs, 2026.

Z.ai. Glm-5.2: Built for long-horizon tasks. Z.ai Blog, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022.

Yuxuan Zhang, Yue Fang, Haifan Zhou, Bowen Yu, Tsz Fung Fung, Qing Liu, Christophe Len, and Hanyu Gao. ProcedureT5: Adaptive experimental procedure prediction with data-augmented pre-training and multi-source data integration. *Reaction Chemistry & Engineering*, pp. 1584–1598, 2026. doi: 10.1039/D5RE00572H.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Shuangjia Zheng, Tao Zeng, Chengtao Li, Binghong Chen, Connor W. Coley, Yuedong Yang, and Ruibo Wu. Deep learning driven biosynthetic pathways navigation for natural products with bionavi-np. *Nature Communications*, 13(1):3342, Jun 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-30970-9.

Weihe Zhong, Ziduo Yang, and Calvin Yu-Chian Chen. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing. *Nature Communications*, 14(1): 3009, 2023. doi: 10.1038/s41467-023-38851-5.

Zipeng Zhong, Jie Song, Zunlei Feng, Tiantao Liu, Lingxiang Jia, Shaolun Yao, Min Wu, Tingjun Hou, and Mingli Song. Root-aligned SMILES: a tight representation for chemical reaction prediction. *Chemical Science*, 13(31):9023–9034, 2022. doi: 10.1039/D2SC02763A.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209. PMLR, 2024.

Yanqiao Zhu, Jingru Gan, Xiaoqi Sun, Fang Sun, Yidan Shi, Md Mofijul Islam, Chao Shang, Wenhao Gao, Connor W. Coley, Yizhou Sun, and Wei Wang. RetroAgent: Harnessing LLMs to search over structured memory for agentic retrosynthesis planning. In *Proceedings of the Conference on Language Modeling*, 2026.

A Additional Related Work

Single-step retrosynthesis models. Neural retrosynthesis was first developed around models that predict candidate precursors for an individual product. Conditional Graph Logic Networks learn when reaction templates are applicable, whereas LocalRetro combines local reaction templates with global molecular context (Dai et al., 2019; Chen & Jung, 2021). Other approaches reduce dependence on explicit reaction templates through representation design or structured generation. Root-aligned SMILES reduces the sequence discrepancy between products and reactants, while Graph2Edits generates precursors through autoregressive molecular-graph edits (Zhong et al., 2022; 2023). These models provide the local reaction proposals on which many multi-step planners operate.

Classical multi-step retrosynthesis planning. Multi-step planners traditionally combine a single-step model with an explicit search procedure. Early neural approaches used Monte Carlo tree search with learned expansion and filtering policies, while Retro* introduced neural-guided A*-like search over an AND–OR tree (Segler et al., 2018; Chen et al., 2020). Subsequent work improved this paradigm by learning from successful planning experience, replacing rollouts with experience-guided search, and jointly optimizing single-step predictions with route-level value estimates (Kim et al., 2021; Hong et al., 2023; Liu et al., 2023). DESP further introduced goal-constrained bidirectional search for routes that must contain specified starting materials (Yu et al., 2024). Despite their different search objectives and heuristics, these methods construct complete routes by repeatedly selecting and expanding externally proposed single-step reactions. A complementary direction generates entire routes directly. DirectMultiStep uses a task-specific encoder–decoder Transformer to generate a complete multi-step route as a structured sequence (Shee et al., 2025). In Solvent, complete-route generation is embedded in an LLM policy that evaluates and revises its proposals through inventory interaction.

Reasoning, bootstrapping, and tool use. Chain-of-thought prompting established intermediate reasoning traces as an effective interface for complex problem solving (Wei et al., 2022). Fine-tune-CoT transfers such traces from a larger teacher, while STaR bootstraps reasoning from successful model-generated rationales (Ho et al., 2023; Zelikman et al., 2022). Toolformer further trains language models to decide when and how to invoke external APIs (Schick et al., 2023). Complementary work on test-time computation shows that allocating additional inference effort can substantially improve reasoning when it is paired with suitable generation and selection procedures (Snell et al., 2025). These foundations motivate models that sustain long-horizon reasoning while incorporating external evidence into subsequent decisions.

B Agentic Reinforcement Learning

Agentic reinforcement learning optimizes an LLM over complete multi-turn trajectories, allowing later decisions to condition on observations from earlier interactions (Zhou et al., 2024).

Agent–environment interaction. ReAct organizes interactive problem solving as an interleaving of reasoning, actions, and observations, so that information returned by an environment can inform subsequent decisions (Yao et al., 2023). For a task input x, let h_t denote the interaction history before turn t, and let r_t, a_t, and o_t denote the reasoning, action, and observation at that turn. An

interaction step is written as

$$(r_t, a_t) \sim \pi_\theta(\cdot \mid x, h_t), \qquad o_t = \mathcal{E}_{\text{env}}(a_t), \qquad h_{t+1} = h_t \oplus (r_t, a_t, o_t), \tag{5}$$

where $\mathcal{E}_{\text{env}}$ maps an action to an observation and $\oplus$ appends the current turn to the interaction history. For Solvent, $\mathcal{E}_{\text{env}}$ is the building-block inventory, which returns precursor-availability observations for a submitted route.

Group-relative policy optimization. For each input $x \sim \mathcal{D}$, the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G trajectories $\{\tau_i\}_{i=1}^{G}$. Let L_i be the number of policy-generated tokens in τ_i, $y_{i,t}$ its t-th token, and $H_{i,t}$ the interaction history before that token. Let R_i be the trajectory reward, and let $\bar{R}$ and σ_R be the mean and standard deviation of rewards within the group. GRPO estimates the advantage from these relative rewards and applies importance sampling and clipping at the token level (Shao et al., 2024):

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) &= \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{t=1}^{L_i}\min\Big(w_{i,t}(\theta)\widehat{A}_i, \text{clip}(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\widehat{A}_i\Big)\right], \\ w_{i,t}(\theta) &= \frac{\pi_\theta(y_{i,t} \mid H_{i,t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid H_{i,t})}, \qquad \widehat{A}_i = \frac{R_i - \bar{R}}{\sigma_R + \varepsilon_A}. \end{aligned} \tag{6}$$

Here, ϵ is the token-level clipping bound and $\varepsilon_A = 10^{-6}$ prevents division by zero. Under GRPO, every token in a trajectory shares the same group-relative advantage, while importance ratios and clipping are computed token by token.

Sequence-level policy optimization. GSPO retains group-relative advantage estimation but computes a length-normalized sequence ratio and applies clipping at the sequence level (Zheng et al., 2025). Let ϵ_l and ϵ_h denote the lower and upper sequence-level clipping bounds. Its objective is

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{\tau_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\Big(s_i(\theta)\widehat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon_l, 1+\epsilon_h)\,\widehat{A}_i\Big)\right]. \tag{7}$$

The sequence-level ratio is

$$s_i(\theta) = \exp\left[\frac{1}{L_i}\sum_{t=1}^{L_i}\log\frac{\pi_\theta(y_{i,t} \mid H_{i,t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid H_{i,t})}\right]. \tag{8}$$

C Experimental Settings

C.1 Benchmarks and Evaluation

Benchmark composition. Our evaluation spans patent-derived synthesis routes and structurally complex bioactive targets. USPTO-190 contains 190 targets with reference routes extracted from USPTO-Full (Chen et al., 2020). ChEMBL-1000 comprises 1,000 compounds sampled from the ChEMBL bioactivity database and provides no reference routes (Liu et al., 2023; Gaulton et al., 2012). The two Pistachio benchmarks were constructed from patent routes under constraints on reaction overlap and pairwise target similarity (Yu et al., 2024). Pistachio Reachable contains shorter, comparatively tractable routes, whereas Pistachio Hard emphasizes longer and deeper synthesis plans. Table 4 summarizes the resulting benchmark statistics. Reference-route statistics characterize the datasets only and are not used to score predicted routes.

Table 4: **Summary of the multi-step retrosynthesis benchmarks.** Reaction counts and depths refer to the provided reference routes; these statistics are unavailable for ChEMBL-1000.

Benchmark	Targets	Avg. reactions	Avg. depth
USPTO-190	190	6.7	6.0
ChEMBL-1000	1,000	–	–
Pistachio Reachable	150	5.5	5.4
Pistachio Hard	100	7.5	7.2

Training–evaluation separation. We remove molecular overlaps with the four evaluation benchmarks from every training stage and further assess structural proximity beyond exact identity. For each test target, we compute its maximum Tanimoto similarity to any retained training target using radius-2, 2,048-bit Morgan fingerprints. A target is scaffold-novel when its canonical Bemis–Murcko scaffold is absent from all retained training targets; targets with empty acyclic scaffolds are excluded. Table 5 summarizes the resulting diagnostics.

Table 5: **Structural separation between training and evaluation targets.** Similarities are computed against the union of retained targets from all training stages. Success uses the same fixed single-trajectory evaluation as Table 1.

Diagnostic	Value
Evaluation targets	1,440
Exact target overlap	0
Median maximum Morgan similarity	0.466
Targets with maximum similarity ≥ 0.8	5.28%
Targets with novel Murcko scaffolds	81.49%
Success on novel-scaffold targets	60.26%

The structural separation remains substantial beyond exact deduplication. In particular, only 5.28% of targets have a close training neighbor at similarity ≥ 0.8. Moreover, 81.49% of scaffold-bearing targets have no Murcko-scaffold match in the training data, and Solvent retains a 60.26% benchmark-macro success rate on this novel-scaffold subset. Among the 410 targets below 0.4 maximum similarity, its target-pooled success rate remains 55.37%. These results support Solvent's generalization beyond structurally similar training examples, rather than mere memorization of the training data.

Building-block inventory. Following prior evaluations of search-based retrosynthesis systems (Liu et al., 2025; Song et al., 2026), our building-block inventory comprises commercially available compounds from eMolecules (release dated April 1, 2026). We freeze the inventory before evaluation and apply the same lookup procedure to all methods evaluated in our pipeline. The prior search-based retrosynthesis systems likewise follow the eMolecules-based inventory protocol of the corresponding benchmark. Each entry is reduced to its parent fragment, neutralized where possible, and represented by canonical isomeric SMILES; predicted route leaves undergo identical normalization before lookup. We use the inventory to check precursor availability during evaluation and provide feedback during Solvent's multi-turn interaction.

Route-level evaluation. Let $d_{\mathcal{T}}^{-}(v)$ denote the in-degree of molecule v in the product-to-precursor route $\mathcal{T} = (\mathcal{V}, \mathcal{E})$. We first define structural and stock success as

$$\begin{aligned} S_{\text{route}}(\mathcal{T}; x, \mathcal{B}) = \mathbf{1}[\text{root}(\mathcal{T}) = x,\ |\mathcal{E}| > 0] \\ \times \mathbf{1}[\mathcal{T} \text{ is connected and acyclic}] \\ \times \prod_{v \in \mathcal{V} \setminus \{x\}} \mathbf{1}\big[d_{\mathcal{T}}^{-}(v) = 1\big] \prod_{\ell \in \mathcal{L}(\mathcal{T})} \mathbf{1}[\ell \in \mathcal{B}]\,. \end{aligned} \tag{9}$$

Here, $\mathbf{1}[\cdot]$ denotes the indicator function, equal to one when its condition holds and zero otherwise. This criterion requires a nontrivial decomposition rooted at the target, a unique parent for every non-root molecule, and termination entirely in the fixed inventory.

Solvent constructs complete synthesis plans directly rather than confining its reasoning to reaction candidates enumerated by a single-step model, allowing it to consider a broader range of transformations. Planning the route as a whole allows the model to coordinate global synthesis strategy with individual disconnection decisions, thereby better exploiting its capacity for multi-step reasoning.

Step-level chemical-validity check. For each proposed step $e = (p, \mathcal{R})$, RDKit first parses and canonicalizes the product p and the precursor set $\mathcal{R}$. We assess each step through two complementary checks on reaction patterns and atom sources. The reaction-support term $c_{\text{sup}}(e)$ checks

whether the product and precursors match the corresponding substructures of an ASKCOS reaction template (Tu et al., 2025). This substructure-based check accommodates new substrate combinations. The atom-source term $c_{\mathrm{atom}}(e)$ checks whether the precursors contain enough atoms of key elements to account for the product. Both terms are binary indicators, equal to one when the corresponding check passes and zero otherwise. Together, these checks allow flexibility in substrate choice while screening for unsupported reaction patterns and missing atom sources. We define $c(e) = c_{\mathrm{sup}}(e)c_{\mathrm{atom}}(e)$. The reported success indicator is

$$S(\mathcal{T}; x, \mathcal{B}) = S_{\mathrm{route}}(\mathcal{T}; x, \mathcal{B}) \prod_{e \in \mathcal{E}} c(e). \tag{10}$$

Thus, every step must pass both checks. Together, these criteria provide a practical assessment of route plausibility, although experimental feasibility can ultimately be established only through laboratory validation. Developing reliable computational proxies for experimental outcomes, including yield, selectivity, and competing reactions, remains an open challenge in synthesis planning (Tu et al., 2025; Roh et al., 2026) and an important direction for future work.

C.2 Evaluation Configurations and Search Budgets

Evaluated LLMs. Our LLM comparison includes four proprietary frontier LLMs: Claude Opus 4.8 (Anthropic, 2026), Gemini 3.5 Flash (Google DeepMind, 2026), Grok 4.5 (SpaceXAI, 2026), and GPT-5.5 (OpenAI, 2026a). The open-weight frontier models are GLM-5.2 (Z.ai, 2026), Kimi K2.5 (Kimi Team, 2026a), DeepSeek-V4-Pro (DeepSeek-AI, 2026), and Kimi K3 (Kimi Team, 2026b); the mid-scale general-purpose models are Qwen3-8B (Yang et al., 2025), Qwen3.5-9B (Qwen Team, 2026a), and Qwen3.6-Flash (Qwen Team, 2026b). We additionally include R^3, an 8B LLM specialized for multi-step route generation (Wang et al., 2026). All evaluated LLMs receive the same target-only prompt and inventory interface and are evaluated with one generation per target, up to five reasoning-and-action turns, and a maximum response length of 32,000 tokens. Temperature, top-p, and model-specific reasoning controls follow each provider's recommended defaults. All outputs are scored by the same route parser, inventory, and chemical-validity criteria.

