RETROPLANNER: MULTI-ROUTE EXPANSION AND MULTI-OBJECTIVE REASONING FOR LLM-BASED RETROSYNTHESIS PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-step retrosynthesis planning is often considered solved once a route to purchasable starting materials is found, but such a route should also be practical in real-world chemistry. Recent LLM-based planners formulate this process as sequential decision-making over an AND-OR graph, yet typically expand a single molecule per turn, which limits global comparison across alternative routes before the search budget is committed. We introduce RETROPLANNER, an LLM-based planner that jointly compares multiple routes to efficiently search toward higher synthesizability, as measured by reaction plausibility, round-trip consistency, and starting-material price. RETROPLANNER is trained to reason over chemical evidence: structural progress from the target molecule, bond changes and reaction classes, where candidate routes diverge from one another, and how they rank across multiple objectives. Experiments on multi-step retrosynthesis benchmarks demonstrate that RETROPLANNER recovers more alternative routes than prior LLM-based planners while improving synthetic route quality and maintaining a high success rate.

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.

1 INTRODUCTION

Multi-step retrosynthesis planning is a fundamental task in organic synthesis and drug discovery that identifies a sequence of reactions connecting a target molecule to starting materials available in a set of purchasable building blocks (Corey & Wipke, 1969). This task is commonly formulated as a search over an AND-OR graph (Schreck et al., 2019; Chen et al., 2020), in which a single-step retrosynthesis model proposes candidate precursors and a planner determines the next molecule to expand. A planner's effectiveness depends on how it explores the search space within the available expansion budget and on how it maintains the quality of the routes it recovers. Conventional planners control the search procedure using predefined algorithms (Segler et al., 2018; Kishimoto et al., 2019; Xie et al., 2022) or learned value functions that estimate the future cost from a candidate (Liu et al., 2023; Han et al., 2022), allocating the budget by a static rule aimed only at reaching building blocks. Recent studies turned to treating route practicality as an explicit objective, formulating planning as a multi-objective search over criteria beyond success rate (Hastedt et al., 2026), with benchmarks following the same shift by asking whether each reaction along a route has synthetic precedent (Zagribelnyy et al., 2026).

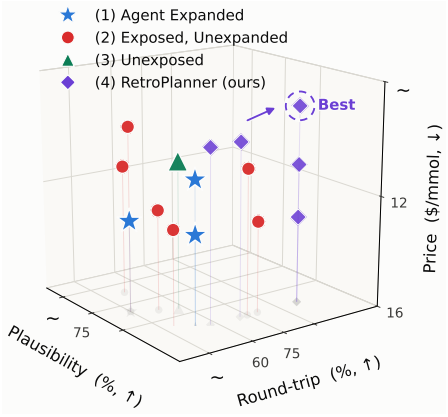

Figure 1: Many routes that dominate the agent's solutions in reaction plausibility, round-trip consistency, and starting-material price are already exposed during search but left unexpanded. RETROPLANNER recovers such routes by jointly expanding multiple frontier molecules and reasoning over them in a multi-objective decision within the same turn.

Large language models have recently been used as agentic planners that interact with the single-step model through a sequence of tool calls (Liu et al., 2026; Zhu et al., 2026; Chang et al., 2026). Existing agents may attach multiple reaction candidates when expanding a molecule, but each turn still commits the search to a single frontier molecule. This makes it difficult to compare competing routes under the limited search budget, even though the candidates at the other frontier molecules could be considered in the same turn. Such a comparison starts at the step level, where each decision determines the synthesizability of the resulting route: the chosen reaction should be plausible under the distribution of known reactions (Lowe, 2017), as measured by the reaction filter models used in multi-step search software (Genheden et al., 2020; Tu et al., 2025; Maziarz et al., 2025). In addition, each decision should consider whether the predicted precursors reproduce the product under a forward model, known as round-trip consistency (Schwaller et al., 2020; Lin et al., 2022), and at the route level, whether the route reaches inexpensive starting materials, as estimated by a price predictor (Hastedt et al., 2025).

To motivate our approach, we examine the routes available within a retrosynthetic search space built for targets from the subset of the USPTO-190 (Chen et al., 2020) benchmark that are solvable within a depth of five, where R-SMILES (Zhong et al., 2022) returns its top-4 reaction candidates for each molecule. We compare these routes from four different views: (1) *Agent Expanded*: routes successfully found by RetroAgent (Zhu et al., 2026), (2) *Exposed, but Unexpanded*: routes whose molecules are opened by the agent during search but are not further expanded, (3) *Unexposed*: routes recovered only by exhaustively expanding that space with the same candidates, and (4) RETROPLANNER: routes recovered by our planner through joint multi-route expansion and multi-objective comparison. As shown in Figure 1, many routes that dominate the agent's routes in reaction plausibility, round-trip consistency, and starting-material price are already within its reach, among those the agent itself opened and left unexpanded. Among the 42 targets solved by the agent in this 60-target subset, the agent's route is dominated by some route in the search space on 78.6% of them, and on 64.3% that dominating route is already among those exposed but left unexpanded. Recovering these routes benefits from two components: opening several molecules within a turn places their candidates in view at once, and comparing candidates across the three objectives helps retain synthesizable routes. RETROPLANNER is designed to realize this behavior by jointly expanding multiple frontier molecules and comparing their candidate routes in one multi-objective decision.

In this paper, our contributions are summarized as follows.

- **Multi-Route Expansion.** We formulate frontier expansion as a joint decision over multiple molecules. RETROPLANNER opens several frontier molecules within the same turn, so that competing routes are evaluated against one another, yielding more diverse routes.
- **Multi-Objective Reasoning.** We define route quality as a reasoning objective over reaction plausibility, round-trip consistency, and starting-material price, so that each expansion decision explicitly states the structural change of the molecule, examines how generally applicable the reaction is at the bond and reaction-class level, and specifies where the candidate stands among the routes already explored and which objective separates it from the alternatives.
- **Diverse and Synthesizable Route Discovery.** RETROPLANNER recovers substantially more alternative routes than prior LLM-based planners while improving their synthesizability across decision signals, approaching dedicated multi-objective search methods while maintaining a high success rate.

2 METHOD

2.1 PROBLEM FORMULATION

We first formulate multi-step retrosynthesis planning as a sequential search over an AND–OR graph that evolves over search turns. Let $\mathcal{M}$ be the set of all molecules and $\mathcal{R}$ the set of all reactions, and let $t = 0, 1, \ldots$ index the search turn, with $t = 0$ the initial state containing only the target molecule m_0. At turn t, the current search graph is denoted by:

$$\mathcal{G}_t = (\mathcal{V}_t^M, \mathcal{V}_t^R, \mathcal{E}_t) \tag{1}$$

where $\mathcal{V}_t^M \subset \mathcal{M}$ and $\mathcal{V}_t^R \subset \mathcal{R}$ denote the sets of molecule nodes $m \in \mathcal{M}$ (OR nodes) and reaction nodes $r \in \mathcal{R}$ (AND nodes) discovered up to turn t, respectively. $\mathcal{E}_t$ is the set of directed edges

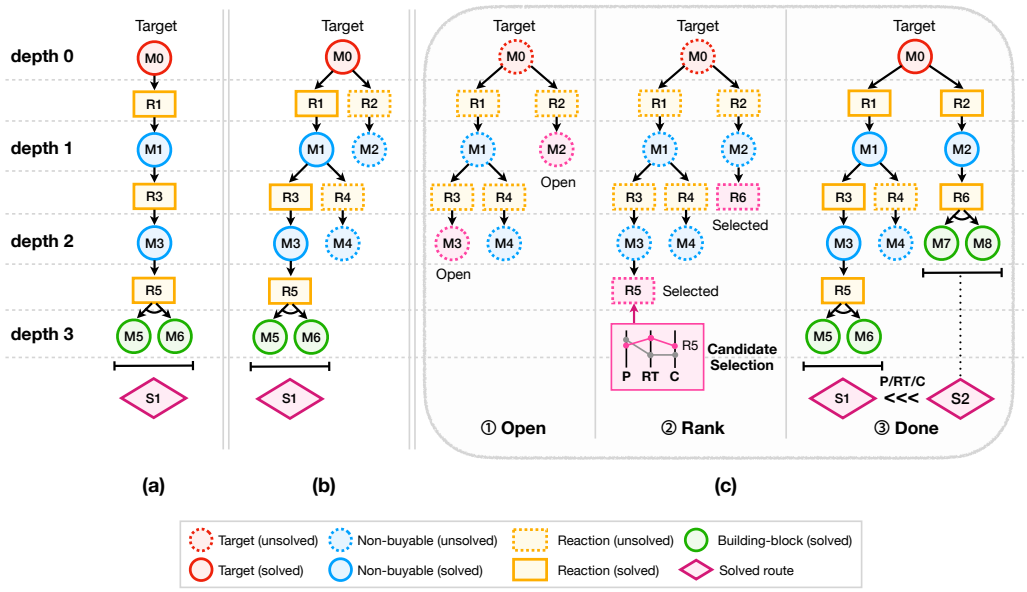

Figure 2: Planning on the same target under three expansion patterns, corresponding to the strategies analyzed in Section 4.1. (a) Single-route expansion extends one route per turn, as in Retro-R1. (b) Sequential multi-route expansion grows several routes, but each turn opens and ranks one frontier molecule, as in RetroAgent. (c) Parallel multi-route expansion, used by RETROPLANNER, runs three stages per turn: `open` selects multiple molecules from the frontier set $\mathcal{O}_t$ to expand at once, `rank` jointly orders the reaction candidates of the opened molecules for expansion through multi-objective reasoning over reaction plausibility (P), round-trip consistency (RT), and starting-material price (C). `done` commits routes whose leaves are all in the building-block set $\mathcal{BB}$.

between molecules and reactions at turn t. We keep $\mathcal{G}_t$ a directed acyclic graph (DAG), where a molecule appears as a single node shared by every route that contains it. RETROPLANNER therefore observes the explored routes on the same graph, as shown in Figure 2(c), so a candidate can be compared against the routes already explored.