Cost-controlled subset. To ensure fair and representative comparisons under limited evaluation budgets, all entries marked with † in Table 1 are evaluated on the same frozen, model-independent subset. Structure- and complexity-stratified sampling preserves the chemical diversity of each benchmark while holding the evaluated targets fixed across models. The resulting manifest contains 160 targets from ChEMBL-1000, 70 from Pistachio Hard, 80 from Pistachio Reachable, and 90 from USPTO-190.

For each benchmark $D_d = \{x_i\}_{i=1}^{N_d}$, we compute a radius-2, 2048-bit Morgan fingerprint and six molecular descriptors: heavy-atom, ring, heteroatom, rotatable-bond, and stereocenter counts, and topological polar surface area. Let $P_{dj}(x_i) \in [0, 1]$ denote the within-benchmark percentile rank of target x_i under descriptor j. We define its complexity score as

$$\kappa_d(x_i) = \frac{1}{6} \sum_{j=1}^{6} P_{dj}(x_i). \tag{11}$$

Targets are first partitioned by exact Tanimoto k-medoids clustering of their fingerprints, using eight clusters for ChEMBL-1000 and four for each remaining benchmark. Each cluster is then divided at its median κ_d into lower- and higher-complexity strata. Targets containing bridgehead or spiro atoms form dedicated lower- and higher-complexity strata so that these uncommon ring topologies remain represented.

Let D_{dh} be stratum h, with population N_{dh}, and let H_d be the number of nonempty strata. Given dataset quota n_d, we assign one target to every stratum and distribute the remaining quota approximately in proportion to residual stratum capacity. Equivalently, constrained largest-remainder rounding is applied to

$$q_{dh} = 1 + (n_d - H_d)\frac{N_{dh} - 1}{N_d - H_d}, \qquad \sum_{h=1}^{H_d} n_{dh} = n_d, \quad 1 \le n_{dh} \le N_{dh}. \tag{12}$$

We sample n_{dh} targets uniformly without replacement within each stratum using deterministic seeds derived from the global seed 20260713, dataset, and stratum. The final 400-target manifest is fixed across all subset-evaluated models. For model m, the reported rate on dataset d is the empirical mean on this shared subset,

$$\widehat{p}_{m,d} = \frac{1}{n_d} \sum_{x_i \in \mathcal{S}_d} S(\widehat{\mathcal{T}}_{m,i}; x_i, \mathcal{B}), \qquad \widehat{p}_m = \frac{1}{4} \sum_{d=1}^{4} \widehat{p}_{m,d}, \tag{13}$$

where $\mathcal{S}_d = \bigcup_h \mathcal{S}_{dh}$ and S is defined in Eq. 10. This design retains broad structural and complexity coverage within each benchmark while enabling the direct, target-matched comparison reported in Table 6.

On exactly the same 400 targets, Solvent reaches a 63.26% macro average, within 0.31 percentage points of GPT-5.5 and 0.23 points above Kimi K3, while exceeding the same-sized R^3 by 43.08 points. The target-matched comparison therefore preserves the conclusion of Table 1: Solvent is competitive with frontier LLMs and substantially stronger than mid-scale alternatives. Its relative standing is not attributable to differences in the evaluated target sets.

Table 6: **LLM performance on the shared 400-target subset.** Every model is evaluated on the same frozen targets using the success criterion in Eq. 10. We report benchmark-specific success rates and their unweighted macro average. Em dashes denote undisclosed parameter counts. Gray shading marks the best result among frontier LLMs, and boldface marks the best result among mid-scale LLMs.

Method	Params.	Pistachio Hard	Pistachio Reachable	USPTO-190	ChEMBL-1000	Average (%)
Frontier LLMs (proprietary)						
Claude Opus 4.8	—	38.57	77.50	54.44	55.63	56.54
Gemini 3.5 Flash	—	42.86	72.50	52.22	65.63	58.30
Grok 4.5	—	52.86	71.25	50.00	62.50	59.15
GPT-5.5	—	58.57	82.50	54.44	58.75	63.57
Frontier LLMs (open-weight)						
GLM-5.2	753 B	1.43	6.25	2.22	5.00	3.73
Kimi K2.5	1 T	27.14	36.25	25.56	19.38	27.08
DeepSeek-V4-Pro	1.6 T	42.86	58.75	41.11	46.88	47.40
Kimi K3	2.8 T	54.29	80.00	52.22	65.63	63.03
Mid-scale LLMs						
Qwen3-8B	8 B	0.00	0.00	1.11	0.00	0.28
Qwen3.5-9B	9 B	0.00	1.25	0.00	0.00	0.31
Qwen3.6-Flash	35 B	10.00	22.50	8.89	3.13	11.13
R^3	8 B	18.57	28.75	17.78	15.63	20.18
Solvent	8 B	**64.29**	**81.25**	**50.00**	**57.50**	**63.26**

Prior search-based retrosynthesis systems. Greedy DFS recursively expands the highest-probability reaction returned by a single-step model (Chen et al., 2020). Retro* performs neural-guided A*-like search over an AND–OR tree, combining accumulated reaction cost with a learned estimate of the remaining synthesis cost; Retro*-0 removes the learned future-cost estimate (Chen et al., 2020). Retro*+ and Retro*+-0 apply the self-improvement procedure of Kim et al. (2021) to the single-step models used by Retro* and Retro*-0, respectively, using successful routes discovered by the planner to improve subsequent reaction proposals. EG-MCTS learns an experience-guidance network from successful and failed search trajectories and uses it to guide Monte Carlo tree search (Hong et al., 2023). PDVN alternates planning and policy updates while learning separate value functions for molecular synthesizability and route cost (Liu et al., 2023). Retro-R1 instead trains an LLM controller to call a plug-in single-step model, select among its candidate reactions, and accumulate the resulting decisions in a synthesis graph (Liu et al., 2025).

For the Pistachio comparisons, RootAligned is a single-step sequence model based on root-aligned SMILES and is paired separately with MCTS and Retro* search (Zhong et al., 2022; Segler et al., 2018; Chen et al., 2020). The LLM (MCTS) and LLM (Retro*) baselines likewise use an LLM for single-step reaction prediction within the corresponding search algorithm (Wang et al., 2025). LLM-Syn-Planner generates complete routes and improves a population of candidates through evaluation, selection, and LLM-based mutation (Wang et al., 2025); AOT* generates complete downstream routes for selected molecules, maps valid proposals into an AND–OR tree, and reuses intermediate structures through systematic tree search; we report its DeepSeek-V3 configuration (Song et al., 2026).

Search budget N. Search methods that use single-step retrosynthesis models, including Retro*, Retro*+, EG-MCTS, PDVN, and RETRO-R1, define N as the limit on calls to a single-step retrosynthesis model (Chen et al., 2020; Kim et al., 2021; Hong et al., 2023; Liu et al., 2023; 2025). Route-level LLM planners, such as LLM-Syn-Planner and AOT*, define N by the number of route-generation or tree-expansion operations in their separate search loops (Wang et al., 2025; Song et al., 2026). These two definitions are not interchangeable: the former counts one-step molecule expansions, whereas the latter counts operations that generate or update route-level search states. Solvent uses a different workflow: it directly generates and revises complete routes over multiple turns, without calling a single-step model or using a separate route-search loop.

To compare Solvent with these methods, we need a common concept that covers the different counting schemes described above. We adopt the search budget N for this purpose. Both types of counting fit under this concept because each limits the amount of route-search work that can be performed before a successful route is found. Search methods that use single-step retrosynthesis models count calls to a single-step model, each of which expands one selected molecule and returns candidate reactions. Route-level planners count route-level operations, each of which generates or revises a complete route or expands an existing route tree.

Following the same principle, we define Solvent's search budget strictly and fairly. Solvent's budget is the total number of reaction steps across all submitted routes. Counting each submitted reaction step gives a strict account of route-construction work and prevents complete-route generation from receiving an artificial budget advantage. For a target x, let $\{\widehat{\mathcal{T}}_q\}_{q=1}^{Q}$ denote the sequence of Q route submissions across inventory queries and parallel trajectories, with q indexing submissions. Let $\mathcal{E}(\widehat{\mathcal{T}}_q)$ be the set of reaction steps in submission q, and let C_k be the cumulative number of reaction steps through submission k. The success indicator $S(\mathcal{T}; x, \mathcal{B})$ equals one if the route is structurally valid, rooted at x, terminates entirely in inventory $\mathcal{B}$, and passes both chemical checks at every step; otherwise, it equals zero (Eq. 10). We define

$$C_k = \sum_{q=1}^{k} \left| \mathcal{E}(\widehat{\mathcal{T}}_q) \right|, \qquad \text{Solved}\,@N = \mathbf{1}\Big[\exists k \in \{1, \ldots, Q\} :\ C_k \le N \ \wedge\ S(\widehat{\mathcal{T}}_k; x, \mathcal{B}) = 1 \Big]. \tag{14}$$

A target is counted as solved at budget N only if a successful route is found without exceeding the cumulative reaction-step cost N. Other measures of inference cost cannot provide a common budget for all compared methods. Token counts are not defined for methods without an LLM, and even when available they cover only language-model output, excluding single-step neural predictions and tree-search operations. FLOPs are unavailable for proprietary APIs and, when available, measure model computation rather than route search. Wall-clock time varies with hardware, batching, serving infrastructure, and implementation. The reaction-step count therefore offers a common measure of search effort: each counted step represents a chemically interpretable synthesis hypothesis, and the cumulative count tracks how much reaction space is explored before a viable route is found. The search budget also has a clear chemical meaning. Before a proposed route can be tested in the laboratory, each reaction step needs chemical assessment and concrete reaction conditions (Johnson, 2023; Zhang et al., 2026). The budget therefore approximates the amount of reaction-level work involved in screening candidate routes, which is relevant to practical synthesis planning. Appendix F.2.3 separately reports the available parameter counts, capped output-token counts, and inventory-query counts for the evaluated LLMs.

Solvent inference configuration. For the final benchmark evaluations, we evaluate Solvent at temperature 0.8 and top-p 1.0, with up to five interaction turns and a cumulative output limit of 32k

tokens. For Table 1, we sample one trajectory per target, as for the other LLMs. For Table 2, we use parallel sampling to generate a fixed pool of 100 trajectories per target and accumulate their reaction steps, counting a target as solved if a successful route is found within the search budget N.

D Post-Training Data Construction

D.1 Building Retrosynthetic Intuition

This warm-up stage equips the backbone with broad, route-grounded retrosynthetic knowledge before it is trained to interact with the inventory. We first strictly deduplicate the route corpus used to train R^3 (Wang et al., 2026) against all evaluation benchmarks, yielding 104,867 target–route pairs. Each pair is passed to Intern-S1 (Bai et al., 2025) through an annotation-generation prompt that includes both the target and its route. Intern-S1 then produces a target-level structural analysis and a stepwise account of every product-to-precursor transformation. The resulting annotation is paired with a separate, target-only task prompt to form the warm-up fine-tuning example. Thus, the route guides annotation generation but is not included in the input presented to the model being fine-tuned. The complete prompts are provided below; their placeholders are populated at runtime with the corresponding target and route.