Let $\mathcal{BB} \subset \mathcal{M}$ denote the set of purchasable building-block molecules, and $\mathrm{ch}(\cdot)$ denote the children of a node. Standard AND–OR semantics then give

$$\texttt{SOLVED}(m) \iff m \in \mathcal{BB} \ \vee \ \exists r \in \mathrm{ch}(m) : \ \texttt{SOLVED}(r), \tag{2}$$

$$\texttt{SOLVED}(r) \iff \forall m' \in \mathrm{ch}(r) : \ \texttt{SOLVED}(m') \tag{3}$$

so a molecule is solved once any one of its reactions is, whereas a reaction needs all of its precursors. We define the set of frontier molecules at turn t as

$$\mathcal{O}_t = \{\, m \in \mathcal{V}_t^M : \neg\texttt{SOLVED}(m) \,\} \tag{4}$$

$\mathcal{O}_t$ is the set of molecules that are not yet solved and therefore remain candidates for expansion. Since opened molecules retain their candidates, the agent can revisit them at later turns and expand several at once, yielding alternative routes through the same subgraph.

2.2 OVERVIEW OF RETROPLANNER

LLM planner with environment. At turn t the planner observes the search graph $\mathcal{G}_t$ and acts according to LLM policy $\pi_\theta(\cdot \mid \mathbf{h}_t)$ with history $\mathbf{h}_t = (\mathcal{G}_0, \tilde{a}_0, \cdots, \mathcal{G}_{t-1}, \tilde{a}_{t-1}, \mathcal{G}_t)$. An action $\tilde{a}_t$ consists of structured decisions, where z denotes a structured reasoning trace preceding the corresponding decision,

$$\tilde{a}_t = (z_t^{\texttt{open}}, a_t^{\texttt{open}}, z_t^{\texttt{rank}}, a_t^{\texttt{rank}}, z_t^{\texttt{done}}, a_t^{\texttt{done}}) \tag{5}$$

The three decisions correspond to the `open`, `rank`, and `done` stages of a turn. The episode terminates when the planner takes the terminal action $a^{\texttt{final}}$. They are generated autoregressively so that every decision is conditioned on history produced earlier in the turn, including the trace that precedes it.

Multi-Route Expansion. Existing agentic planners typically operate on a single open molecule per turn, even when multiple reaction candidates are considered for it (Zhu et al., 2026). As illustrated in Figure 2(a) and (b), attaching several candidates to a single frontier molecule widens the graph only locally. We instead allow *set-valued* decisions over multiple molecules $m \in \mathcal{O}_t$ to simultaneously explore diverse routes and balance multi-objective trade-offs within a single turn, as in Figure 2(c).

In the `open` stage, the agent jointly selects a subset of molecules to expand,

$$a_t^{\texttt{open}} \subseteq \mathcal{O}_t, \tag{6}$$

and invokes `open`; for each selected m, the environment provides $f_K(m)$, the top-K reaction candidates proposed by the single-step model $f(\cdot)$.

In the `rank` stage, the action $a_t^{\texttt{rank}}$ orders reaction candidates of each opened molecule under the multi-objective criterion and determines how many of the top-ranked reactions to expand for each opened molecule.

$$a_t^{\texttt{rank}} \in \prod_{m \in a_t^{\texttt{open}}} \mathcal{S}_{\leq 3}\big(f_K(m)\big), \qquad K = 10, \tag{7}$$

where $\mathcal{S}_{\leq 3}(A)$ denotes the set of ordered tuples containing one to three distinct elements of A. For each opened molecule, the agent re-ranks its candidates and specifies how many of them, counted from the top of that ranking, are expanded in this turn. Each reaction candidate $r \in f_K(m)$ is displayed with its reaction plausibility $p(r)$, forward round-trip consistency $rt(r)$, and starting-material price $c_{\text{prec}}(r)$, aggregated over the predicted prices of its precursor molecules, so that candidates can be compared within each molecule and across the opened molecules. The environment attaches every selected reaction with its precursors to the graph, so several routes grow within a single turn. In the `done` stage, $a_t^{\texttt{done}}$ commits the routes the agent judges to be complete, and the search continues on the routes that remain.

Multi-Objective Reasoning. Expanding multiple routes exposes alternatives with different quality trade-offs. We consider synthesizability through three objectives: reaction plausibility, round-trip consistency, and starting-material price. We therefore train the agent to reason over these objectives rather than optimizing only for success rate. Rather than imposing a fixed weighting across the search, different objectives can guide individual expansion decisions, collectively shaping the trade-off among the three objectives at the route level. For a complete route ρ, route quality is the multi-objective vector

$$\mathbf{u}(\rho) = \big[\, u_{\text{p}}(\rho),\, u_{\text{rt}}(\rho),\, u_{\text{c}}(\rho) \,\big], \tag{8}$$

aggregating the three objectives over reactions and terminal molecules in ρ. This reasoning takes place in the rank trace $z_t^{\texttt{rank}}$, where the agent compares candidates along the three objectives before committing to the ordered selection $a_t^{\texttt{rank}}$. The agent also considers auxiliary signals, such as single-step model confidence and structural information from the current graph, which inform the comparison of candidates and partial routes.

2.3 Learning How to Plan

Evidence-Grounded Reasoning. We require the agent to specify the evidence behind each choice in reasoning fields. First, the `MOLECULE` field compares the molecule at the current depth with the target, characterizing how far its structure has been simplified through changes in scaffold, size, rings, stereochemistry, and precursor-fragment decomposition. Second, the `DISCONNECTIONS` field identifies the bond changes and associates the transformation with a reaction class when available, allowing each candidate to be recognized as a known transformation. Third, the `DECIDING` field references numerical signals provided on the search graph, such as reaction plausibility, round-trip consistency, and starting-material price, so that candidates and the routes they extend can be compared against all explored partial routes, with every selected candidate carrying two subfields, `diverging from` and `ordered by`. Detailed examples for all three fields are provided in Section A.6.

In the `diverging from` subfield, the agent refers to the explored route or the competing candidate in the same turn that is most relevant to the candidate, based on the acquired chemical evidence and the search history. It also notes when the molecule has not been expanded by any explored route. In the `ordered by` subfield, the agent identifies the objective most decisive for ordering the candidate among its competing alternatives in the current decision. Different objectives can therefore

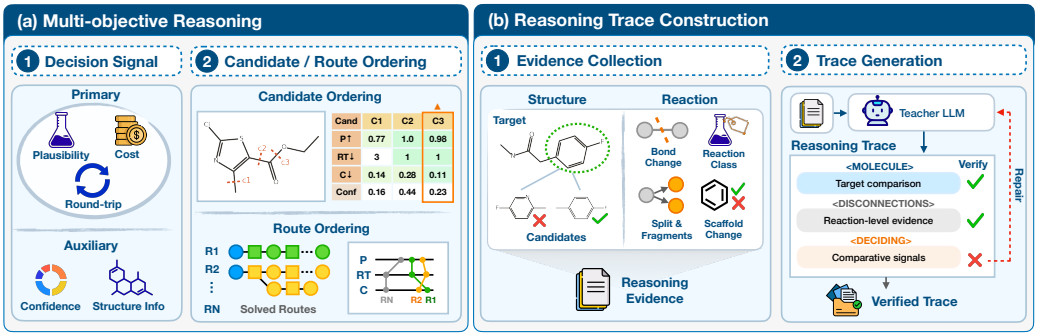

Figure 3: **(a)** Multi-objective reasoning. Reaction candidates are compared on three primary decision signals and auxiliary signals, which are referenced to order the candidates of each molecule. **(b)** Reasoning trace construction. Structural and reaction-level evidence is collected for the target and its candidates; a teacher LLM writes the `MOLECULE` / `DISCONNECTIONS` / `DECIDING` trace from that evidence, and each field is verified against the search graph, with detected inconsistencies returned as repair feedback until the trace passes.

be decisive at different steps along the same route. The primary objectives are reaction plausibility p, forward round-trip consistency rt, and starting-material price c, the last on a normalized log scale. Single-step confidence q serves as an auxiliary signal when the leading candidates are nearly tied on the primary objectives.

Training with Verified Reasoning. During training corpus construction, the structured reasoning schema enables direct verification against the search graph, as illustrated in Figure 3(b). Every numerical value cited in the reasoning trace must match what the graph records. For example, a reasoning trace reporting `value` : 0.82 for a selected candidate whose recorded plausibility is 0.67 fails verification. For each selected candidate, the route cited in `diverging from` must exist in the graph and differ from the selected candidate as stated, and the objective cited in `ordered by` must separate the candidate from its competitors in the stated objective. A failed trace is returned to the teacher LLM with the detected inconsistencies as repair feedback, or regenerated when the defects cannot be localized. When no trace passes verification, the action has no reasoning trace, so only verified reasoning is used for supervised fine-tuning. Given a search-state input x and its verified reasoning-and-action sequence y, we optimize the standard autoregressive objective: $\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{t=1}^{|y|} \log p_\theta(y_t \mid x, y_{<t})$.