Annotation-generation prompt

```
You are an expert organic retrosynthesis instructor writing knowledge-distillation supervision
    .

This task is not to simulate a model's private chain-of-thought and not to write an informal
    problem-solving monologue.
Write a structured, polished teaching explanation that injects retrosynthesis knowledge into a
     training example.

You are given:
- a target molecule as a SMILES string
- a route that should be used to construct the supervised explanation

### Input
Target Molecule: {TARGET_MOLECULE}
Route: {REFERENCE_ROUTE}

### Output Protocol
Do not output <think>, </think>, <reasoning>, or </reasoning>.
Those tags are reserved for downstream conversion.
The first visible token of your response must be exactly:
Target analysis:

Do not prepend any sentence before Target analysis.
Do not write an introduction, plan, acknowledgment, or meta explanation.

Return only the following structured text:

Target analysis: <complete target SMILES>
<target-only structural analysis>

Step 1: PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES
Explanation:
<knowledge explanation for Step 1>

Step 2: PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES
Explanation:
<knowledge explanation for Step 2>

...

<answer>{REFERENCE_ROUTE}</answer>

The <answer> block must be valid JSON and must exactly equal:
{REFERENCE_ROUTE}

### Style Requirements
Use formal scientific English.
Do not write in first person.
Do not write conversational openings such as "Okay", "Let me", "I need to", "I will", "Hmm", "
    Starting with", "First, I need", or similar thinking-aloud phrasing.
Do not mention the user, prompt, task setup, supplied route, internal route, given route,
    reference route, answer key, gold answer, or any external supervision.
```

```
Do not say that a route was provided or that the explanation is being generated from a route.

The output should read like a concise expert annotation for a retrosynthesis dataset:
- structured
- target-driven
- chemically specific
- explanatory rather than speculative
- suitable as supervised training text

### Target Analysis Rules
The Target analysis section should analyze only the target molecule itself.
It must not describe any specific retrosynthetic step, precursor, route, final answer, or
    planned disconnection.
Do not say which bond will be disconnected first.
Do not mention precursors, route steps, or the final route in Target analysis.

Target analysis should cover:
- the complete target SMILES exactly as written
- visible functional groups and heteroatom environments
- ring systems, aromatic systems, charges, and encoded stereochemical markers if present
- strategically relevant bonds or motifs visible in the target
- structural features that make the target synthetically nontrivial

Keep target analysis detailed enough to orient the molecule, but keep all step-specific bond-
    breaking logic for the Step explanations.

### Step Explanation Rules
Each Step N line must state the exact transformation first:
Step N: PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES

Then write an Explanation section for that step.
The Explanation should teach why this exact route step is a reasonable retrosynthetic
    relationship.
It should be detailed and chemically grounded, but it should not imitate a live thinking
    process.

For each step, explain these aspects in an elaborated, insightful and professional way:
- which current molecule is being simplified
- which local bond, functional relationship, ring closure, substituent relationship, or
    activation relationship is represented by the transformation
- why the listed precursor set is a plausible retrosynthetic simplification of that current
    molecule
- what atoms, substituents, functional groups, or redox/activation states are retained
    unchanged across the transformation
- what exact atoms, bonds, leaving handles, or functional groups differ between the product
    side and precursor side
- the role of every precursor written after ">>", including small or reagent-like molecules
- which precursor continues to a later step, if it appears as the product side of a later
    transformation
- which precursors terminate as leaf materials or reagent-like leaf components
- how any activated handle, halogenated precursor, acyl-transfer component, carbonate, small
    molecule, or reagent-like precursor participates at the route-planning level

For functional-group interconversions, redox steps, and formal precursor relationships,
    explain the chemical knowledge behind the transformation rather than only naming it.
    Clarify why the precursor form is a useful synthetic equivalent of the product form,
    which part of the scaffold is preserved, and what forward operation would connect the
    precursor to the product at the route-planning level.

For bond-forming or bond-breaking steps, explain why that local connection is strategically
    meaningful, how each precursor maps onto a visible part of the current molecule, and why
    no listed small molecule should be omitted as a mere condition.

Do not collapse the Explanation into a reaction-name label.
Use concrete SMILES-grounded details.
Do not introduce hidden reagents, implicit byproducts, catalysts, solvents, workup components,
     or alternative partners that are not present in the route.

### SMILES Fidelity
Exact SMILES-string fidelity is the highest priority.
Use only the SMILES strings provided in the target and route.
Infer atom identity, bond order, connectivity, charge, stereochemistry, and functional groups
    from the current SMILES itself.
Do not imply access to a drawing, molecular graph, database annotation, external route
    metadata, or unlisted reaction conditions.

Before explaining a step, compare the molecule before ">>" with every listed precursor after
    ">>".
Discuss only the structural relationship represented by that exact transformation.
```

```
If a named functional group, reaction class, heterocycle class, or oxidation-state description
     is uncertain, prefer direct atom-and-bond descriptions.
Do not say that the target, precursor, or final transformation is a typo, artifact, impossible
    , invalid, inconsistent, mismatched, or chemically wrong.
If a transformation is formal or non-obvious, describe it as a route-level precursor
    relationship and explain the visible structural correspondence.

### Route Discipline
The first step must start from the target molecule.
A precursor continues only if it appears as the product of a later transformation.
Leaf precursors are explicitly listed precursor molecules that are not further disconnected.

Every molecule explicitly listed after ">>" is part of the precursor set.
Do not drop small molecules or reagent-like components because they look like conditions.
Do not call a terminal precursor an intermediate, product, final product, or target product.
Do not add transformations outside the route JSON.
Do not create self-loops, repeated transformations, extra roots, side branches, or alternative
     routes.

### Final Requirements
The explanation before <answer> must contain:
- exactly one Target analysis section
- one Step N line for every transformation in the route, in order
- one Explanation: label after every Step N line
- no <think> tags
- no <reasoning> tags
- no first-person or conversational thinking-aloud language
- no statement that a route, answer, prompt, user, or instruction was supplied

Now generate the structured knowledge-distillation explanation.
```

Warm-up training prompt

```
You are an expert in retrosynthesis planning. Your task is to propose a retrosynthetic route
    for a target molecule from plausible starting materials.

Given a target molecule in SMILES format, work backwards and produce a route in the following
    format:

<think>
Target analysis: <complete target SMILES and substantive structural analysis>

Step 1: PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES
Explanation:
Explain in detail why this exact disconnection is reasonable, what local substructure or bond
    relationship is being simplified, what structural change connects the product to the
    listed precursors, the role of every listed precursor, and which precursor continues to
    later steps if any.

Step 2: PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES
Explanation:
Explain the next disconnection in the same target-driven, chemically specific style.

...
</think><answer>["PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES", "..."]</answer>

Strategy and chemistry requirements:
- Analyze the target molecule first in enough detail to orient the route. Identify visible
    functional groups, heteroatoms, charges, ring systems, and strategically important bonds
    from the SMILES string.
- Work backwards from the target and maintain a coherent retrosynthesis tree. The root is the
    target molecule; later steps should further decompose intermediates generated by earlier
    steps.
- Each route step should be chemically reasonable and follow plausible retrosynthetic logic,
    such as amide or urea formation, acylation, substitution, SNAr, esterification, activated
     electrophile use, halogenation or functional-group interconversion when appropriate.
- Prefer disconnections that simplify the molecule while preserving chemically meaningful
    fragments. Between possible disconnections, favor the one that leaves complex fragments
    for further decomposition and simple fragments as leaves.
- Continue decomposing any intermediate that appears too complex to be a realistic starting
    material.
- Stop when all leaf molecules are reasonably plausible starting materials or reagent-like
    components.
- Prefer leaf molecules that are small and simple: low molecular complexity, few rings,
    limited functional groups, simple aliphatic amines, alcohols, carboxylic acids, acyl
    chlorides, sulfonyl chlorides, isocyanates, nitriles, simple heterocycles, common
    reagents, or activated electrophiles.
```

```
- If a molecule contains multiple fused rings, multiple heteroatoms, or several different
    functional groups, it is more likely to be an intermediate and should usually be
    decomposed further.
- Treat every precursor written after ">>" as part of the route, including small molecules,
    activated electrophiles, halogenating or acyl-transfer reagents, and other reagent-like
    components.
- Do not omit a precursor because it looks like a reagent, condition, additive, or solvent if
    it is explicitly written in the route step.

Output requirements:
- The <answer> block must be valid JSON.
- Each route step must use the exact "PRODUCT>>PRECURSORS" format.
- Multiple precursors must be separated by dots.
- The list in <answer> must contain all route steps needed to synthesize the target from the
    final leaf materials.
- The Step N line inside <think> must state the proposed step first, followed by an
    Explanation: section.
- Keep each Explanation section target-driven, chemically specific, and focused on the current
     step.
- In each Explanation section, explain why the disconnection is plausible, what structural
    change it represents, the role of each precursor, which precursor continues to later
    steps if any, and which precursors terminate as leaf materials or reagent-like components
    .
- Do not reduce the reasoning to short labels or generic reaction names. Use concrete SMILES-
    grounded details, but do not introduce hidden reagents, unlisted conditions, alternative
    routes, or molecules not present in the route.

Target molecule: {TARGET_MOLECULE}
```

We retain an annotation only when it contains exactly one valid JSON answer block matching the source route, one ordered step and explanation for every transformation, and a target analysis free of precursor or disconnection information. We further reject incomplete outputs, conversational or meta-level language, disclosures of the conditioning route, and any deviations from the source transformations. Intern-S1's hidden reasoning is removed before dataset assembly; only the visible, structured annotation is retained. This validation accepts 102,212 annotations for warm-up fine-tuning.

Table 7: **Configuration used to generate retrosynthetic warm-up annotations with Intern-S1.**

Setting	Value
Annotation model	Intern-S1
Source target–route pairs	104,867
Generations per route	1
Temperature	0.7
Top-p / top-k	1.0 / 50
Reasoning mode	Enabled
Valid annotations	102,212

D.2 Teacher-Guided Cold Start for Long-Horizon Reasoning

This stage establishes the model's ability to perform long-horizon reasoning. We assemble 2,200 structurally challenging targets from USPTO-full (Dai et al., 2019) after strictly removing all overlaps with the evaluation benchmarks. Selection uses only molecular structure, balancing molecular size, ring topology, heteroaromaticity, stereochemistry, flexibility, and functional-group composition while limiting repeated Murcko scaffolds. DeepSeek-V4-Pro (DeepSeek-AI, 2026) then generates eight independent tool-interaction trajectories per target. Because the model typically produces long reasoning traces, each trajectory allows at most three inventory-query turns followed by one finalization turn. This retains multiple opportunities for route revision while keeping the complete trajectory within the 32k-token generation budget.

The target-only task prompt is identical during rollout generation and fine-tuning. DeepSeek-V4-Pro generates the reasoning, inventory queries, and final answer, interleaved with observations returned by the tool; this complete trace becomes the autoregressive target paired with the same user prompt. The target placeholder in the shared prompt below is populated at runtime, while the inventory tool schema is supplied through the model's function-calling interface.

Shared rollout and fine-tuning prompt

```
You are an expert organic chemist solving a multi-step retrosynthesis problem with one tool.

Language rule:
- Write all reasoning, tool-call arguments, and final answer in English only.

Available tool:
- check_emolecules_stock: checks whether every final leaf material in a proposed
    retrosynthetic route is available in the eMolecules stock database.

Core requirements:
- Every retrosynthetic disconnection must be chemically reasonable and must correspond to a
    plausible forward reaction.
- Do not invent arbitrary stocked molecules as leaves. Stock availability is only an
    availability check, not a correctness check.
- Use your chemistry knowledge to analyze each disconnection before checking stock
    availability.
- The stock check should be applied to the complete proposed route after the route has been
    chemically reasoned through.
- You may only finalize after check_emolecules_stock returns stock_ok=true for the complete
    route.
- If stock_ok=false, revise the route and check the complete revised route again.
- Treat every precursor written after ">>" as part of the route, including small or reagent-
    like molecules.

Task:
Given only the target SMILES below, propose a chemically plausible multi-step retrosynthetic
    route.

Target molecule:
{target}

Required workflow:
1. Analyze the target molecule as a whole. Identify important rings, heteroatoms, functional
    groups, charges, stereochemical markers, strategic bonds, and plausible first
    disconnections.
2. Decompose the molecule step by step. For each proposed single-step disconnection, reason
    from atoms, bonds, local connectivity, functional groups, retained fragments, and
    precursor roles, and check whether the corresponding forward reaction is chemically
    plausible.
3. Continue until the route ends in plausible leaf materials or reagent-like components.
4. Call check_emolecules_stock with the complete proposed route.
5. After the stock check, continue from the tool result. If stock_ok=true and the route is
    chemically plausible, provide the final route. If stock_ok=false, revise the route and
    check again before finalizing.

Final answer format:
<answer>["PRODUCT_SMILES>>PRECURSOR1_SMILES.PRECURSOR2_SMILES", "PRECURSOR1_SMILES>>
    SIMPLER_PRECURSOR.SMALL_REAGENT", "..."]</answer>

Formatting requirements:
- The <answer> block must contain valid JSON.
- Each route step must be a string in exact PRODUCT_SMILES>>PRECURSOR_SMILES format.
- Multiple precursors must be separated by dots.
- The first PRODUCT_SMILES must exactly match the target molecule.
- The route must form one connected retrosynthetic tree rooted at the target.
- Do not include anything after the final </answer> tag.
```

Candidate trajectories undergo three stages of filtering. Route-level checks require a parseable, nontrivial tree rooted at the target and terminating entirely in available building blocks. Interaction-level checks require the final route to match a preceding query that received a successful inventory observation. Finally, each reaction must pass the reaction-support and atom-source checks defined in Appendix C.1. We additionally remove malformed, truncated, excessively long, or repetitive traces.

We rank the surviving candidates using

$$Q(\tau) = 0.22C(\tau) + 0.18L(\tau) + 0.16D(\tau) + 0.18U(\tau) + 0.18J(\tau) + B_{\text{chem}}(\tau) - 0.06R(\tau), \tag{15}$$

where C, L, and D measure target complexity, normalized trace length, and route depth; U rewards underrepresented structural attributes; J is the plausibility assessment produced by DeepSeek-V4-Pro; R is the repeated-n-gram ratio; and B_{chem} is 0.09 for the strict chemical tier, 0.02 for the relaxed tier, and 0 otherwise. The strict tier requires every reaction to receive a high-confidence feasible judgment, whereas the relaxed tier permits a limited fraction of medium-confidence judgments but no infeasible or low-confidence reactions. For each canonical target, we first prioritize

candidates that pass the LLM-based assessment and then maximize Q. The final set is selected greedily by combining Q with bonuses for underrepresented descriptor groups and penalties for redundant groups, subject to at most six trajectories per Murcko scaffold. This procedure yields 700 teacher-generated trajectories for fine-tuning π_{warmed}.

Table 8: **Configuration used to generate and curate the teacher-guided cold-start trajectories.**

Setting	Value
Teacher model	DeepSeek-V4-Pro
Target source	USPTO-full
Rollout targets	2,200
Rollouts per target	8
Maximum inventory-query turns	3
Cumulative generation limit	32k tokens
Temperature / top-p	1.0 / 1.0
Reasoning effort	High
Maximum samples per Murcko scaffold	6
Selected trajectories	700

D.3 Target Selection for First-Round RL

The first-round RL curriculum comprises 10,000 targets drawn from the USPTO-full product pool (Dai et al., 2019) and strictly deduplicated against all evaluation benchmarks. We retain canonical, single-component products that are absent from the building-block inventory and exclude structurally trivial molecules. Rather than sampling uniformly, we stratify targets by structural complexity to span progressively more demanding route-construction problems. The curriculum increases the representation of polycyclic and fused-ring systems, bridged and spiro scaffolds, heteroaromatics, stereocenters, flexible linkers, and densely functionalized molecules, while controlling the concentration of each Murcko scaffold family. This construction preserves broad scaffold coverage as route complexity increases.

D.4 Curating Trajectories for Reasoning Bootstrapping

Sampling candidate trajectories. Reasoning bootstrapping begins with trajectories sampled from π_{seed}. We construct target pools from structurally challenging USPTO-full products (Dai et al., 2019) to cover variation in molecular size, ring topology, stereochemistry, flexibility, and functional-group composition. Before rollout, we canonicalize and deduplicate the targets and remove all exact overlaps with the evaluation benchmarks. Best-of-N sampling under the configuration in Table 9 yields 400,000 candidate trajectories. Rollout generation and SFT use the same target-only task prompt and inventory schema reported in Appendix D.2.