3 Experiments

3.1 Experimental Setup

Baselines. We compare our method against two groups of baselines. First, we include conventional multi-step search algorithms, MCTS (Segler et al., 2018), Retro*-0, Retro* (Chen et al., 2020), and MORetro* (Hastedt et al., 2026), evaluated under the same single-step retrosynthesis model and search budget N_B. All main results use $N_B = 50$, while the ablation study raises this to $N_B = 300$ so that each variant is compared under a budget large enough not to be the limiting factor. For a controlled comparison, we replace the confidence score that drives node selection in these algorithms with an equal-weight scalarization of reaction plausibility, round-trip consistency, and starting-material price. For MORetro*, we retain its original Bayesian-optimization (BO) sampling procedure over objective weights. Second, we include trained LLM-based planners such as Retro-R1 (Liu et al., 2026) and RetroAgent (Zhu et al., 2026). RetroPlanner is initialized from GPT-OSS-20B (OpenAI, 2025) and supervised fine-tuned on our training corpus to learn multi-route expansion and multi-objective reasoning.

Training Setup. We construct the training set from the PaRoutes (Genheden & Bjerrum, 2022) training split. Each target is searched once with Retro*-0 (Chen et al., 2020), using R-SMILES (Zhong et al., 2022) as a template-free single-step model to sample diverse retrosynthetic candidates while reducing dependence on template coverage. We Pareto-rank the searched routes

over the three synthesis objectives and merge a diverse subset of high-ranking routes at their shared intermediates into a single directed acyclic graph, with the selection procedure detailed in Section A.3. We then replay this graph breadth-first in a deterministic simulation of the planning environment, which converts each step into one of the planner's actions: `open`, `rank` with the number of committed candidates, or `done`, and ends the episode with `final`. We use Qwen3.8-27B (Qwen Team, 2026) as a teacher model to generate reasoning traces for these actions, verified as described in Section 2.3.

Evaluation Setup. We evaluate on the in-distribution USPTO-190 (Chen et al., 2020) and the out-of-distribution ChEMBL-1000 (Liu et al., 2023). Search baselines are run once per budget cell under the common expansion budget. LLM-based planners instead run multiple rollouts that consume a cumulative expansion budget until exhausted. Following reported settings (Liu et al., 2026; Zhu et al., 2026), Retro-R1 searches within a single continuous rollout over multiple rounds, whereas RetroAgent restarts failed rollouts until a route is found. RETROPLANNER ends each rollout at its first complete route and restarts, spending the remaining budget on alternative routes. Unlike the baselines, it does not shuffle the environment and instead samples its reasoning and decisions at temperature 0.6, so that variation across rollouts comes from the agent itself. Since these settings differ across baselines, we further train each expansion strategy on the same backbone and search mechanism, as detailed in Section 4.1.

Metrics. We group metrics into **Search Outcome** and **Route Quality**. *Success Rate (Succ.)* is the percentage of targets for which a route reaching purchasable building blocks from eMolecules[1] is found within the expansion budget, and *Routes* is the number of routes returned per solved target. For route quality, we report the composite-best route recovered by each method for each target, selected jointly over reaction *Plausibility (Plaus.)*, *Round-Trip (RT)* consistency, and starting-material *Price*. *Better-Than-Ref (BTR)* is the fraction of targets for which the selected route attains a higher composite score than the best published reference route over the same three objectives, with unsolved targets counted as failures. Additionally, *1stCuts* counts the distinct first-cut precursor sets recovered per target and *1stCutDiff* is the fraction of route pairs with different first-cut precursor sets, both measuring how broadly a planner explores alternative branches at the initial retrosynthetic decision. Further evaluation details are provided in Section A.4.

3.2 MAIN RESULTS

Q1. Does the planner efficiently recover diverse, high-quality routes under a constrained search budget?

We evaluate all methods under an expansion budget of $N_B = 50$, focusing on how effectively each planner allocates a limited search budget toward solvable and high-quality routes under the three evaluation objectives.

Search Outcome. As shown in Table 1, LLM-based planners generally achieve higher success rates than conventional search algorithms under the limited search budget, indicating that learned planning policies allocate search effectively toward solvable routes. On USPTO-190, MORetro* optimizes weighted combinations of the three route-quality objectives, whereas its expansions are governed by sampled objective weights rather than route-level context-aware decision-making, with a success rate 24.22 percentage points lower than that of RETROPLANNER. Among the LLM-based planners, RetroAgent attains the highest success rate on both USPTO-190 and ChEMBL-1000, as its policy is primarily optimized toward reaching a solved route. However, it does not explicitly account for synthetic route quality during expansion, whereas RETROPLANNER incorporates these considerations into its reasoning, resulting in success rates 4.73 and 5.90 percentage points lower, respectively, but substantially more alternative routes with more favorable trade-offs across route-quality objectives.

Route Quality. On USPTO-190, RETROPLANNER achieves the highest *Better-Than-Ref*, more often recovering solved routes that exceed the reference under the three evaluation objectives, while maintaining strong performance across all three route-quality metrics. Compared with MORetro*, which explicitly searches over the same three objectives, RETROPLANNER achieves a higher success

[1] `https://downloads.emolecules.com/`

Table 1: Main results on multi-step retrosynthesis planning on USPTO-190 and ChEMBL-1000 with search budget $N_B = 50$. *Search Outcome* measures whether and how many routes are found, *Route Quality* evaluates the found routes on reaction plausibility, round-trip consistency, and starting-material price. *Better-Than-Ref* is N/A on ChEMBL-1000, which provides no reference routes. Best and second-best results per dataset are shown in **bold** and underlined.

Method	Search Outcome		Route Quality			
	Success Rate ↑	Routes ↑	Better-Than-Ref ↑	Plausibility ↑	Round-Trip ↑	Price ↓
USPTO-190						
Search Algorithms						
MCTS	44.74%	16.88	27.89%	43.33%	25.88%	391.19
Retro*-0	52.63%	29.55	34.21%	49.16%	38.99%	328.02
Retro*	57.89%	33.34	38.95%	54.93%	41.64%	281.53
MORetro*	67.89%	**43.16**	52.11%	66.16%	58.94%	250.63
LLM-based Search						
Retro-R1	82.11%	2.99	38.42%	71.64%	39.34%	197.29
RetroAgent	**96.84%**	3.07	46.84%	85.85%	38.16%	**138.83**
RetroPlanner	92.11%	7.41	**66.84%**	**86.03%**	**61.11%**	167.80
ChEMBL-1000						
Search Algorithms						
MCTS	59.30%	28.31	N/A	56.05%	49.47%	331.59
Retro*-0	72.70%	88.83	N/A	69.22%	64.40%	249.71
Retro*	74.90%	**89.21**	N/A	71.45%	66.11%	239.78
MORetro*	77.80%	26.60	N/A	75.65%	72.37%	221.22
LLM-based Search						
Retro-R1	82.90%	2.74	N/A	74.74%	53.97%	253.53
RetroAgent	**95.90%**	5.28	N/A	**87.86%**	58.35%	207.43
RetroPlanner	90.00%	10.59	N/A	87.33%	**74.22%**	**190.89**

rate while improving reaction plausibility, round-trip consistency, and starting-material price on both benchmarks. On the other hand, RetroAgent achieves comparable reaction plausibility and starting-material price, but substantially lower round-trip consistency on both benchmarks, whereas RETROPLANNER maintains strong performance across all three synthesizability objectives. RETROPLANNER achieves a more balanced performance across the three objectives and more frequently recovers routes that score above the reference route under the composite metric on USPTO-190, while showing consistent performance across these objectives on ChEMBL-1000, where reference routes are unavailable.

4 ABLATION AND ANALYSIS

We analyze which components of RETROPLANNER contribute to its search behavior and route quality. In this section, a rollout continues searching after its first complete route, and rollouts are repeated per target until every variant reaches the same number of single-step calls. We separate the expansion strategy into three axes: whether frontier nodes are selected at random or greedily by score alone, whether a single or multiple frontier nodes are expanded per turn, and whether the expansions are performed sequentially or in parallel. For the reasoning schema, we examine which forms of chemical evidence grounding are useful for planning and how much reasoning over the three synthesis objectives contributes to route diversity and quality. Since every candidate the planner selects from is proposed by the single-step model, we also replace this model to verify that the planner's gains hold across its variants.

4.1 EXPANSION STRATEGY

Q2. How does the expansion strategy shape route-set coverage and search efficiency under extended search?

Expansion Strategy Variants. For a controlled comparison of expansion strategies on USPTO-190, all methods use the same extended search budget $N_B = 300$. Figure 4(a) compares *Multi-Route*

(a)

Method	Search Outcome		Diversity		Quality
	Succ. ↑	Routes ↑	1stCutDiff ↑	1stCuts ↑	BTR ↑
Random	0.879	1.55	1.000†	1.55	0.280
Score-only	0.937	1.59	1.000†	1.59	0.672
Single-Route	0.984	6.05	0.450	2.23	0.780
Multi-Route (Seq.)	0.995	**9.10**	**0.660**	**3.88**	0.812
Multi-Route (Par.)	**1.000**	7.75	0.654	3.54	**0.823**

(b)

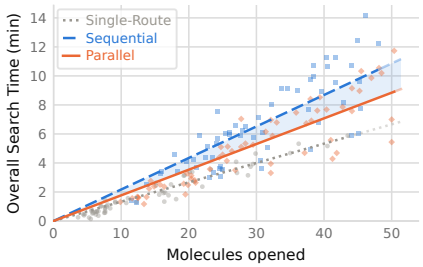

Figure 4: (a) Effect of the expansion strategy on route diversity and quality. All variants share the same backbone, GPT-OSS-20B, and differ only in the expansion strategy. †Computed only on targets with at least two recovered routes; since Random and Score-only recover few routes, 1.00 reflects complete first-cut disagreement among the available pairs rather than route coverage. (b) Overall search time versus molecules opened per rollout. Although the two multi-route arms open a comparable number of molecules per episode, *Sequential* spends more time per molecule than *Parallel*.