Strict trajectory filtering. Candidates must satisfy route-level, interaction-level, and reaction-level criteria. The final answer must define a valid, nontrivial route tree rooted at the target and terminating entirely in available building blocks. Its interaction trace must be well-formed, agree with the final answer, and include a successful inventory check of the submitted route. Each reaction must also pass the chemical-validity assessment in Appendix C.1. We discard malformed or truncated outputs, answers emitted before the corresponding inventory observation, and training sequences that exceed 32k tokens after formatting.

Reasoning-quality selection. Following LIMO (Ye et al., 2025), the reasoning score q_{reason} combines elaboration, self-verification, exploratory revision, and adaptive granularity with weights of 0.30, 0.20, 0.25, and 0.25, respectively. The style score q_{style} combines balanced coverage of these dimensions, moderate generation uncertainty, tool-use patterns, and successful route revision after the inventory identifies unavailable precursors, with weights of 0.35, 0.25, 0.20, and 0.20. We rank the surviving trajectories by

$$Q(\tau) = 0.85 q_{\mathrm{reason}}(\tau) + 0.15 q_{\mathrm{style}}(\tau) - 0.30 r_{\mathrm{rep}}(\tau), \tag{16}$$

where r_{rep} is the repeated 5-gram ratio. For each canonical target, we retain only the highest-scoring trajectory. A final decontamination pass against the evaluation benchmarks removes two additional targets, leaving 9,550 trajectories from 9,550 unique targets.

Table 9: **Generation and curation configuration for reasoning bootstrapping.**

Setting	Value
Generating policy	π_{seed}
Target source	USPTO-full
Trajectory sampling	Best-of-N
Temperature / top-p	1.0 / 1.0
Maximum reasoning-and-action turns	5
Cumulative generation limit	32k tokens
Candidate trajectories	400,000
Selected trajectories / unique targets	9,550 / 9,550

D.5 Target Curriculum for Final-Stage RL

The final-stage RL curriculum concentrates learning on unresolved yet structurally diverse route-construction problems. We first build a base pool from canonical, single-component USPTO-full products (Dai et al., 2019) that are absent from the building-block inventory. Candidate priority reflects the preceding policy's failure profile, molecular complexity, and structural attributes, while diversity-aware selection limits repeated Murcko scaffolds. During base-pool construction, we exclude evaluation targets and targets already selected for SFT.

We then identify 2,909 persistent hard cases whose sampled trajectories during reasoning bootstrapping yield a 0% success rate. These targets replace an equal number of lower-priority examples in the base pool, preserving its overall size and broad scaffold coverage while shifting the curriculum toward unresolved problems. After final decontamination against the four evaluation benchmarks, the resulting curriculum contains 10,000 unique targets.

E Training Recipe

We initialize Solvent from the retrosynthesis-pretrained Qwen3-8B checkpoint developed for R^3 (Wang et al., 2026) and use it as the backbone for our post-training pipeline.

E.1 Warm-Up for Retrosynthetic Intuition

We begin post-training with full-parameter SFT of the retrosynthesis-pretrained 8B backbone on the curated Intern-S1 annotations described in Appendix D.1. Of the 102,212 accepted examples, 1,000 are reserved for validation. Table 10 summarizes the training configuration. The resulting policy, π_{warmed}, initializes the subsequent teacher-guided cold start.

Table 10: **Training configuration for the retrosynthetic-intuition warm-up.**

Setting	Value
Initialization	R^3 retrosynthesis-pretrained Qwen3-8B checkpoint
Fine-tuning method	Full-parameter SFT
Examples / validation split	102,212 / 1,000
Training epochs	2
Maximum sequence length	4,096 tokens
Global batch size	64
Peak learning rate	3×10^{-5}
Learning-rate schedule	Cosine
Warm-up ratio	0.05
Precision	BF16

E.2 TEACHER-GUIDED COLD START

We next fine-tune π_{warmed} on the 700 DeepSeek-V4-Pro trajectories curated as described in Appendix D.2. Teacher-guided SFT cold-starts the model for long-horizon reasoning. We select the checkpoint with the lowest validation loss and use it to initialize first-round RL. Table 11 reports the training configuration.

Table 11: **Training configuration for the teacher-guided cold start.**

Setting	Value
Initialization	π_{warmed}
Fine-tuning method	Full-parameter SFT
Examples / validation split	700 / 50
Training epochs	2
Maximum sequence length	32,768 tokens
Global batch size	8
Peak learning rate	2×10^{-5}
Learning-rate schedule	Cosine
Warm-up ratio	0.05
Precision	BF16

E.3 FIRST-ROUND RL FOR STABLE LONG-HORIZON INTERACTION

Starting from the teacher-guided cold-start policy, we apply GSPO (Zheng et al., 2025) to the 10,000-target curriculum described in Appendix D.3. The reward follows Equation 3, assigning weights of 0.6, 0.2, and 0.2 to route success, inventory verification, and interaction validity, respectively. Truncated trajectories are excluded from policy updates, while dynamic group filtering and overlong-response penalties improve training stability. The resulting policy is denoted by π_{seed}.

Table 12: **Training configuration for first-round RL.**

Setting	Value
Initialization	Teacher-guided cold-start policy
Optimization algorithm	GSPO
Training targets	10,000
Optimization updates	50
Rollout targets per batch	128
Rollouts per target	8
Global batch size	512
Maximum interaction turns	5
Context / response limit	32,768 / 30,000 tokens
Rollout temperature / top-p	0.9 / 0.9
Learning rate / schedule	1×10^{-6} / constant
Warm-up updates	20
Precision	BF16

E.4 REASONING BOOTSTRAPPING

To construct π_{boot}, we return to π_{warmed} and perform full-parameter SFT on the 9,550 trajectories curated in Appendix D.4. We monitor validation performance and held-out entropy during training, selecting the first-epoch policy to improve retrosynthesis performance while retaining exploratory capacity. The configuration is reported in Table 13.

Table 13: **Training configuration for reasoning bootstrapping.**

Setting	Value
Initialization	π_{warmed}
Fine-tuning method	Full-parameter SFT
Examples / validation split	9,550 / 100
Training epochs	1
Maximum sequence length	32,768 tokens
Global batch size	8
Peak learning rate	2×10^{-5}
Learning-rate schedule	Cosine, minimum 1×10^{-6}
Warm-up ratio	0.05
Precision	BF16

E.5 Final-Stage RL

We optimize π_{boot} with GSPO on the 10,000-target curriculum described in Appendix D.5, using the gated reward in Equation 4. Dynamic sampling concentrates updates on informative rollout groups, truncated trajectories are excluded from the policy loss, and overlong responses receive a length penalty. A KL penalty with coefficient 0.01 further limits policy drift. Table 14 summarizes the core training configuration.

Table 14: **Training configuration for final-stage RL.**

Setting	Value
Initialization	π_{boot}
Optimization algorithm	GSPO
Training targets	10,000
Optimization updates	780
Rollout targets per batch	64
Rollouts per target	8
Global batch size	512
Maximum interaction turns	5
Context / response limit	32,768 / 30,000 tokens
Rollout temperature / top-p	1.0 / 1.0
Learning rate / schedule	1×10^{-6} / constant
Warm-up updates	20
Precision	BF16

F Additional Experiments

F.1 Additional Ablation Studies

F.1.1 Retrosynthetic-Intuition Warm-Up

Retrosynthetic-intuition warm-up provides a stronger initialization for long-horizon reasoning. We isolate its contribution by fine-tuning either the retrosynthesis-pretrained backbone or the warmed-up policy π_{warmed} on the same reasoning-bootstrapping corpus. Both models use the same training data and optimization settings and are evaluated at the same training steps. We measure success rate on the four benchmarks, comprising 1,440 targets, with eight trajectories per target, temperature 1.0, top-p 1.0, and at most five reasoning-and-action turns.

Table 15: **Retrosynthetic-intuition warm-up consistently improves success rates under matched reasoning-bootstrapping SFT.** Results are macro averages across the four benchmarks.

	Pass@1 (%)			Pass@8 (%)		
SFT epochs	Without warm-up	With warm-up	Δ	Without warm-up	With warm-up	Δ
1	24.99	**26.42**	+1.43	65.24	**69.62**	+4.38
2	30.21	**32.61**	+2.40	70.92	**73.10**	+2.18
3	28.50	**32.94**	+4.44	74.70	**75.26**	+0.56

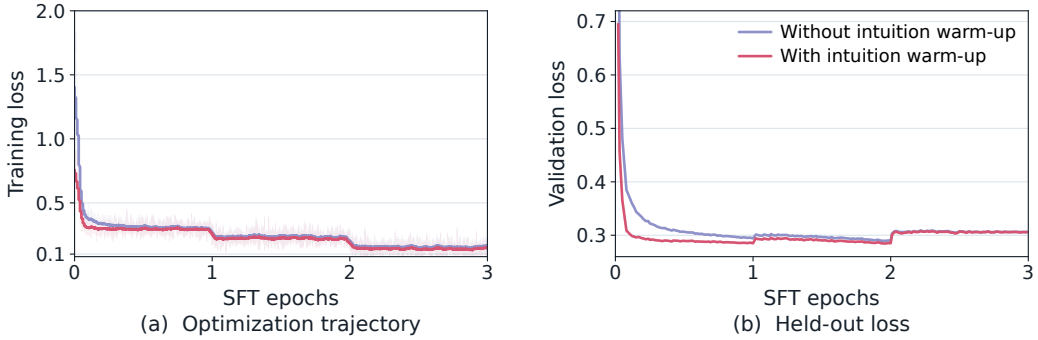


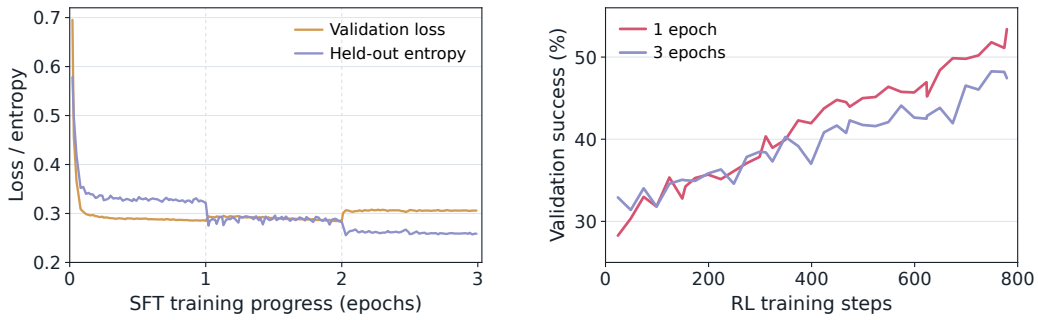
Figure 5: **Retrosynthetic-intuition warm-up improves the optimization of subsequent reasoning-bootstrapping SFT.** Without and with intuition warm-up, training starts from the retrosynthesis-pretrained backbone and π_{warmed}, respectively. (a) Per-step training loss under the two matched initializations; pale traces show individual updates and solid curves show centered 61-update averages. (b) Validation loss throughout training.

Figure 5 shows that warm-up substantially lowers the initial training and validation losses and retains a validation advantage through the first two epochs before the two runs converge. More importantly, this optimization advantage translates into stronger route-solving performance across training depths (Table 15).

Warm-up improves both macro Pass@1 and Pass@8 at all three training depths. Together with the lower early training and validation losses, these consistent gains demonstrate that retrosynthetic-intuition warm-up provides a more effective initialization for subsequent reasoning-bootstrapping SFT.

F.1.2 Training Depth for Reasoning Bootstrapping

Limiting reasoning bootstrapping to one epoch preserves exploration capacity for stronger downstream RL. We isolate the effect of training depth by fine-tuning π_{warmed} on the same curated trajectory corpus for one, two, or three epochs under an otherwise identical configuration. Table 16 reports validation loss, entropy, and task performance after each epoch. We then train the one- and three-epoch policies with the same final-stage RL configuration to compare their subsequent performance.



(a) SFT diagnostics (b) Downstream RL performance

Figure 6: **One epoch of reasoning bootstrapping yields stronger downstream RL.** (a) Validation loss and held-out entropy throughout three epochs of SFT. (b) Validation success during matched final-stage RL initialized from the one- and three-epoch policies. RL validation uses the default rollout decoding configuration ($T = 1.0$, top-$p = 1.0$; different from the benchmark decoding configuration).

Table 16: **Validation loss, entropy, and task performance after different numbers of reasoning-bootstrapping epochs.** Macro success is averaged across the four benchmarks. Bold marks the column maximum, except for loss, where lower is better. Final parse denotes the rate of parseable final answers.

Epoch	Eval loss	SFT entropy	Macro P@1 (%)	Macro P@8 (%)	Rollout NLL	Final parse (%)
1	**0.2850**	**0.3217**	26.42	69.62	**0.3151**	71.53
2	0.2851	0.2830	32.61	73.10	0.2488	**81.32**
3	0.3058	0.2586	**32.94**	**75.26**	0.1927	77.01

Performance before RL improves with training depth: macro Pass@1 rises from 26.42% to 32.94%, while Pass@8 increases from 69.62% to 75.26% (Table 16). However, the full SFT trace shows that validation loss has largely converged by the end of the first epoch, whereas held-out entropy continues to decline throughout later training (Figure 6(a)). Held-out entropy measures uncertainty in next-token predictions on fixed validation sequences. Rollout NLL measures the average negative log-probability of tokens sampled during generation. Both decrease with additional SFT, suggesting reduced exploration despite higher success rates before RL.

The RL comparison shows that higher success before RL does not necessarily lead to better performance afterward (Figure 6(b)). Although the three-epoch policy begins RL with higher validation success, its improvement plateaus earlier. The one-epoch policy overtakes it as training proceeds and reaches 53.40%, compared with 47.43% from the three-epoch policy. We therefore select the one-epoch policy for final-stage RL.