(Parallel), which selects multiple frontier molecules and expands them together within a single turn, against variants that each change one component. *Random* and *Score-only* replace the learned policy with fixed rules: *Random* selects frontier molecules and reaction candidates uniformly at random, and *Score-only* ranks candidates by an equal-weight scalarization of normalized reaction plausibility, round-trip consistency, and starting-material price. The other two are separately trained policies that differ from *Multi-Route (Parallel)* only in how frontier molecules are expanded: *Single-Route* expands a single frontier molecule per turn, and *Multi-Route (Sequential)* selects multiple frontier molecules but expands them one per turn instead of together.

Effect of Expansion Strategy. *Random* and *Score-only* return fewer than two routes per target. Scalarizing the three objectives raises *BTR* from 0.280 to 0.672, but scoring each candidate independently does not accumulate alternative routes across the search. Among the learned policies, selecting multiple frontier molecules broadens the search, as *Multi-Route (Sequential)* raises the number of distinct first cuts from 2.23 to 3.88 over *Single-Route*. *Multi-Route (Parallel)* recovers fewer routes than *Multi-Route (Sequential)* but keeps a comparable *1stCutDiff*, indicating fewer redundant routes under each first cut, and achieves the highest success rate and *Better-Than-Ref*. Comparing candidates across frontier molecules within a single turn therefore allocates the search budget toward alternatives that differ from those the graph already holds, rather than toward repeats of the same first cut.

Parallel expansion is also cheaper. As shown in Figure 4(b), search time grows linearly with the number of molecules opened. The two multi-route variants open a comparable number of molecules per rollout, 30.4 and 29.6, yet *Multi-Route (Sequential)* spends 13.0 s per molecule against 10.6 s for *Multi-Route (Parallel)*, about 23% more, or 6.5 against 5.1 minutes per rollout. The gap comes from how much is generated per expansion: per molecule, parallel expansion takes 1.71 turns against 2.30 and 80.2 output tokens against 106.9, so the reasoning over the search graph is shared across the opened molecules rather than repeated for each expansion. This brings its cost per molecule close to *Single-Route* at 78.5 tokens, while opening twice as many molecules per rollout, 29.6 against 14.8. Overall, *Multi-Route (Parallel)* obtains its quality gains while searching faster than sequential multi-route search.

4.2 Reasoning Schema

Q3. Which components of the reasoning schema contribute to route diversity and quality?

We construct a controlled hierarchy from the same training trajectories and target actions. All variants share the same action format and differ only in the reasoning trace that precedes each action.

Effect of Reasoning Schema. As shown in Table 2, compared with the model trained without explicit reasoning (L_0), the full schema (L_3) improves all diversity metrics and slightly improves

Table 2: Ablation on the reasoning schema on USPTO-190, with all variants run under the same search budget of at most $N_B = 300$. Schema blocks: **M** = `MOLECULE`, **D** = `DISCONNECTIONS`, and **C** = `DECIDING`. The **Rel.** columns indicate the `diverging from` (**div**) and `ordered by` (**ord**) subfields of `DECIDING`. **Par.** denotes the parent variant from which each delta is computed. Best results are in **bold**.

Variant	Schema			Rel.		Par.	Search Outcome		Diversity		Quality
	M	D	C	div	ord		Succ. ↑	Routes ↑	1stCutDiff ↑	1stCuts ↑	BTR ↑
L_0	–	–	–	–	–	–	0.989	4.78	0.628	2.79	0.720
L_1	✓	–	–	–	–	L_0	0.989 (+0.000)	5.49 (+0.71)	0.691 (+0.063)	3.15 (+0.36)	**0.769** (+0.049)
L_2	✓	✓	–	–	–	L_1	**0.995** (+0.006)	5.53 (+0.04)	0.722 (+0.031)	3.16 (+0.01)	0.742 (−0.027)
$L_{3\text{-div}}$	✓	✓	✓	✓	–	L_2	0.989 (−0.006)	**5.78** (+0.25)	0.726 (+0.004)	3.34 (+0.18)	0.720 (−0.022)
$L_{3\text{-ord}}$	✓	✓	✓	–	✓	L_2	0.979 (−0.016)	5.08 (−0.45)	**0.756** (+0.034)	3.20 (+0.04)	0.737 (−0.005)
L_3	✓	✓	✓	✓	✓	L_2	0.984 (−0.011)	5.59 (+0.06)	0.745 (+0.023)	**3.35** (+0.19)	0.726 (−0.016)

BTR, while maintaining a comparable success rate. Adding molecule-level structural reasoning (`MOLECULE`, L_1) gives the largest single gain, raising *BTR* from 0.720 to 0.769. Disconnection-level evidence such as bond changes and reaction classes (`DISCONNECTIONS`, L_2) and the retained-candidate record (`DECIDING`, L_3) mainly improve diversity. `diverging from` ($L_{3\text{-div}}$) yields the largest number of recovered routes, whereas `ordered by` ($L_{3\text{-ord}}$) achieves the highest *1stCutDiff*. Enabling both (L_3) maintains high route coverage while yielding the most distinct first cuts and partially recovering the *BTR* lost by `diverging from` alone.

Molecule-level evidence, which states how far a candidate simplifies the target, improves both the breadth and quality of the retained routes, while disconnection-level evidence, which identifies what a candidate changes, mainly reduces routes that share the same first cut. With both subfields, the planner retains diverse alternatives that remain grounded in both the search history and the synthesis objectives. Overall, the reasoning fields shape different aspects of search behavior, and the full schema combines increased route diversity with a high *BTR*.

4.3 Single-Step and Round-Trip Model Variations

To assess generalization across reaction models, we vary both the single-step candidate generator and the forward model used for round-trip evaluation. Replacing the template-free R-SMILES used during planner training with the template-based LocalRetro (Chen & Jung, 2021), RetroPlanner retains the highest *Better-Than-Ref*, showing that the learned planning policy transfers to an unseen single-step model without additional training. Similarly, replacing ReactionT5v2 (Sagawa & Kojima, 2025) with RXNGraphormer (Xu et al., 2025) for round-trip evaluation preserves RetroPlanner's advantage in both *Better-Than-Ref* and *Round-Trip*, including over MORetro*, which directly optimizes the same route-quality objectives during search. Detailed results are provided in Section A.5.

5 Limitations and Conclusion

We introduced RetroPlanner, an LLM-based retrosynthesis planner that formulates frontier expansion as a joint decision over multiple molecules rather than committing to a single molecule at each step. By combining multi-route expansion with chemical evidence-grounded multi-objective reasoning, the planner compares competing routes using reaction plausibility, round-trip consistency, and starting-material price, while specifying how candidate routes differ from previously explored alternatives. Across both in-distribution and out-of-distribution benchmarks, RetroPlanner recovers a broader set of synthesis routes while maintaining a high success rate and improving the quality of the retained alternatives. Ablations further show that both the multi-route expansion strategy and the reasoning schema contribute to this behavior, and that the learned planning policy is robust to the choice of single-step and round-trip consistency models. Our evaluation operationalizes synthesizability through three measurable proxies rather than capturing experimental feasibility in full. Future work could optimize these route-level objectives directly through reinforcement learning, with credit assignment across multi-step and multi-route decisions, and extend them to laboratory factors such as reaction conditions, yield, and safety.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, literature retrieval and discovery, research ideation and implementation support, and the generation of synthetic training data. For writing, we used LLMs to refine phrasing and improve the clarity of our drafts, and we reviewed and edited all resulting text. For retrieval and discovery, we used LLM-based search to locate candidate related work, and we read and verified every cited paper. For research ideation and execution, we used LLMs to discuss design alternatives for the planning agent and to assist in writing experiment and analysis code, which we reviewed and tested before use. For synthetic data generation, LLMs were used to construct the supervised fine-tuning corpus of reasoning trajectories, as described in Section 2.3; every trajectory was verified against the recorded search graph, and only traces passing this verification were used for training. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or the release of new experimental data. All experiments use publicly available reaction datasets and benchmarks, and all models and building-block catalogs are used in accordance with their licenses. Retrosynthesis planning is a domain where a method that proposes practical routes could in principle be applied to harmful target molecules. Our method does not introduce a new reaction generator or expand the underlying candidate space; instead, it changes how routes proposed by existing single-step models are compared and reordered during search. The proposed routes are computational suggestions evaluated using model-based signals and require expert review and experimental validation before any laboratory use.

REPRODUCIBILITY STATEMENT

We provide the search formulation, reasoning schema, and training procedure of RETROPLANNER in Section 2, and the evaluation protocol and ablation settings in Section 3. The appendix describes the training data construction, input prompt and output format, hyperparameters, and evaluation details in Sections A.3, A.4 and A.6. All datasets, the single-step model, the base model, and the scoring tools we use are publicly available. Additional details and code are provided in the supplementary material.

REFERENCES

Liao Chang, Luotian Yuan, Yiping Ke, and Ying Wei. Retrorchestrator: A multi-step retrosynthesis agent dynamically orchestrating single-step transition models. In *Forty-third International Conference on Machine Learning*, 2026.

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

Elias James Corey and W Todd Wipke. Computer-assisted design of complex organic syntheses: Pathways for molecular synthesis can be devised with a computer and equipment for graphical communication. *Science*, 166(3902):178–192, 1969.

Maarten R Dobbelaere, István Lengyel, Christian V Stevens, and Kevin M Van Geem. Rxn-INSIGHT: fast chemical reaction analysis using bond-electron matrices. *J. Cheminform.*, 16(1), March 2024.

Nuwan Ganganath, Chi-Tsun Cheng, and K Tse Chi. Multiobjective path planning on uneven terrains based on namoa. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1846–1849. IEEE, 2016.