F.1.3 The Large Final-Stage RL Gain Depends on Reasoning Bootstrapping

Effective RL benefits from a starting policy that balances task performance with exploratory capacity (Li et al., 2026). We test whether the intermediate training stages establish this starting point by comparing the full path initialized from π_{boot} with direct application of the same GSPO recipe to the teacher-guided Round 1 SFT policy. The latter therefore bypasses both Round 1 RL and reasoning bootstrapping. The target curriculum, reward, optimizer, rollout configuration, and validation protocol remain fixed. Figure 7 compares the two paths through 600 training updates.

The results show that the bootstrapped policy provides a stronger starting point for final-stage RL, reaching 45.69% validation success compared with 9.31% when the same RL procedure starts from the Round 1 SFT policy.

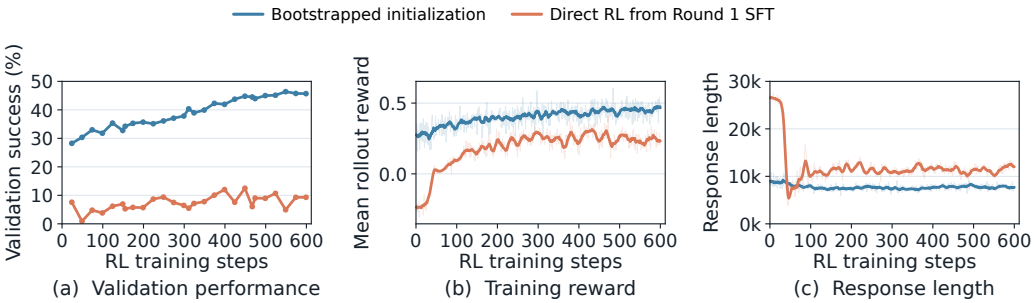


Figure 7: **Final-stage RL yields large capability gains only after reasoning bootstrapping.** (a) Validation success. (b) Mean reward over training rollouts. (c) Mean training-rollout response length. Pale traces in (b) and (c) show individual updates, and solid curves show centered 11-update averages. Both runs use the default rollout decoding configuration ($T = 1.0$, top-$p = 1.0$); curves are shown through update 600.

F.1.4 The Role of Dynamic Sampling in Training Solvent

Dynamic sampling improves RL training. Dynamic sampling in DAPO excludes groups whose sampled outcomes are uniformly correct or incorrect, concentrating policy updates on prompts that retain a useful learning signal (Yu et al., 2025). We adapt this principle to the two route-level outcomes used in our training procedure. For target x_i, let $\mathcal{G}_i = \{\tau_{i,j}\}_{j=1}^{G}$ denote a group of $G = 8$ trajectories. For each trajectory $\tau_{i,j}$, $r_{i,j}^{\text{stock}} = 1$ if its final route is structurally valid and stock-complete, and $r_{i,j}^{\text{chem}} = 1$ if that route is structurally valid and passes the chemical-validity checks; each indicator is zero otherwise. We define

$$\mathbf{z}_{i,j} = \left(r_{i,j}^{\text{stock}}, r_{i,j}^{\text{chem}}\right) \in \{0,1\}^2, \qquad m_i = \mathbb{I}[\exists j : \mathbf{z}_{i,j} \neq (0,0)]\,\mathbb{I}[\exists j : \mathbf{z}_{i,j} \neq (1,1)]\,. \tag{17}$$

A group is retained when $m_i = 1$. This rule removes groups in which every trajectory fails both criteria or every trajectory satisfies both, while preserving partial and mixed outcomes. Sampling continues until each update contains $B = 64$ retained groups.

We evaluate the effect of dynamic sampling during final-stage RL by disabling it while keeping all other training settings fixed. Figure 8 compares their training dynamics.

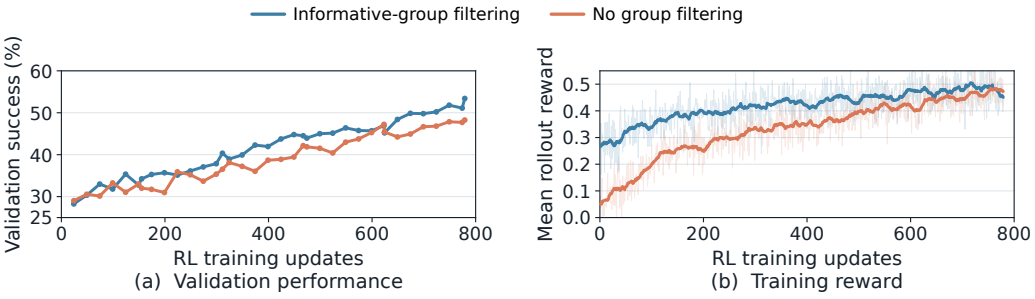


Figure 8: **Effect of dynamic sampling during final-stage RL.** (a) Validation success with and without dynamic sampling. (b) Mean reward over the training rollouts retained for each policy update. Pale traces show individual updates, and solid curves show centered 21-update averages. Validation uses the default rollout decoding configuration ($T = 1.0$, top-$p = 1.0$).

With dynamic sampling, validation success reaches 53.40%, compared with 48.26% without it, a gain of 5.14 percentage points.

F.1.5 Adaptive Revision from Availability Feedback

Availability feedback drives adaptive revision in long-horizon reasoning. To examine the role of availability feedback, we conduct a paired ablation study under three feedback settings. We select a separate set of 500 targets from USPTO-full, stratified by molecular size, ring count, and heteroatom count independently of model outputs. After canonicalization, the set has no exact target overlap with the training data or the four evaluation benchmarks. *True feedback* returns the actual inventory status of every queried precursor. *Masked feedback* preserves the interaction but withholds all availability information. *Shuffled feedback* preserves the number of available and unavailable precursors while permuting their assignments among the queried molecules. The three conditions use the same checkpoint, decoding configuration, and target-specific sampling seed.

Table 17: **Paired intervention on inventory feedback.** Results are computed from one trajectory for each of 500 held-out targets. The success rate requires a structurally valid, stock-complete route that passes the chemical-validity assessment.

Feedback	Success (%)	Stock-complete (%)	Chemical validity (%)	Mean response tokens	Mean inventory queries
True	**68.20**	**80.40**	**73.60**	7,020	2.40
Shuffled	39.00	43.80	46.60	9,158	3.27
Masked	17.60	19.20	47.20	7,891	3.63

As shown in Table 17, true feedback improves success by 29.2 percentage points over shuffled feedback (95% paired bootstrap CI: 24.4–34.2) and by 50.6 points over masked feedback (45.8–55.4). These results show that accurate precursor-availability feedback helps Solvent revise routes successfully.

F.2 Further Analyses

Held-out configuration-selection protocol. We construct a frozen 500-target validation split from USPTO-full for selecting the inference configuration. Selection is independent of model outputs and stratified by molecular size, ring count, and heteroatom count under Murcko-scaffold caps. After canonicalization, we remove targets present in the building-block inventory and enforce zero exact target overlap with all training corpora and the four evaluation benchmarks. This split is used only for configuration selection.

F.2.1 Selecting the Number of Interaction Turns on Held-Out Data

Held-out selection supports the five-turn inference budget. The main evaluation protocol prespecifies a maximum of five interaction turns. We select among turn limits of 2–5 on the held-out split using one trajectory per target, temperature 0.8, and top-p 1.0; six- and seven-turn limits provide sensitivity analyses beyond this protocol. Table 18 reports performance and inference behavior across turn limits.

Table 18: **Held-out performance and inference behavior across turn limits.** Blue shading marks the selected configuration, and bold marks the maximum in each performance column. Tool calls are averaged over all trajectories or successful trajectories only.

	Performance			Inference behavior			
Turn limit	Success rate (%)	Stock-complete routes (%)	Chemically valid routes (%)	Mean tool calls (all)	Mean tool calls (successful)	Mean response tokens	Truncated (%)
2	27.20	28.40	27.80	1.71	1.00	4,524	0.00
3	50.20	56.00	51.60	2.07	1.38	5,743	0.20
4	62.80	70.40	65.40	2.33	1.71	6,563	0.00
5	68.00	77.40	71.80	2.48	1.92	7,063	0.20
6	69.20	**83.60**	74.00	2.49	2.10	7,294	0.20
7	**70.60**	83.40	**76.20**	2.54	2.12	7,459	0.20

Within the predefined selection range, success increases monotonically from 27.20% with two turns to 68.00% with five. Allowing six or seven turns yields smaller gains of 1.2 and 2.6 percentage points, respectively; both paired 95% bootstrap confidence intervals include zero (-3.4–5.6 and -2.0–7.2).

Mean inventory use and response length grow gradually as the turn limit increases, while truncation remains at or below 0.20% throughout. The small behavioral changes beyond five turns mirror the saturation in success. A five-turn limit is therefore selected within the prespecified range and fixed before benchmark evaluation; allowing additional turns yields modest gains.

F.2.2 Selecting the Sampling Temperature on Held-Out Data

Temperature 0.8 provides the strongest held-out performance with stable generation. We vary the temperature from 0.6 to 1.0 with top-p fixed at 1.0, a five-turn limit, and a 32k-token generation limit. Each setting produces one trajectory per held-out target. Table 19 reports the success rate and its two components, together with generation diagnostics. Sampled-token NLL is the mean negative log-probability of generated tokens and measures sampled-token surprisal rather than full-vocabulary entropy.

Table 19: **Inference performance and generation behavior across sampling temperatures.** Results are computed on the frozen 500-target validation split. Blue shading marks the selected configuration; bold marks the highest task metrics and the lowest truncation rate.

	Performance (%)			Generation behavior			
T	Success rate	Stock-complete	Chemical validity	Token NLL	Response tokens	Inventory queries	Truncated (%)
0.6	65.80	76.00	69.60	0.1084	7,055	2.52	0.80
0.7	66.80	77.20	71.40	0.1322	6,911	2.48	0.40
0.8	**68.00**	**77.40**	**71.80**	0.1638	7,063	2.48	**0.20**
0.9	65.00	76.80	70.80	0.2103	7,595	2.47	0.60
1.0	63.00	73.60	70.80	0.2847	7,958	2.43	**0.20**

Temperature 0.8 yields the highest success rate, stock-completion rate, and rate of chemically valid routes. Its 0.20% truncation rate is tied for the lowest among the evaluated temperatures, while its mean response length remains close to the lower-temperature settings and below those at 0.9 and 1.0. Relative to temperature 1.0, it improves paired success by 5.0 percentage points (95% bootstrap CI: 0.2–9.8). Its point estimate also exceeds those at temperatures 0.6, 0.7, and 0.9, although the corresponding paired intervals include zero. We therefore select temperature 0.8 with top-p 1.0 on the held-out split and fix this configuration for the main results in Tables 1 and 2.

F.2.3 Inference Cost

Table 20 characterizes inference cost using parameter count, mean output tokens, and mean inventory queries per trajectory. Evaluations marked with † use the shared 400-target subset, whereas unmarked results use the full benchmark test sets. Output-token counts are capped at the shared 32k trajectory budget before averaging.

Solvent achieves comparable average success to Kimi K3 at a smaller model size and with fewer output tokens: 61.86% versus 63.03%, with 8B versus 2.8T parameters and 9.20k versus 18.45k output tokens per trajectory.

F.2.4 Complementary In Silico Assessments of Route Quality

We complement the primary success criterion with three orthogonal post hoc assessments on the frozen 400-target subset shared by all LLMs in Table 1. These diagnostics examine distinct properties of stock-complete routes: structural simplification, support from documented reaction contexts, and consistency with a learned forward-reaction model. None is used for training, model selection, or the primary chemical-validity assessment. The ChemCensor and round-trip diagnos-

Table 20: **Inference cost of the evaluated LLMs.** Success and usage are macro-averaged; daggers mark the shared 400-target subset, and em dashes denote undisclosed parameter counts.

Method	Params.	Success (%)	Output tokens	Inventory queries
Frontier LLMs (proprietary)				
Claude Opus 4.8†	—	56.54	3.57k	2.73
Gemini 3.5 Flash†	—	58.30	20.33k	2.70
Grok 4.5†	—	59.15	14.30k	1.72
GPT-5.5†	—	63.57	5.32k	2.57
Frontier LLMs (open-weight)				
GLM-5.2†	753 B	3.73	30.50k	0.09
Kimi K2.5†	1 T	27.08	9.77k	2.59
DeepSeek-V4-Pro	1.6 T	42.70	23.94k	2.25
Kimi K3†	2.8 T	63.03	18.45k	1.61
Mid-scale LLMs				
Qwen3-8B	8 B	0.16	3.49k	2.27
Qwen3.5-9B	9 B	0.17	5.67k	2.11
Qwen3.6-Flash†	35 B	11.13	29.80k	0.53
R^3	8 B	19.03	2.33k	0.00
Solvent	8 B	**61.86**	9.20k	2.89

tics apply an all-step route criterion. For a route $\tau = (e_1, \ldots, e_L)$, acceptance is the conjunction $C(\tau) = \prod_{j=1}^{L} c(e_j)$: a single unsupported transformation rejects the complete route. Full-route acceptance therefore becomes more stringent with route length. Under an illustrative independence assumption, a 70% per-step acceptance rate corresponds to only 16.8% over five steps. A low route-level rate therefore does not imply that most individual reactions are unsupported. At the reaction level, recent out-of-distribution retrosynthesis evaluations report round-trip rates ranging from 5% to 30% and show through manual review that recovery depends on the coverage of the forward model (Choe et al., 2026). For transformations outside the recorded reaction space, absence of support should therefore be interpreted as insufficient computational evidence rather than evidence of experimental failure (Bradshaw et al., 2025; Choe et al., 2026). Taken together, these *in silico* diagnostics provide complementary evidence of route plausibility, but do not establish experimental feasibility or substitute for wet-lab validation. Further discussion appears in Appendix H.

Synthetic accessibility. The SA score estimates molecular accessibility from fragment contributions and structural complexity, with lower values indicating more accessible structures (Ertl & Schuffenhauer, 2009). We measure the decrease from the target score to the mean score of the terminal precursors and the fraction of retrosynthetic steps whose precursors are, on average, more accessible than their product. Solvent achieves a target-to-leaf decrease of 0.827, and 84.69% of its steps reduce the SA score. The corresponding values are 0.662 and 66.32% for Kimi K3, and 0.567 and 64.09% for R^3, indicating that Solvent more consistently decomposes targets into structurally accessible precursors.

ChemCensor. ChemCensor evaluates whether a proposed reaction center and its functional-group context are supported by documented reaction precedents (Zagribelnyy et al., 2026). We use ChemCensor v1.2.0 and report the number of stock-complete routes for which every step passes the reaction-center (RC) or functional-group (FG) criterion per 100 evaluated reaction steps. This normalization rewards complete, consistently supported routes while accounting for the amount of explicit reaction construction. Solvent obtains 8.62 RC-supported and 4.11 FG-supported routes per 100 steps, comparable to Kimi K3 (8.03 and 3.65), Claude Opus 4.8 (8.35 and 3.49), and Grok 4.5 (8.48 and 4.48). Disabling ChemCensor's exact-reaction lookup leaves Solvent's results unchanged and alters every model's route-level rate by at most 0.31 percentage points. The evidence therefore reflects compatibility with broader reaction contexts rather than exact database overlap.

Table 21: **Complementary *in silico* assessments on the shared 400-target subset.** Stock routes are stock-complete. ΔSA is the decrease from target SA to mean leaf SA; reducing steps have a lower mean precursor SA than product SA. ChemCensor columns report all-step-supported stock-complete routes per 100 evaluated reaction steps, macro-averaged across benchmarks. Route RT is the percentage of all 400 targets whose routes pass exact top-20 forward recovery at every step.

			Synthetic accessibility		ChemCensor		Round trip
Method	Params.	Stock routes (/400)	Target–leaf ΔSA	Reducing steps (%)	RC routes /100 steps	FG routes /100 steps	Route RT@20 (%)
Frontier LLMs (proprietary)							
Claude Opus 4.8	—	304	0.611	70.10	8.35	3.49	14.42
Gemini 3.5 Flash	—	300	0.607	68.60	9.89	6.02	15.75
Grok 4.5	—	295	0.618	63.89	8.48	4.48	13.06
GPT-5.5	—	289	0.539	66.57	9.98	5.33	10.90
Frontier LLMs (open-weight)							
GLM-5.2	753 B	18	0.890	73.01	22.90	12.84	1.22
Kimi K2.5	1 T	235	0.539	68.23	2.42	0.87	2.50
DeepSeek-V4-Pro	1.6 T	307	0.711	68.67	4.95	1.86	7.52
Kimi K3	2.8 T	309	0.662	66.32	8.03	3.65	13.44
Mid-scale LLMs							
Qwen3-8B	8 B	13	0.787	78.24	0.00	0.00	0.00
Qwen3.5-9B	9 B	2	0.576	75.00	0.00	0.00	0.00
Qwen3.6-Flash	35 B	62	0.750	78.73	6.98	1.54	0.78
R^3	8 B	100	0.567	64.09	6.53	4.75	4.03
Solvent	8 B	293	0.827	84.69	8.62	4.11	9.02

Table 22: **Orthogonal assessments grouped by primary chemical validity.** We split 293 stock-complete Solvent routes into routes that pass and routes that do not pass the primary chemical-validity assessment. Each row reports the mean score of an independent diagnostic in the two groups. Difference is the mean for primary-pass routes minus the mean for primary-non-pass routes; 95% confidence intervals use 10,000 benchmark-stratified bootstrap resamples.

Orthogonal diagnostic	Primary pass (%)	Primary non-pass (%)	Difference (95% CI)
ChemCensor RC step support	59.72	45.92	+13.80 (6.65, 21.15)
ChemCensor FG step support	45.14	29.90	+15.24 (7.67, 22.76)
Exact route RT@20	14.98	0.00	+14.98 (10.67, 19.50)

Forward round-trip consistency. We additionally apply the USPTO-50K-finetuned Chemformer forward model (Irwin et al., 2022) to each proposed step and test whether the original product is recovered among its top-20 predictions. We assess route-level consistency by requiring recovery at every step. Solvent reaches 9.02%, exceeding DeepSeek-V4-Pro (7.52%) and R^3 (4.03%), approaching GPT-5.5 (10.90%), and falling within the range of the frontier LLMs in Table 21.

Consistency of the primary chemical-validity assessment. We assess agreement on the 293 stock-complete Solvent routes, thereby holding inventory completion fixed. Routes accepted by the primary chemical-validity assessment receive consistently stronger ChemCensor support and forward-model recovery than routes not accepted by it (Table 22). The differences remain positive after joint stratification by benchmark and route length. These convergent results support the practical discriminative value of the primary assessment while preserving the distinct scope of each *in silico* diagnostic.

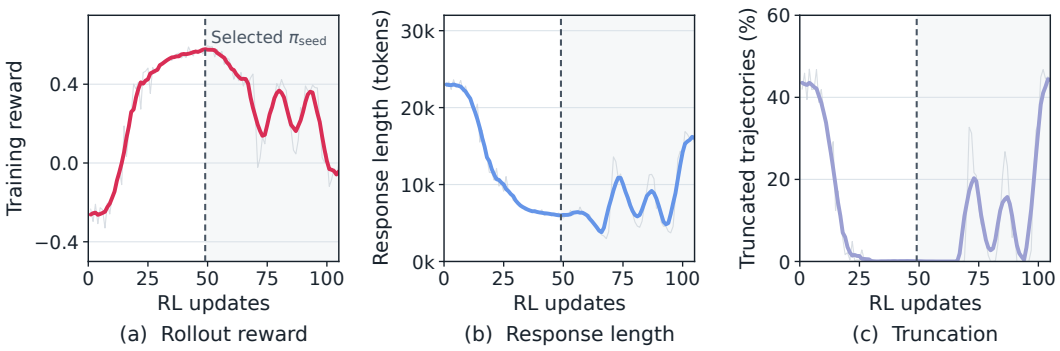


Figure 9: **Round 1 RL stabilizes long-horizon interaction.** (a) Training reward, (b) mean response length, and (c) the fraction of truncated trajectories. Faint lines show per-update measurements and colored lines show centered seven-update moving averages. The dashed line marks the selected π_{seed}; the shaded region denotes the exploratory continuation beyond this checkpoint.

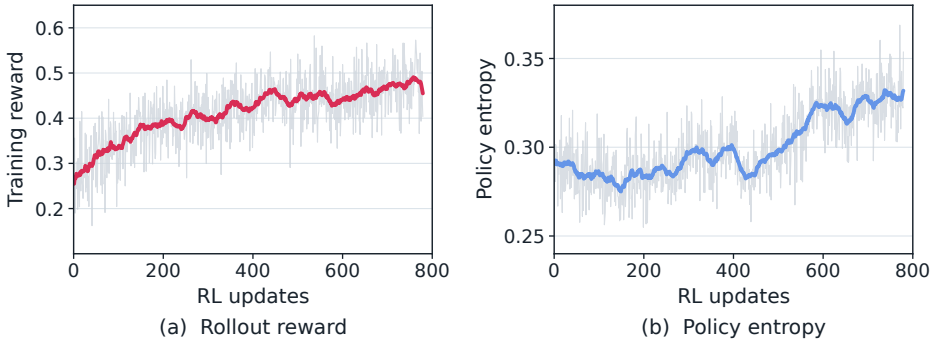


Figure 10: **Reward and entropy during final-stage RL.** (a) Mean rollout reward and (b) policy entropy. Faint lines show per-update measurements and colored lines show centered 25-update moving averages.

F.2.5 Behavioral Dynamics of the Two RL Stages

Round 1 RL stabilizes long-horizon interaction. Figure 9 traces first-round RL through the checkpoint selected as π_{seed} and an exploratory continuation beyond it. Across the first 49 updates, five-update averages show reward rising from -0.255 to 0.569, while mean response length falls from 22.9k to 6.1k tokens and truncation from 43.2% to 0.12%. Continued optimization reverses these trends. We therefore use the update-49 checkpoint as π_{seed} for subsequent trajectory collection.

Final-stage RL improves reward while maintaining policy entropy. As shown in Figure 10, mean rollout reward increases throughout final-stage RL, from 0.274 over the first 25 updates to 0.484 over the last 25. Meanwhile, policy entropy is maintained throughout training as rollout reward improves.

G Representative Case Studies

G.1 Strategic Retrosynthetic Reasoning and Feedback-Guided Revision

G.1.1 Inventory-Guided Convergent Route Planning

This example illustrates how Solvent revises a high-level disconnection in response to precursor availability and then develops both branches of a convergent synthesis. The target is an unsymmetrical diaryl urea containing bis(morpholinyl)pyrimidine and chlorothiophene fragments (Figure 11).

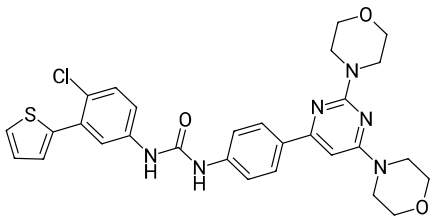


Figure 11: Target molecule for the convergent-route case study.

Target SMILES.

```
O=C(Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1)Nc1ccc(Cl)c(-c2cccs2)c1
```

Model trajectory. The selected verbatim excerpts retain the structural analysis, inventory-guided revisions, and final route review; bracketed ellipses mark omitted intermediate analysis and tool payloads.

Selected reasoning trace

```
Let me analyze the target molecule step by step.

Let me parse this molecule. It has:
- A urea linkage: O=C(N-)...N-
- One side of the urea is attached to a para-substituted phenyl ring: Nc1ccc(-c2cc(N3CCOCC3)nc
    (N3CCOCC3)n2)cc1
- The other side of the urea is attached to another para-substituted phenyl ring: Nc1ccc(Cl)c
    (-c2cccs2)c1

Let me think about retrosynthetic disconnections:

Step 1: Disconnect the urea linkage. The urea can be formed by reacting an amine with an
    isocyanate, or by reacting two amines with a carbonyl source (like CDI).

[...]

Both precursors are not available in eMolecules. I need to revise the route to use available
    starting materials.

Option 1: Disconnect the C-N bonds of the urea differently. Instead of isocyanate + amine, I
    could use CDI (carbonyldiimidazole) to couple the two amines.

[...]

Neither of the amines is available. I need to decompose them further. Let me think about
    constructing the two aniline fragments.

Fragment 1: Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1

Fragment 2: Nc1ccc(Cl)c(-c2cccs2)c1

[...]

Disconnection strategy:
1. Urea → two amines + CDI
2. Each amine → simpler precursors via Suzuki coupling

[...]

Both amines are unavailable. I need to decompose each into simpler precursors. Let me
    disconnect each aniline fragment via Suzuki coupling and then SNAr reactions on
    pyrimidine.

[... inventory query and response omitted ...]

The stock check returned stock_ok=true, meaning all leaf materials are available in eMolecules
    . The route is chemically plausible. Let me finalize.

Let me review the route:

Step 1: Urea formation
```

```
Target: O=C(Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1)Nc1ccc(Cl)c(-c2cccs2)c1
>> Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1.Nc1ccc(Cl)c(-c2cccs2)c1.O=C(N1C=CN=C1)N1C=CN=C1

This is a CDI-mediated urea formation. The two amines react with CDI to form the urea.

Step 2: Suzuki coupling for Fragment 1
Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1
>> Nc1ccc(B(O)O)cc1.Clc1cc(N2CCOCC2)nc(N2CCOCC2)n1

Step 3: SNAr on pyrimidine
Clc1cc(N2CCOCC2)nc(N2CCOCC2)n1
>> Clc1cc(Cl)nc(Cl)n1.C1COCCN1.C1COCCN1

Step 4: Suzuki coupling for Fragment 2
Nc1ccc(Cl)c(-c2cccs2)c1
>> Nc1ccc(Cl)c(Br)c1.OB(O)c1cccs1

The biaryl bond is formed via Suzuki coupling between 4-chloro-3-bromoaniline and thiophen-2-
    ylboronic acid.

All leaf materials are available. The route is chemically plausible.
```

The three inventory queries progressively expand the route until every terminal precursor is available:

Table 23: Inventory feedback during route revision.

Query	Proposed route boundary	Available leaves
1	Aniline and aryl isocyanate	0/2
2	Two anilines and CDI	1/3
3	Both aniline branches expanded to simpler precursors	6/6

Figure 12 renders the accepted route in forward synthetic order. The two aniline fragments are prepared independently and joined through the central urea in the final step.

Final answer

```
["O=C(Nc1ccc(-c2cc(N3CCOCC3)nc(N3CCOCC3)n2)cc1)Nc1ccc(Cl)c(-c2cccs2)c1>>Nc1ccc(-c2cc(N3CCOCC3)
    nc(N3CCOCC3)n2)cc1.Nc1ccc(Cl)c(-c2cccs2)c1.O=C(N1C=CN=C1)N1C=CN=C1", "Nc1ccc(-c2cc(
    N3CCOCC3)nc(N3CCOCC3)n2)cc1>>Nc1ccc(B(O)O)cc1.Clc1cc(N2CCOCC2)nc(N2CCOCC2)n1", "Clc1cc(
    N2CCOCC2)nc(N2CCOCC2)n1>>Clc1cc(Cl)nc(Cl)n1.C1COCCN1.C1COCCN1", "Nc1ccc(Cl)c(-c2cccs2)c1
    >>Nc1ccc(Cl)c(Br)c1.OB(O)c1cccs1"]
```

Manual chemical review. The route was also examined transformation by transformation. (1) The electron-deficient trichloropyrimidine provides a credible substrate for sequential substitution by morpholine, although the two substitutions would ordinarily require staged condition control. (2) Coupling the residual chloropyrimidine with 4-aminophenylboronic acid forms the first aryl fragment through a Suzuki reaction. (3) The second Suzuki reaction exploits the greater reactivity of the aryl bromide while retaining the aryl chloride in the product. (4) CDI provides a plausible carbonyl source for staged coupling of the two anilines into the unsymmetrical urea. As complementary computational evidence, all four transformations pass ChemCensor's reaction-center criterion, three pass its functional-group criterion, and an independently trained Chemformer model recovers every product within its top-5 forward predictions.