Samuel Genheden and Esben Bjerrum. Paroutes: towards a framework for benchmarking retrosynthesis route predictions. *Digital Discovery*, 1(4):527–539, 2022.

Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of cheminformatics*, 12(1):70, 2020.

Peng Han, Peilin Zhao, Chan Lu, Junzhou Huang, Jiaxiang Wu, Shuo Shang, Bin Yao, and Xiangliang Zhang. Gnn-retro: Retrosynthetic planning with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4014–4021, 2022.

Friedrich Hastedt, Klaus Hellgardt, Sophia Yaliraki, Dongda Zhang, and Antonio del Rio Chanorena. Molprice: assessing synthetic accessibility of molecules based on market value. *Journal of Cheminformatics*, 17(1):150, 2025.

Friedrich Hastedt, Dongda Zhang, and Antonio del Rio Chanona. From feasible to practical: Pareto-optimal synthesis planning. *arXiv preprint arXiv:2605.07521*, 2026.

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.

Akihiro Kishimoto, Beat Buesser, Bei Chen, and Adi Botea. Depth-first proof-number search with heuristic edge cost and application to chemical synthesis planning. *Advances in Neural Information Processing Systems*, 32, 2019.

Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

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

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 *International conference on machine learning*, 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. *Advances in Neural Information Processing Systems*, 38:70709–70737, 2026.

Daniel Lowe. Chemical reactions from US patents (1976-Sep2016). 6 2017. doi: 10.6084/m9.figshare.5104873.v1. URL https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.

Krzysztof Maziarz, Austin Tripp, Guoqing Liu, Megan Stanley, Shufang Xie, Piotr Gaiński, Philipp Seidl, and Marwin HS Segler. Re-evaluating retrosynthesis algorithms with syntheseus. *Faraday Discussions*, 256:568–586, 2025.

Ramil Nugmanov et al. Cgrtools: Python library for molecule, reaction, and condensed graph of reaction processing. *Journal of Chemical Information and Modeling*, 59(6):2516–2521, 2019. doi: 10.1021/acs.jcim.9b00102.

OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Qwen Team. Qwen3.8-27b. https://huggingface.co/Qwen/Qwen3.8-27B, 2026.

Tatsuya Sagawa and Ryosuke Kojima. Reactiont5: a pre-trained transformer model for accurate chemical reaction prediction with limited data. *Journal of Cheminformatics*, 17(1):126, 2025.

John S Schreck, Connor W Coley, and Kyle JM Bishop. Learning retrosynthetic planning through simulated experience. *ACS central science*, 5(6):970, 2019.

Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science*, 11(12):3316–3325, 2020.

Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances*, 7(15):eabe4166, 2021.

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

Zhengkai Tu, Sourabh J Choure, Mun Hong Fong, Jihye Roh, Itai Levin, Kevin Yu, Joonyoung F Joung, Nathan Morgan, Shih-Cheng Li, Xiaoqi Sun, et al. Askcos: open-source, data-driven synthesis planning. *Accounts of chemical research*, 58(11):1764, 2025.

Weijia Wang and Michele Sebag. Multi-objective monte-carlo tree search. In *Asian conference on machine learning*, pp. 507–522. PMLR, 2012.

Shufang Xie, Rui Yan, Peng Han, Yingce Xia, Lijun Wu, Chenjuan Guo, Bin Yang, and Tao Qin. Retrograph: Retrosynthetic planning with graph search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2120–2129, 2022.

Li-Cheng Xu, Miao-Jiong Tang, Junyi An, Fenglei Cao, and Yuan Qi. A unified pre-trained deep learning framework for cross-task reaction performance prediction and synthesis planning. *Nature Machine Intelligence*, 7(9):1561–1571, 2025.

Bogdan Zagribelnyy, Ivan Ilin, Nikita Bondarev, Anton Morgunov, Arkadii Lin, Maksim Kuznetsov, Rim Shayakhmetov, Vladimir Aladinskiy, Alex Aliper, and Alex Zhavoronkov. Ursa: Chemistry-aware benchmark for utilitarian retrosynthesis assessment. *arXiv preprint arXiv:2607.04688*, 2026.

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.

Yanqiao Zhu, Jingru Gan, Xiaoqi Sun, Fang Sun, Yidan Shi, Md Mofijul Islam, Chao Shang, Wenhao Gao, Connor W Coley, Yizhou Sun, et al. Retroagent: Harnessing llms to search over structured memory for agentic retrosynthesis planning. *arXiv preprint arXiv:2607.14512*, 2026.

A Appendix

A.1 Related Work

Computer-aided Synthesis Planning. Modern computer-aided synthesis planning typically combines a single-step retrosynthesis model with a multi-step search algorithm (Hong et al., 2023; Chen et al., 2020; Liu et al., 2023; Xie et al., 2022). Single-step models generate candidate precursor sets for a given product molecule and are commonly trained on USPTO-derived datasets such as USPTO-50K and USPTO-FULL (Lowe, 2017). Existing approaches can be broadly categorized into template-based methods, which are interpretable but bounded by the template library, template-free methods, which generalize to unseen transformations at the cost of validity and interpretability, and graph-edit-based methods, which balance the two but depend on accurate atom mapping. Given candidate reactions from a single-step model, multi-step planners determine which molecules and reactions should be expanded. Neural-guided planners commonly adopt Monte Carlo Tree Search (Hong et al., 2023), while methods such as Retro* (Chen et al., 2020) formulate retrosynthesis as an A*-like best-first search using learned or estimated node costs. Other methods, including RetroGraph (Xie et al., 2022) and PDVN (Liu et al., 2023), further improve search efficiency through alternative graph representations or by jointly learning the single-step policy and value estimates. Despite these differences, the central objective of conventional planners remains finding at least one route to purchasable building blocks.

Multi-Objective Retrosynthetic Search. Retrosynthetic planning can involve multiple criteria, including reaction plausibility, route length, precursor cost, and other measures of synthetic feasibility, which a single solvable route does not necessarily satisfy. Multi-objective search methods address this issue by optimizing several route-level objectives and recovering multiple non-dominated routes rather than a single minimum-cost route (Wang & Sebag, 2012; Ganganath et al., 2016; Hastedt et al., 2026). Approaches such as MO-MCTS (Wang & Sebag, 2012) extend conventional retrosynthetic planning toward Pareto-front exploration, building on multi-objective extensions of classical search algorithms such as NAMOA* (Ganganath et al., 2016), while MORetro* (Hastedt et al., 2026) combines weighted-sum search with adaptive sampling over the weight vector. However, the objective space is specified in advance, and diversity is obtained by covering it through weight sampling or Pareto-search procedures rather than by reasoning about which parts of the search space remain unexplored. Moreover, these objectives are typically assumed to decompose additively over reactions, which constrains the notion of route quality that can be optimized during search. In contrast, we let the planner reason over the current search state to decide which trade-offs to pursue, rather than enumerating them through repeated searches under sampled weights.

LLM Agents for Retrosynthetic Planning. Recent work has explored large language models as planners for retrosynthesis. Earlier systems use LLMs to directly generate synthesis pathways or to guide the construction of retrosynthetic search structures. More recent approaches, such as Retro-R1 (Liu et al., 2026), RetrOrchestrator (Chang et al., 2026), and RetroAgent (Zhu et al., 2026), formulate retrosynthesis as an agentic workflow in which the LLM interacts with single-step models or chemistry tools and iteratively selects the next molecule or reaction to explore. These works demonstrate that LLMs can orchestrate tool-based retrosynthetic search through sequential decision-making. However, expansion in these systems is confined to a single frontier molecule per turn: the agent inspects the search state, selects one molecule, and applies one or more candidate reactions to it before repeating the process. Although the search state itself exposes multiple explored and unexplored branches, existing methods do not explicitly learn to use this global context to coordinate complementary exploration across multiple frontier states. Our work focuses on this gap by treating the search frontier as a joint decision context and learning to expand multiple routes while simultaneously considering their chemical feasibility.

A.2 Single-step Models and Chemistry Tools

R-SMILES. R-SMILES (Zhong et al., 2022) is a template-free single-step model that translates a product SMILES into reactant-set SMILES using a transformer architecture. Given a target molecule, R-SMILES serializes the product by a depth-first traversal from a chosen root atom, tokenizes the string, and emits a reactant-set string. Since the reactant is serialized from the same root atom, the input and output sequences stay closely aligned, and the decoder edits the string locally in-

stead of regenerating the molecule. Beam search yields K candidate reactant sets per product, each with its sequence likelihood. We use R-SMILES as a single-step model available to the planner.

LocalRetro. LocalRetro (Chen & Jung, 2021) is a template-based single-step model built on the observation that the atom and bond changes of a reaction are mostly local. LocalRetro extracts local templates that record only the edits at the reaction center, then scores every template at every atom and bond of the product graph. Applying the highest-scoring templates at predicted centers yields precursor sets ranked by scores. We take the top K as the candidate set for the planner.

Reaction Plausibility Filter. AiZynthfinder (Genheden et al., 2020) is open-source software for retrosynthetic planning that couples a Monte Carlo tree search with a template-based expansion policy. The expansion policy takes a product fingerprint and outputs a probability over a library of reaction templates. Each template is a SMARTS rule extracted by atom mapping from reactions reported in the USPTO corpus (Lowe, 2017), on which the policy is also trained. We repurpose the expansion policy of AiZynthfinder as a reaction plausibility filter.

ReactionT5v2. ReactionT5v2 (Sagawa & Kojima, 2025) is a forward reaction prediction model that treats reaction prediction as a sequence-to-sequence translation task. Given a set of reactants as a SMILES string, the model predicts the SMILES of the resulting product. ReactionT5v2 is trained on the Open Reaction Database using a T5 encoder-decoder. We use ReactionT5v2 to compute round-trip consistency (Schwaller et al., 2020).

RXNGraphormer. RXNGraphormer (Xu et al., 2025) couples a graph neural network with a Transformer and is pre-trained on 13 million reactions, then fine-tuned for forward- and retro-synthesis planning as a sequence generation task. We use its forward-synthesis checkpoint as an alternative forward model for round-trip consistency, since its graph-based architecture differs from the sequence-to-sequence models typically used for this axis.

MolPrice. MolPrice (Hastedt et al., 2025) is a molecular price prediction model used as a proxy for synthetic accessibility. Given a molecular structure, it predicts the price in USD per mmol, trained on 5.5M purchasable compounds extracted from the Molport catalogue. Since the catalogue contains only easily purchasable molecules, a contrastive objective self-generates price labels for hard-to-synthesize structures so that the model assigns them higher prices. We use the predicted price to quantify the price of the building blocks a route terminates in.

A.3 Implementation Details

Training Data Construction. We draw 3,208 targets from the PaRoutes (Genheden & Bjerrum, 2022) training split, with gold-route depths ranging from 2 to 14 steps (median 5, mean 4.89). Terminal molecules are matched against the PaRoutes training stock by InChIKey skeleton. To avoid overlap with evaluation, we filter molecules appearing in the USPTO-190 (Chen et al., 2020) reference routes and intermediates appearing in the R-SMILES (Zhong et al., 2022) training data. A final target-level check finds two targets overlapping with USPTO-190 and none with ChEMBL-1000 (Liu et al., 2023), and removing them leaves 3,206 targets. Each target is searched once with Retro*-0 under a budget of 300 single-step calls. We Pareto-rank the searched routes over the three synthesis objectives and retain the first two fronts. The first front alone often concentrates on routes with similar inexpensive starting-material sets, while including the second front broadens route diversity without substantially relaxing route quality. From these, up to eight routes with distinct starting-material sets are retained per target. For Pareto ranking, round-trip consistency uses top-1 recovery, a stricter criterion than the top-5 recovery used in evaluation.

Chemical Evidence. Bond changes are obtained from RXNMapper (Schwaller et al., 2021) atom mappings and represented with CGRtools (Nugmanov et al., 2019) as formed, broken, or changed bonds at the reaction center, while RDKit (Landrum, 2016) is used to describe the affected atoms and functional groups. Reaction classes and names are assigned with Rxn-INSIGHT (Dobbelaere et al., 2024), while template matching separately determines whether the corresponding transformation is applicable to the product and whether it reproduces the proposed precursors. Molecular descriptors summarize scaffold, ring, size, and stereochemical structure for each molecule and their changes from product to precursors. When mapping fails, the result is reported as not measured, never as no bond changed, since conflating the two teaches the model to deny perfectly ordinary transformations.

Table 3: Training and decoding hyperparameters for RETROPLANNER.

Hyperparameter	Value
Training	
Base model	GPT-OSS-20B
Train / val trajectories	3,142 / 64
Epochs	2
Steps	429
Batch size	4 (packed, ∼393K tokens)
Max sequence length	98,304
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Learning rate	1e-5
LR scheduler	Cosine, min LR 1e-6
Warm-up steps	20
Weight decay	0.1
Gradient clipping	1.0
Decoding	
Temperature	0 (single-pass) / 0.6 (iterative)
Top-p	1.0
Max new tokens per turn	3,000
Context length	131,072
Hardware	
Training	4×B200
Inference	8×H100

Hyperparameters. Training is performed on 4 NVIDIA B200 GPUs with FSDP and bf16 precision, while all inference experiments are conducted on 8 NVIDIA H100 GPUs. Full training hyperparameters are reported in Table 3.

Training Objective. We use token-level cross-entropy on packed multi-turn trajectories, supervising the assistant reasoning and action outputs while masking all input and tool-observation tokens. The loss is normalized over supervised tokens across data-parallel ranks.

A.4 EVALUATION DETAILS

Environments. When reaction candidates need to be compared, the planner observes a textual rendering of the input search graph. Each candidate is annotated with four numerical signals, followed by its precursor SMILES. q is the confidence of the single-step model. p is reaction plausibility, given by the AiZynthFinder (Genheden et al., 2020) filter trained on USPTO, and a reaction is counted as plausible when $p \geq 0.05$. rt is round-trip consistency, reported as the rank at which the ReactionT5v2 (Sagawa & Kojima, 2025) forward model recovers the opened molecule from the proposed precursors. $c_{\text{prec}}(r)$ is the aggregate precursor price displayed for reaction candidate r, denoted by \$, computed from the MolPrice (Hastedt et al., 2025) predictions $c(m)$ of its precursor molecules by log-sum-exp and normalized to $[0, 1]$, where lower is cheaper. Each precursor's individual price $c(m)$ is shown next to its SMILES, and $\star$ indicates that the precursor is in stock.

Evaluation Metrics. All metrics are computed over the full benchmark $\mathcal{T}$. Let $\mathcal{R}_T$ be the set of routes returned for target $T \in \mathcal{T}$ within the budget N_B whose leaves all lie in the eMolecules stock. We denote solved targets set $\mathcal{S} = \{T : \mathcal{R}_T \neq \emptyset\}$, and evaluate *Success Rate* following:

$$\textit{Success Rate} = |\mathcal{S}|/|\mathcal{T}|, \tag{9}$$

We denote $r \in \rho$ for the reactions of a route ρ, and $\text{leaf}(\rho)$ for terminal molecules of route ρ. We use the predictors in Section A.2: the single-step reaction plausibility filter $p(r)$, the round-trip $rt(r)$ at which the forward model recovers the product of r, and molecule price predictor $c(m)$ that predicts molecule price. For a reaction candidate r, we denote by $c_{\text{prec}}(r)$ the aggregate predicted price of

its precursor molecules. We first define the route-level quantities as

$$u_p(\rho) = \frac{1}{|\rho|}\sum_{r\in\rho}\mathbf{1}[p(r)\geq 0.05], \quad u_{rt}(\rho) = \frac{1}{|\rho|}\sum_{r\in\rho}\mathbf{1}[rt(r)\leq 5], \quad \tilde{u}_c(\rho) = \sum_{m\in\text{leaf}(\rho)} c(m), \tag{10}$$

Targets for which no route is found are not excluded but scored as 0 on plausibility and round-trip, and as a price cap $P_{\max} = \$1000$.

Route quality is evaluated on a single representative route $\rho_T^\star$ per solved target. While u_p and u_{rt} are already bounded in $[0,1]$, starting-material prices are heavy-tailed and span a much wider numerical range, so we cap the route price at $P_{\max}$ and normalize it on a logarithmic scale to place the cost objective on a comparable scale:

$$\tilde{u}_c^{\text{cap}}(\rho) = \min(\tilde{u}_c(\rho), P_{\max}), \qquad \bar{u}_c(\rho) = \frac{\log(1+\tilde{u}_c^{\text{cap}}(\rho))}{\log(1+P_{\max})}. \tag{11}$$

Here, $\bar{u}_c(\rho)\in[0,1]$, where lower values correspond to cheaper routes. We reverse its direction as $u_c(\rho) = 1-\bar{u}_c(\rho)$ so that all three objectives follow the same higher-is-better convention. The representative route is then selected by equal-weight averaging:

$$s(\rho) = \frac{u_p(\rho)+u_{rt}(\rho)+u_c(\rho)}{3}, \qquad \rho_T^\star = \arg\max_{\rho\in\mathcal{R}_T} s(\rho). \tag{12}$$

Plausibility and *Round-Trip* average the representative route's rates over the benchmark, while price is aggregated geometrically under the same cap:

$$\textit{Plausibility} = \frac{1}{|\mathcal{T}|}\sum_{T\in\mathcal{S}} u_p(\rho_T^\star), \quad \textit{Round-Trip} = \frac{1}{|\mathcal{T}|}\sum_{T\in\mathcal{S}} u_{rt}(\rho_T^\star),$$

$$\widetilde{u}_{c,T} = \begin{cases}\min\big(\tilde{u}_c(\rho_T^\star), P_{\max}\big), & T\in\mathcal{S},\\ P_{\max}, & T\notin\mathcal{S}\end{cases} \quad \text{Price} = \left(\prod_{T\in\mathcal{S}}\min\big(\tilde{u}_c(\rho_T^\star), P_{\max}\big)\cdot P_{\max}^{|\mathcal{T}|-|\mathcal{S}|}\right)^{1/|\mathcal{T}|}. \tag{13}$$

Better-Than-Ref: The other metrics compare planners with one another. This one compares them with chemistry that has already been carried out: the benchmark ships with a published route for each target, and we ask how many targets the planner recovers a route that scores better than the published one. We denote ρ_T^{ref} for the published route of T,

$$\textit{Better-Than-Ref} = \frac{|\{T\in\mathcal{S} : s(\rho_T^\star) > s(\rho_T^{ref})\}|}{|\mathcal{T}|} \tag{14}$$

An unsolved target, or one whose representative route ties with the reference, counts as a loss.