Interpretation. The trajectory begins with a concise urea disconnection, abandons an unavailable isocyanate route, and independently expands both aryl branches before returning a stock-complete convergent plan. The value of the example lies in this coordination of global assembly, branch-specific reaction selection, and revision from inventory evidence rather than in any single isolated transformation.

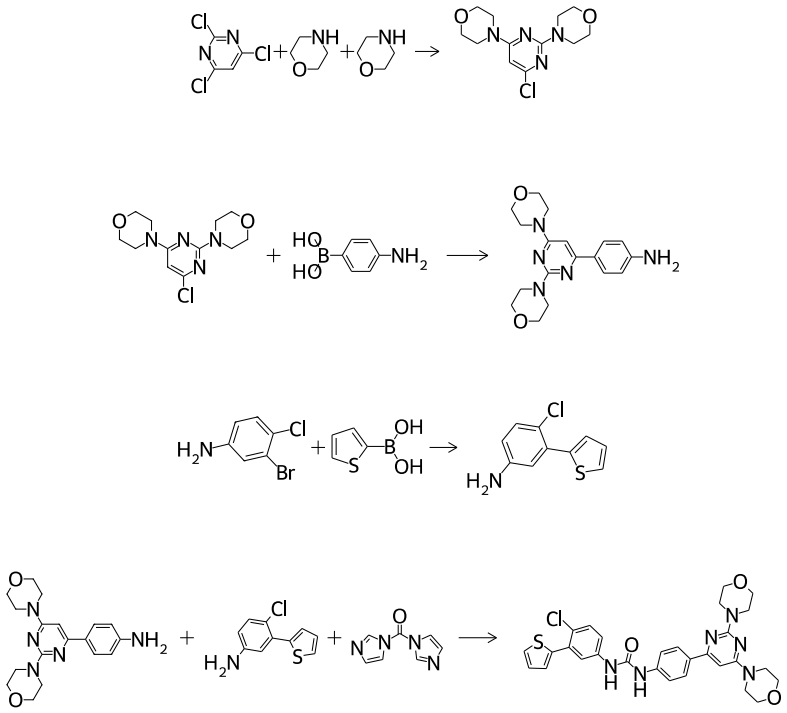


Figure 12: Inventory-verified convergent route rendered in forward synthetic order.

G.1.2 Boc Protection Enables Chemoselective Piperazine Functionalization

This example illustrates how inventory feedback prompts a concrete protecting-group strategy. Solvent initially proposes 1-(4-methoxyphenyl)piperazine as a terminal precursor, but the inventory reports that it is unavailable. The model then constructs this intermediate from N-Boc-piperazine and 4-bromoanisole, retaining the Boc group during selective N-arylation and removing it before the remaining piperazine nitrogen is coupled to the activated maleimide.

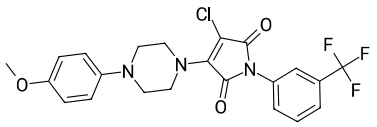


Figure 13: Target molecule for the Boc-protection case study.

Target SMILES.

```
COc1ccc(N2CCN(C3=C(Cl)C(=O)N(c4cccc(C(F)(F)F)c4)C3=O)CC2)cc1
```

Model trajectory. The selected verbatim excerpt preserves the failed inventory query, the protecting-group decision, the revised route, and the final verification; bracketed ellipses mark omitted intermediate analysis.

Inventory-guided protecting-group revision

```
The 4-methoxyphenylpiperazine (COc1ccc(N2CCNCC2)cc1) is not in eMolecules stock. I need to
    revise the route to use a different available piperazine derivative.
```

```
[...]

Option 2: Use a protected piperazine (e.g., Boc-protected) and deprotect later.

[...]

Route:
Step 1: COc1ccc(N2CCN(C3=C(Cl)C(=O)N(c4cccc(C(F)(F)F)c4)C3=O)CC2)cc1>>COc1ccc(N2CCNCC2)cc1.O=
    C1C(Cl)=C(Cl)C(=O)N1c1cccc(C(F)(F)F)c1
Step 2: COc1ccc(N2CCNCC2)cc1>>CC(C)(C)OC(=O)N1CCN(c2ccc(OC)cc2)CC1
Step 3: CC(C)(C)OC(=O)N1CCN(c2ccc(OC)cc2)CC1>>CC(C)(C)OC(=O)N1CCNCC1.COc1ccc(Br)cc1

Let me check this route.

[...]

The stock check returned 'stock_ok: true' -- all three leaf materials are available in
    eMolecules.
```

The first query identifies 1-(4-methoxyphenyl)piperazine as the sole unavailable leaf. In the revised route, all three terminal precursors—N-Boc-piperazine, 4-bromoanisole, and the N-aryl dichloromaleimide—are available, and the second query verifies the complete route.

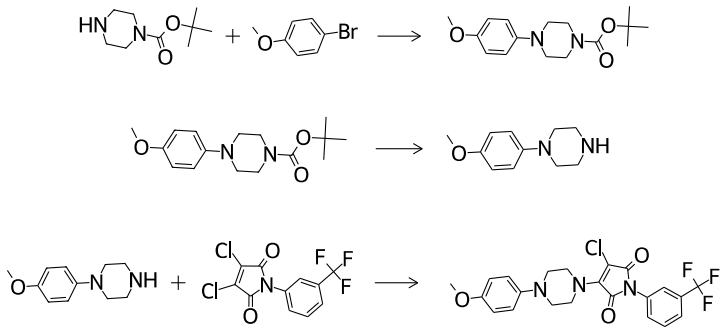


Figure 14: Inventory-verified protecting-group route rendered in forward synthetic order.

Final answer

```
[
  "COc1ccc(N2CCN(C3=C(Cl)C(=O)N(c4cccc(C(F)(F)F)c4)C3=O)CC2)cc1>>COc1ccc(N2CCNCC2)cc1.O=C1C(Cl
     )=C(Cl)C(=O)N1c1cccc(C(F)(F)F)c1",
  "COc1ccc(N2CCNCC2)cc1>>CC(C)(C)OC(=O)N1CCN(c2ccc(OC)cc2)CC1",
  "CC(C)(C)OC(=O)N1CCN(c2ccc(OC)cc2)CC1>>CC(C)(C)OC(=O)N1CCNCC1.COc1ccc(Br)cc1"
]
```

Manual chemical review. We examined the route transformation by transformation in the forward direction. (1) Mono-Boc piperazine exposes one nitrogen for palladium-catalyzed N-arylation with 4-bromoanisole while blocking reaction at the second nitrogen. (2) Standard acidic Boc cleavage then reveals 1-(4-methoxyphenyl)piperazine without disturbing the anisole substituent. (3) The remaining secondary amine can displace one vinylic chloride from the electron-poor N-aryl 3,4-dichloromaleimide; the two adjacent carbonyl groups activate this substitution, while the second chloride is retained in the target. The sequence therefore assigns the Boc group a chemically specific role: it controls which piperazine nitrogen is functionalized first and exposes the other only when required for the final bond formation. The primary chemical-validity assessment accepts all three transformations. As complementary orthogonal evidence, ChemCensor also supports every step under both its reaction-center and functional-group criteria.

Interpretation. The protecting group is introduced in direct response to an unavailable precursor rather than added as an incidental decoration. Solvent preserves it through the selective arylation, removes it at the appropriate point, and then uses the newly exposed nitrogen in the final conver-

gent substitution. This case provides a concrete example of inventory-guided revision yielding a chemically purposeful, stock-complete synthesis strategy.

G.2 Solvent Generates Diverse Plausible Routes for the Same Target

The preceding examples show how Solvent revises a route within a single trajectory. Beyond this iterative behavior, Solvent can propose different chemically plausible synthesis plans for the same target. Figure 15 presents stock-complete routes obtained in independent trials that differ in both the reaction used to form the benzylic C–N bond and the available aromatic precursor. We reproduce only the decisive verbatim excerpts; bracketed ellipses indicate omitted reasoning.

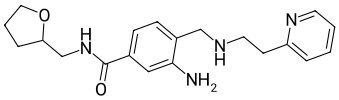


Figure 15: Target molecule shared across independently sampled trajectories.

Target SMILES.

```
Nc1cc(C(=O)NCC2CCCO2)ccc1CNCCc1ccccn1
```

G.2.1 Route A: Benzyl Substitution

After an aldehyde-based proposal fails the inventory check, this trajectory changes the key benzylic C–N disconnection from reductive amination to substitution with a benzyl bromide.

Selected reasoning excerpt

```
The aldehyde Nc1cc(C(=O)O)ccc1C=O is not available. I need to revise the route.
[...]
Instead of reductive amination, I could use a different strategy:
1. The amine could be formed by SN2 reaction of a benzyl halide with the amine.
[...]
What if the benzyl bromide is available? Let me try route with benzyl bromide:
```

The revised route first forms the aryl amide and then couples the resulting benzyl bromide with 2-(pyridin-2-yl)ethan-1-amine (Figure 16). The subsequent inventory query confirms that all three terminal precursors are available.

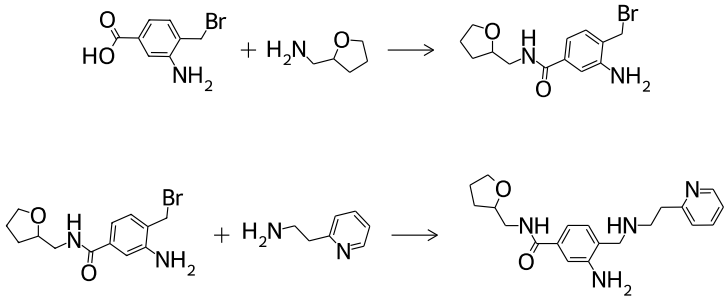


Figure 16: Stock-complete route based on benzyl substitution, rendered in forward synthetic order.

G.2.2 Route B: Reductive Amination

In another trial, Solvent retains the aldehyde disconnection but replaces the unavailable acid with an available methyl ester, producing a different route to the same target.

Selected reasoning excerpt

```
The secondary amine could be formed by reductive amination between an aldehyde (at position 4
    of the benzene) and the primary amine NCCc1ccccn1.
[...]
Actually, let me check the simplest alternatives. Let me try the route using methyl ester
    instead and see:
[...]
The methyl ester Nc1cc(C(=O)OC)ccc1C=O might be available. Let me try.
```

In this proposal, aminolysis of the aromatic ester furnishes the amide-bearing aldehyde, which is then coupled with 2-(pyridin-2-yl)ethan-1-amine by reductive amination (Figure 17). The inventory query again confirms the availability of all three terminal precursors.

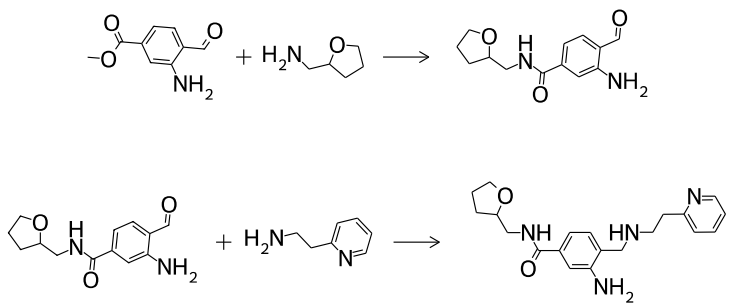


Figure 17: Stock-complete route centered on reductive amination, rendered in forward synthetic order.

Interpretation. The routes construct the same benzylic C–N bond through different chemistry: one uses nucleophilic substitution of a benzyl bromide, whereas the other uses reductive amination of an aldehyde. They consequently rely on different aromatic precursors from the inventory. Together, these trials show that Solvent can generate chemically plausible synthesis plans with distinct reaction strategies for the same target.

G.3 Representative Cases of Long-Horizon Reasoning across RL Stages

For each case, we compare outputs from policies before and after an RL stage on the same target. The first comparison covers Round 1 RL and shows a shift from unstable teacher-guided traces to concise and reliable interaction. The second covers final-stage RL and shows the post-RL policy incorporating a protection-group strategy into the submitted route. These examples provide qualitative views of how Solvent's long-horizon behavior changes after RL; bracketed ellipses mark omitted text.

G.3.1 From Unstable Imitation to Concise, Reliable Reasoning

Round 1 RL markedly improves the reliability of model-generated trajectories. Under matched decoding, we compare 12,896 paired trajectories from the teacher-guided SFT policy and π_{seed} (Table 24). RL reduces truncation from 78.27% to 0.79% and malformed or incomplete interactions from 71.32% to 39.69%, while increasing the final-answer parse rate from 12.93% to 70.91%.

Table 24: Rollout reliability before and after Round 1 RL under matched decoding. Malformed or incomplete interactions fail the tool-trace validity check.

Policy	Truncated (%)	Parseable final answer (%)	Malformed or incomplete interaction (%)
Teacher-guided SFT	78.27	12.93	71.32
π_{seed}	0.79	70.91	39.69

A comparison on the same target illustrates this transition. The SFT policy generates 31,964 tokens and issues three unsuccessful inventory queries without producing a final answer. In contrast, π_{seed} returns a stock-complete route after 2,856 tokens and a single inventory query.