1stCutDiff measures diversity at the initial retrosynthetic decision. For each recovered route ρ of target T, we identify the reaction whose product is T and define $\text{Prec}_1(\rho)$ as the set of canonicalized precursor molecules produced by this first disconnection. Given the set of distinct recovered routes $\mathcal{R}_T$, the target-level score is

$$\textit{1stCutDiff}(T) = \frac{1}{\binom{|\mathcal{R}_T|}{2}}\sum_{\{\rho_i,\rho_j\}\subseteq\mathcal{R}_T}\mathbb{I}[\text{Prec}_1(\rho_i)\neq\text{Prec}_1(\rho_j)]. \tag{15}$$

Thus, two routes are considered different at the first cut only when they decompose the target into different precursor sets, regardless of how their downstream synthesis steps differ. We compute *1stCutDiff* only for targets with $|\mathcal{R}_T|\geq 2$ and report the macro-average over these targets. We additionally report *1stCuts*, the number of distinct first-cut precursor sets found for each target, *1stCuts*$(T) = |\{\text{Prec}_1(\rho) : \rho\in\mathcal{R}_T\}|$. While *1stCutDiff* captures pairwise diversity, *1stCuts* measures the absolute breadth of initial synthesis branches reached by the planner.

Table 4: Success rate (%) under increasing search budgets on USPTO-190 and ChEMBL-1000, with R-SMILES as the single-step model for all methods. The search budget N_B is the number of unique single-step model calls. Best and second-best results are in **bold** and underlined, respectively.

Method	Success Rate (%) at Search Budget N_B					
	$\mathbf{N_B = 50}$	$\mathbf{N_B = 100}$	$\mathbf{N_B = 200}$	$\mathbf{N_B = 300}$	$\mathbf{N_B = 400}$	$\mathbf{N_B = 500}$
USPTO-190						
Search Algorithms						
MCTS	44.74	52.11	58.95	67.89	71.58	72.11
Retro*-0	52.63	59.47	63.68	70.00	72.11	74.74
Retro*	57.89	64.21	71.58	79.47	80.53	84.74
MORetro*	67.89	81.58	88.95	91.58	94.74	96.32
LLM-based Search						
Retro-R1	82.11	93.68	96.84	98.95	98.95	98.95
RetroAgent	**96.84**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
RetroPlanner	92.11	97.37	98.95	98.95	98.95	98.95
ChEMBL-1000						
Search Algorithms						
MCTS	59.30	64.30	70.60	78.10	82.20	82.40
Retro*-0	72.70	78.00	80.00	81.50	82.20	82.50
Retro*	74.90	79.40	80.50	81.70	83.00	83.40
MORetro*	77.80	84.30	89.40	91.30	92.40	93.00
LLM-based Search						
Retro-R1	82.90	89.10	91.70	92.20	92.30	92.40
RetroAgent	**95.90**	**98.10**	**99.60**	**99.70**	**99.90**	**99.90**
RetroPlanner	90.00	92.40	94.80	95.90	96.30	96.50

A.5 Additional Results

Baseline Comparison. Following the evaluation protocol of RetroAgent (Zhu et al., 2026), we compare RetroPlanner with conventional search algorithms and recent LLM-based planners on USPTO-190 and ChEMBL-1000, as reported in Table 4. For all methods, we use R-SMILES (Zhong et al., 2022) as the single-step retrosynthesis model and measure the search budget N_B by the number of unique single-step model calls. *Success Rate* is defined as the percentage of targets for which at least one complete route is recovered within the given budget. As the budget increases, the success rate of RetroPlanner rises to 98.95% and 96.50% at $N_B = 500$ on USPTO-190 and ChEMBL-1000, respectively. RetroAgent attains the highest success rates overall, but RetroPlanner remains close while recovering more routes of higher quality.

Single-step Model Variation. Since the planner selects only among the candidates proposed by the single-step model, we also evaluate each planner with the template-based LocalRetro (Chen & Jung, 2021) under the same budget of $N_B = 50$ as shown in Table 5. Restricting candidates to the known reaction templates of LocalRetro lowers the performance of all three planners, which perform best with R-SMILES in the main comparison. With both models, RetroPlanner attains the highest *BTR*, plausibility, and round-trip consistency, while RetroAgent leads in success rate and starting-material price. MORetro* recovers by far the most routes but solves considerably fewer targets than the LLM-based planners. The quality gains of RetroPlanner therefore hold for both the template-based and template-free single-step models.

Round-Trip Model Variation. We evaluate the robustness of round-trip consistency to the choice of forward model by replacing ReactionT5v2 (Sagawa & Kojima, 2025) with RXNGraphormer (Xu et al., 2025) and recomputing both *Round-Trip* and *Better-Than-Ref.* As shown in Table 6, RetroPlanner retains strong performance under this change, achieving 58.59% Round-Trip and 64.21% *BTR*, compared with 61.11% and 66.84% under ReactionT5v2. Notably, RetroPlanner remains stronger on both metrics than MORetro*, which explicitly optimizes the same route-quality objectives during search. While MORetro* drops from 58.94% to 47.17% in Round-Trip, RetroPlan-

Table 5: Effect of the single-step retrosynthesis (SSR) model on USPTO-190 with $N_B = 50$. Each planner is paired with a template-based (LocalRetro) and a template-free (R-SMILES) SSR model. **BTR** (*Better-Than-Ref*) is the percentage of targets whose best recovered route achieves a higher composite score than the reference route, and **Plaus.**, **RT**, and **Price** denote the reaction plausibility, round-trip consistency, and starting-material price of the recovered routes. Best and second-best results within each SSR model are in **bold** and underlined, respectively.

SSR Model	Method	Search Outcome		Route Quality			
		Succ. ↑	Routes ↑	BTR ↑	Plaus. ↑	RT ↑	Price ↓
LocalRetro *(template-based)*	MORetro*	55.79%	**20.99**	43.16%	53.41%	44.96%	326.36
	RetroAgent	**88.95%**	4.08	38.42%	74.29%	32.96%	**195.36**
	RetroPlanner	83.16%	5.06	**53.16%**	**79.45%**	**50.38%**	207.95
R-SMILES *(template-free)*	MORetro*	67.89%	**43.16**	52.11%	66.16%	58.94%	250.63
	RetroAgent	**96.84%**	3.07	46.84%	85.85%	38.16%	**138.83**
	RetroPlanner	92.11%	7.41	**66.84%**	**86.03%**	**61.11%**	167.80

Table 6: Robustness to the forward model used for round-trip evaluation on USPTO-190. **BTR** (*Better-Than-Ref*) and **RT** (*Round-Trip*) are recomputed using ReactionT5v2 (**RT5**) and RXN-Graphormer (**RXNG**). Δ denotes the percentage-point change from **RT5** to **RXNG**.

Method	BTR (%) ↑			RT (%) ↑		
	RT5	RXNG	Δ	RT5	RXNG	Δ
MCTS	27.89	26.32	−1.57	25.88	24.21	−1.67
Retro*-0	34.21	31.05	−3.16	38.99	33.96	−5.03
Retro*	38.95	36.32	−2.63	41.64	34.78	−6.86
MORetro*	52.11	50.53	−1.58	58.94	47.17	−11.77
Retro-R1	38.42	34.74	−3.68	39.34	37.56	−1.78
RetroAgent	46.84	46.32	−0.52	38.16	38.03	−0.13
RETROPLANNER	**66.84**	**64.21**	−2.63	**61.11**	**58.59**	−2.52

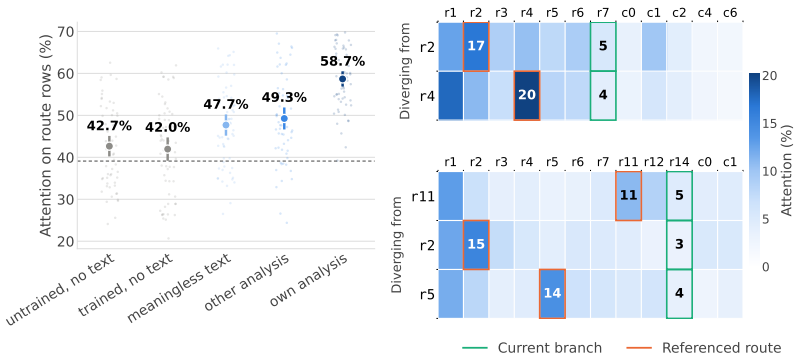

Figure 5: *Left:* Attention from action tokens to the route rows of the input search graph, under different contents of the reasoning field. Attention is highest when the field holds the instance's own analysis, compared with no reasoning before or after training, meaningless text, or the analysis of another instance. *Right:* Attention from `diverging from` tokens to the rows of the search graph in two examples. Attention concentrates on the referenced route (orange) rather than the current branch (green).

NER decreases only from 61.11% to 58.59%. These results indicate that the route-quality gains of RETROPLANNER persist under an alternative forward evaluator, rather than being specific to the model that provides the round-trip signal during search.

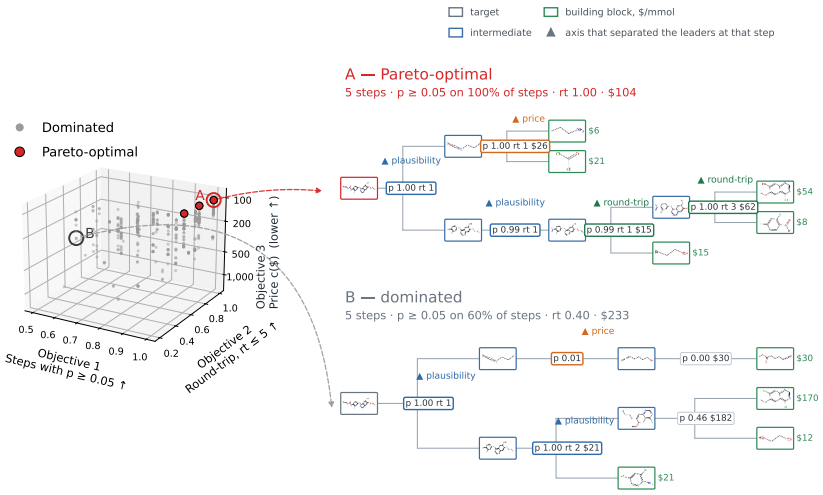

Figure 6: From step-level decisions to route-level objectives for a single target. *Left:* Routes recovered by RETROPLANNER on three route-level objectives: the fraction of steps with plausibility $p \geq 0.05$, the fraction of steps with round-trip consistency $rt \leq 5$, and the total starting-material price c, where lower is better. Pareto-optimal routes are shown in red and dominated routes in gray. *Right:* A Pareto-optimal route (A) and a dominated route (B) for the same target. Each reaction is annotated with its p, rt, and building-block price, and ▲ marks the objective that separated the leading candidates at that step. In route A, choices justified by different objectives at each step combine into a route that is not dominated on any objective.