Teacher-guided SFT: repeated speculative restarts

```
This is getting too complicated. Let me try the direct glycine route:
Step 2 revised:
The amino amide can be made from glycine methyl ester, then:
1. N-Boc-glycine methyl ester + NaHMDS → anion at alpha
2. Add 3-cyclohexylpropyl bromide to make N-(3-cyclohexylpropyl)-N-Boc-glycine methyl ester
     with loss of Br
3. Hydrolyze → N-(3-cyclohexylpropyl)-N-Boc-glycine
4. Further hydrolysis of ester (or use directly as acid)
5. Amidation with NH3 (HATU, etc.) → NH2-CH(CCC-C6H11)-CONH2
6. Deprotect Boc → N[C@@H](CCC1CCCCC1)C(N)=O
But step 5 has matching issues... we'd get NH2-C(CCC-C6H11)-CONH2. The product has NH-C(CONH2)
     -R. So we need the amine on the alpha carbon, and the alkyl group also on the alpha
     carbon. That means the two groups are on adjacent carbons (like CONH-CH(R)-CONH2, where
     the amine carbon and the alkyl carbon are the same? No, they're distinct alpha and alkoxy
      positions.)
Actually, in the target: CONH-CH(CCC-C6H11)-CONH2
The CONH- is attached to the alpha carbon (the one bearing the R group). There's a CH2 group
     adjacent? No, there's only one CH between the two CONH groups: the alpha carbon has both
     CONH and R, and the beta position has H. So the connectivity is CONH-CH(R)-CONH2.
So the alpha carbon is CH(CCC-cyclohexyl) and the beta position is CH2. We need:
N-C(=O)-CH(R)-C(=O)-NH2
From glycine: N-Boc-CH2-CO2H → anion at alpha → alkylate at the alpha carbon with BrCCC-
     C6H11 → Boc-N-alkylglycine → deprotonate at alpha → add tert-butyl amide (or cyanate
     equivalent) → cyclization → ring opening → amidation...
This is more like building a pyrrolidine ring and then opening it. Not great.
Let me try a completely different disconnection:
What if the central C-N bond is formed by nucleophilic attack of the aniline on an activated
     carbonyl?
Actually, N-alkyl glycine derivatives. In my route, I had:
Target = Ar-CONH-CH(R)-CONH2
If R-CH2-CH2-CH2-cyclohexanone is available (as a protected form), we could:
1. Reductive amination: aniline + ketone → imine → reduction → HN-CH(R)-CONH2
2. But we need the aniline NH2, not HNAr.
Or: use a different reagent like N-(3-cyclohexylpropyl)glycine benzyl ester, then Pd-catalyzed
      C-H deprotonation at methylene and amination?
This is getting very creative but doubtful about commercial availability.
Let me try using more straightforward building blocks. What if we use:
'NCC(=O)O' (glycine) + 'OCCCC1CCCCC1' (3-cyclohexylpropan-1-ol) as starting materials?
Reductive amination: Glycine + 3-cyclohexylpropanol → imine → NaBH4 → HO-CH2-CH2-CH2-C6H11
Then oxidation of alcohol to acid:→ HOOC-CH2-CH2-CH2-C6H11 (3-cyclohexylpropionic acid)
This acid + azodicarboxylic acid + NH3 → Curtius rearrangement? No, that gives isocyanate
     directly attached to ring.
Or: Use the acid chloride: acid + SOCl2 → acid chloride, then hydrazine + CDI → acyl
     carbazate → rearrangement... too many steps.
OK let me try a completely ridiculous route and check: the amino amide could be a leaf
     material if it IS commercially available in a protected form. Let me try just adding it
     to the route without building it from scratch.
Actually no, the tool doesn't check whether individual reagents are in stock; it only checks
     leaf organic materials. So if I propose the amino amide as a leaf material, it should be
     checked.
Let me try a different approach for the amide disconnections. Instead of disconnecting both
     amides to get to these complex leaves, what if I disconnect one bond and keep the amino
     amide intact?
```

After Round 1 RL: route construction and verification

```
Let me analyze the target molecule step by step.
**Target SMILES:** 'Cc1ccc(C(=O)NC(CCC2CCCCC2)C(N)=O)cc1NC(=O)c1cnn(-c2ccccc2)c1'
**Structural analysis:**
- Central 1,3,4-trisubstituted benzene ring
- Position 1: methyl
- Position 3: amide with an amine fragment '--C(=O)NH--CH2--CH2--cyclohexyl--C(=NH2)'
- Position 4: amide with a pyrazole carboxylic acid fragment '--NH--C(=O)--c1cnn(Ph)c1'
**Retrosynthetic strategy:**
1. **First disconnection:** Amide bond between the benzoic acid derivative and the α-amino
     amide
2. **Second disconnection:** Amide bond between the aniline and the 1-phenylpyrazole-4-
     carboxylic acid
Let me check stock availability for these leaf materials.
</think>
<tool_call>
```

{"name": "check_emolecules_stock", "arguments": {"answer_route": "[\"Cc1ccc(C(=O)NC(CCC2CCCCC2)C(N)=O)cc1NC(=O)c1cnn(-c2ccccc2)c1>>Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1.NC(CCC2CCCCC2)C(N)=O\", \"Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1>>Cc1ccc(C(=O)O)cc1N.O=C(O)c1cnn(-c2ccccc2)c1\"]"}}
</tool_call><|im_end|><|im_start|>user
<tool_response>
{"stock_ok": true, "leaf_count": 3, "missing_count": 0, "missing": [], "leaves": [{"leaf": "Cc1ccc(C(=O)O)cc1N", "canonical_parent": "Cc1ccc(C(=O)O)cc1N", "in_emolecules": true}, {"leaf": "NC(CCC2CCCCC2)C(N)=O", "canonical_parent": "NC(=O)C(N)CCC1CCCCC1", "in_emolecules": true}, {"leaf": "O=C(O)c1cnn(-c2ccccc2)c1", "canonical_parent": "O=C(O)c1cnn(-c2ccccc2)c1", "in_emolecules": true}], "all_nodes": [{"node": "Cc1ccc(C(=O)NC(CCC2CCCCC2)C(N)=O)cc1NC(=O)c1cnn(-c2ccccc2)c1", "canonical_parent": "Cc1ccc(C(=O)NC(CCC2CCCCC2)C(N)=O)cc1NC(=O)c1cnn(-c2ccccc2)c1", "is_leaf": false}, {"node": "Cc1ccc(C(=O)O)cc1N", "canonical_parent": "Cc1ccc(C(=O)O)cc1N", "is_leaf": true}, {"node": "Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1", "canonical_parent": "Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1", "is_leaf": false}, {"node": "NC(CCC2CCCCC2)C(N)=O", "canonical_parent": "NC(=O)C(N)CCC1CCCCC1", "is_leaf": true}, {"node": "O=C(O)c1cnn(-c2ccccc2)c1", "canonical_parent": "O=C(O)c1cnn(-c2ccccc2)c1", "is_leaf": true}], "interpretation": "stock_ok=false means at least one predicted leaf is not locally available in eMolecules and the route must be revised. stock_ok=true only checks availability of predicted leaves; it is not a correctness check."}
</tool_response><|im_end|>
<|im_start|>assistant
<think>
The stock check returned `stock_ok: true`, meaning all three leaf materials are available in the eMolecules database:
1. `Cc1ccc(C(=O)O)cc1N` (4-amino-3-methylbenzoic acid)
2. `NC(CCC2CCCCC2)C(N)=O` (2-amino-4-cyclohexylbutanamide)
3. `O=C(O)c1cnn(-c2ccccc2)c1` (1-phenyl-1H-pyrazole-4-carboxylic acid)
The route is chemically plausible:
- Step 1: Amide coupling between 4-amino-3-methylbenzoic acid and 2-amino-4-cyclohexylbutanamide
- Step 2: Amide coupling between 4-amino-3-methylbenzoic acid and 1-phenyl-1H-pyrazole-4-carboxylic acid
Both disconnections are standard amide bond formations. The route is valid and all leaf materials are in stock.
Let me finalize the answer.
All three leaf materials are available in eMolecules stock. The route is chemically sound and all starting materials are accessible. Let me now provide the final answer.
</think>
<answer>["Cc1ccc(C(=O)NC(CCC2CCCCC2)C(N)=O)cc1NC(=O)c1cnn(-c2ccccc2)c1>>Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1.NC(CCC2CCCCC2)C(N)=O", "Cc1ccc(C(=O)O)cc1NC(=O)c1cnn(-c2ccccc2)c1>>Cc1ccc(C(=O)O)cc1N.O=C(O)c1cnn(-c2ccccc2)c1"]</answer><|im_end|>

Interpretation. The SFT trace cycles through increasingly speculative constructions without converging on an executable route. In contrast, π_{seed} identifies two amide disconnections and submits the resulting route for inventory verification. Once precursor availability is confirmed, the policy returns the route without revisiting its plan. This case mirrors the aggregate transition in Table 24 toward concise reasoning and reliable tool use.

G.3.2 A Protection-Group Strategy after Final-Stage RL

We next compare two policies on the same temporary-protection target. Before final-stage RL, the model identifies Boc protection as a way to control the free piperidine nitrogen but does not incorporate it into the final route. The submitted four-step route is stock-complete but fails the chemical-validity assessment.

Round 2 bootstrapping: selected verbatim excerpt

```
Actually, let me think about this differently. The first reasonable disconnection is:

1. N-dealkylation of the benzyl group from the piperidine nitrogen:
   CN(Cc1ccc(C(F)(F)F)cc1N1CC2CC1CN2CC(=O)O)C1(C)CCNCC1 >> CN(Cc1ccc(C(F)(F)F)cc1N1CC2CC1CN2CC
     (=O)O)C1(C)CCN(C(=O)OC(C)(C)C)CC1
   (Boc protection of the piperidine nitrogen)

[...]

Let me review the route for chemical plausibility:

Step 1: N-alkylation of 2,5-diazabicyclo[2.2.1]heptane with acetic acid. This is a reasonable
     reaction, though acetic acid as a nucleophile is atypical

[...]
```

```
Step 2: SNAr reaction between 2,5-diazabicyclo[2.2.1]heptane and 2-bromo-5-(trifluoromethyl)
    phenyl bromide.
```

Table 25: Protection-group strategy before and after final-stage RL.

Policy	Tokens	Steps	Stock-complete	Chemical validity
Round 2 bootstrapping	6,449	4	Yes	No
Final-stage RL	3,935	3	Yes	Yes

After final-stage RL, the policy retains the Boc-protected piperidine through reductive amination and removes the protecting group only after the selective bond formation. The complete final-stage trajectory and route are shown in Figure 18. The revised route provides a concrete illustration of a more mature chemical strategy after final-stage RL: temporary protection is not merely considered but incorporated at the appropriate stage, enabling selective bond formation within a chemically valid route.

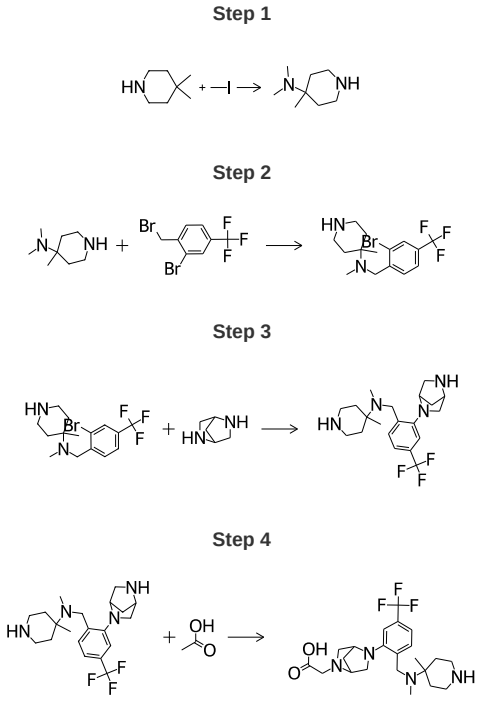
Figure 18 compares the two submitted routes.

H Further Discussion

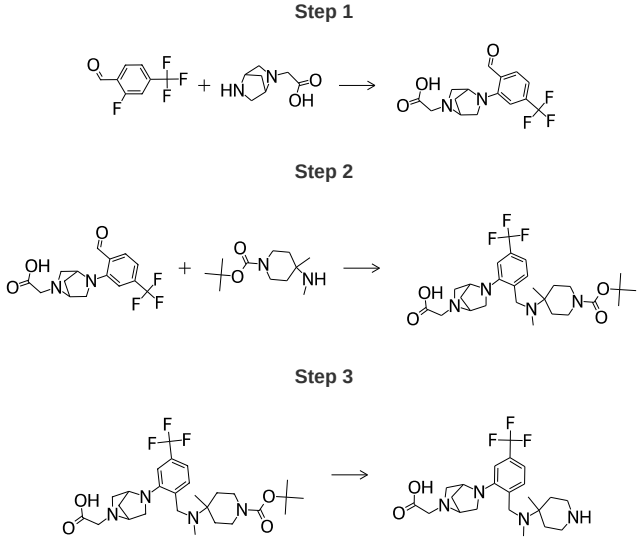
Framing multi-step retrosynthesis as a long-horizon task gives the LLM direct control over global route planning and stepwise disconnection decisions, allowing the model to explore routes beyond the candidates supplied by an external single-step model while making reliable route verification essential, which however, remains a general open challenge in multi-step retrosynthesis (Choe et al., 2025). Our post-generation evaluator combines a permissive reaction-support check with atom-source consistency, providing a practical screen for evident chemical inconsistencies while leaving route generation unrestricted.

Passing these checks and terminating in available building blocks, however, does not establish that a route will succeed experimentally. Laboratory outcomes depend on reaction conditions, chemo- and stereoselectivity, competing pathways, intermediate stability, purification, reproducibility, and scale. The practical value of a route also reflects overall yield, material cost, safety, and environmental impact (Genheden & Shields, 2025). These factors cannot be fully assessed in silico and ultimately require experimental validation.

An important extension is to carry experimental observations across synthesis cycles and use them in later route decisions. ACRA demonstrates that an LLM-based workflow can extract synthetic procedures, translate them into executable robotic instructions, and verify them through real experiments (Pagel et al., 2026). Integrating this experimental interface into Solvent's autonomous long-horizon planning loop would enable more direct evaluation of route outcomes, allow Solvent to adapt its strategy to experimental feedback, and potentially raise its performance ceiling while increasing its practical utility.



(a) Round 2 bootstrapping



(b) Final-stage RL

Figure 18: Routes proposed before and after final-stage RL, with steps shown in forward synthetic order. (a) The Round 2 bootstrapping route is stock-complete but fails the chemical-validity check because its initial methylation does not account for the exocyclic nitrogen. (b) The final-stage RL route carries Boc protection through the selective bond formation and removes it only in the final step.