Effect of Reasoning on Decisions. To examine whether the reasoning trace informs the action rather than merely preceding it, we measure the attention that action tokens place on the route rows of the input search graph while varying the content of the reasoning field, as shown on the left of Figure 5. Training alone does not change this attention, and filling the field with meaningless text or with the analysis of another instance raises it only slightly. Attention to route rows is highest only when the field holds the instance's own analysis, indicating that the action draws on the content of its reasoning rather than on its mere presence. The `diverging from` field shows the same grounding at a finer level: as shown on the right of Figure 5, its tokens attend most strongly to the route they reference rather than to the current branch, tying the stated route relation to the corresponding part of the search graph.

A.6 PROMPTS

Input Prompt — System / Developer / User

System
You are ChatGPT, a large language model trained by OpenAI. Knowledge cutoff: 2024-06. Current date: 2024-06-01. Reasoning: medium. Valid channels: analysis, commentary, final; a channel must be included for every message, and calls to `functions` must go to the commentary channel.

Developer

 # Instructions
Solve the retrosynthesis search by operating on the board.

 # Board semantics
A molecule is an OR: any one reaction under it closes it. A reaction is an AND: every piece under it has to reach purchasable material, or the reaction dies and takes its branch with it. A purchasable piece is closed on arrival. Molecules are shared, so two routes can differ at one reaction and agree everywhere else.
`OPEN` lists every molecule still awaiting a decision. `open` requests the candidate disconnections for a molecule that has no menu. `rank` names the candidates worth taking at a molecule and applies the first `take` of them now (default 1); the rest are the other routes this molecule can give, each one checked to reach purchasable material and carried by number. `done` claims a route: a candidate number for every molecule the route leaves to make, as one object. Handing over is the last message, not a call — it ends the episode and names the routes you claimed, best first.
Every call carries ONE argument, `actions`, and it is always a list:

```
{"actions": [{"type": "open", "mid": "u6"}]}
{"actions": [{"type": "rank", "mid": "u6", "order": [0, 2],
"take": 1}]}
{"actions": [{"type": "done", "choices": {"u6": 0, "n4": 2}}]}
```

`take` above 1 expands several subtrees under one molecule in the same call, so more than one of its cuts is live at once. Actions apply in the order you write them, at most one per molecule per call, and a bare action object is not a call. The only actions are `open`, `rank` and `done`: a branch that cannot close is left alone and the board stops offering it. Nothing moves without a call, and the board never repeats a menu you have already been shown.

 # Board notation
`<mol e3>SMILES</mol>` is a molecule at first appearance and `<mol e3/>` afterwards, the id two characters assigned once. `c0..c9` are candidate numbers, fixed when the molecule is opened and in the model's own order, so a score is not monotone in the index; outside its own block a candidate is `e3.c1`. `depth D` is how far below the target the piece sits. `rN e3.c1 a of b closed` is the ledger: which candidate was spent, how much of the AND is closed, and the ordering you declared. `CLOSED` is every purchasable leaf banked so far with its price; `ROUTES` lists completed routes together with their route-level scores and terminal leaves.

 # The signals
`q` is the single-step model's own confidence in this disconnection; it cannot see what is behind a precursor. `p` is the reaction filter's plausibility with a cut-off of 0.05 — above it, only this one step is said to go, nothing about where the precursors can go. `rt` is the rank at which the forward model returns the product from these precursors; 1 reads cleanly forwards and `---` means it could not be recovered. `*` marks a fragment already purchasable, saying nothing about cost. `($N)` is what a fragment costs on a 0-to-1 scale, and lower is better — it is a cost, the opposite direction from `q` and `p`. The line already carries the total: the figure on a cut is computed from its pieces, so quote it rather than one you worked out.

 # How a route is judged
`ROUTES` carries three axes for every route claimed, all on the line, and the three are compared without a fixed priority: `p 5/6 pass`, how many steps clear the filter, with

`worst at r12` identifying the lowest-plausibility step; `rt 4/6 back`, how many steps the forward model recovers; and `$.218`, what its leaves cost. `6 steps` is printed too and is not one of the three — a shorter route can be beaten on all three and still be the one to run, which is a real argument but not a dominance argument. A route beats another when it is at least equal on all three and better on one, over the routes this board has claimed. Two routes that read the same on all three are one answer wearing two labels. Hand over the routes that are not dominated on the three objectives.

 # Tools

```
## functions

namespace functions {

// Apply actions to the retrosynthesis board and receive the board
that results. Nothing on the board moves without a call.
type board_act = (_: {
  // Actions applied in the order given. At most one per molecule
  -- so a single call may open several molecules at once, and
  usually should: the pieces waiting on the board are the frontier,
  and opening them together brings their disconnections back at the
  same depth, where the next turn can rank rival cuts against each
  other instead of committing to one branch and discovering later
  that it was the wrong one.
  //
  // Every call is wrapped, never a bare action:
  //     {"actions": [ACTION, ...]}
  //
  // Each ACTION is an object:
  //     {
  //       "type":     "open" | "rank" | "done",
  //       "mid"?:     string,                          // open,
  rank
  //       "order"?:   number[],                        // rank
  //       "take"?:    number,                          // rank,
  default 1
  //       "choices"?: {molecule id: candidate number}  // done
  //     }
  actions: any[],
}) => any;

} // namespace functions
```

User

```
OPEN
  m2   depth 0 | no candidates yet
       <mol m2>CN(C)C(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O)NO)n1</mol>
```

Rollout Example — open / rank / done

Turn 3 — assistant (commentary)

```
{"actions": [{"type": "rank", "mid": "uy", "order": [2, 3, 0],
"take": 2},
             {"type": "rank", "mid": "pg", "order": [9, 0, 2],
             "take": 1}]}
```

Turn 3 — tool

```
[Event] uy ranked [c2, c3, c0] | c2 applied -> r3 | pieces:
          <mol px>CNC</mol>
          buyable | $.000
          <mol ba>O=C(C[C@@H](CCCC1CCCCC1)c1nc(C(=O)O)no1)NOC1CCCCO
          1</mol>   open | depth 2
```

```
        r3 1 of 2 closed
[Event] uy ranked [c2, c3, c0] | c3 applied -> r4 | pieces:
          <mol
          d5>CN(C)C(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O)O)n1</mol>
          open | depth 2
          <mol e7>NOC1CCCCO1</mol>
          buyable | $.076
        r4 1 of 2 closed
[Event] pg ranked [c9, c0, c2] | c9 applied -> r5 | pieces:
          <mol xs>CCI</mol>   buyable | $.014
          <mol d5/>           open | depth 2
        r5 1 of 2 closed

r1  m2.c1  0 of 1 closed        m2 ranked: c1 | c2 | c0
r2  m2.c2  1 of 2 closed        m2 ranked: c1 | c2 | c0
r3  uy.c2  1 of 2 closed        uy ranked: c2 | c3 | c0   (was -)
r4  uy.c3  1 of 2 closed        uy ranked: c2 | c3 | c0   (was -)
r5  pg.c9  1 of 2 closed        pg ranked: c9 | c0 | c2   (was -)

OPEN
  ba   under r3 | depth 2 | no candidates yet
  d5   under r4 | depth 2 | no candidates yet

CLOSED  ea*($.000) | px*($.000) | e7*($.076) | xs*($.014)
```

Turn 4 — assistant (analysis)
opening together: ba under r3, d5 under r4 | r3 and r4 are 1 of 2 closed, each owing its single open piece; d5 is shared by r4 and r5, so opening both keeps the uy branch's two cuts alive at one depth and lets the next turn rank them side by side.

Turn 4 — assistant (commentary)


```
{"actions": [{"type": "open", "mid": "d5"}, {"type": "open", "mid":
"ba"}]}
```


Turn 4 — tool

```
r1  m2.c1  0 of 1 closed        m2 ranked: c1 | c2 | c0
r2  m2.c2  1 of 2 closed        m2 ranked: c1 | c2 | c0
r3  uy.c2  1 of 2 closed        uy ranked: c2 | c3 | c0
r4  uy.c3  1 of 2 closed        uy ranked: c2 | c3 | c0
r5  pg.c9  1 of 2 closed        pg ranked: c9 | c0 | c2

OPEN
  ba   under r3 | depth 2
    <c0 q.509 p.723 rt2 $1.000>COC(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O
    )NOC2CCCCO2)n1($1.000)</c0>
    <c1 q.460 p.619 rt5 $.955>CCOC(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O
    )NOC2CCCCO2)n1($.955)</c1>
    <c2 q.006 p.993 rt-
    $.634>COC(=O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O)O)no1($.633) +
    NOC1CCCCO1*($.076)</c2>
    <c3 q.006 p.978 rt2 $.654>NOC1CCCCO1*($.076) +
    O=C(O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O)O)no1($.653)</c3>
    <c4 q.006 p.619 rt- $.955>CCOC(=O)c1noc(C(CCCC2CCCCC2)CC(=O)NOC
    2CCCCO2)n1($.955)</c4>
    <c5 q.005 p.723 rt- $1.000>COC(=O)c1noc([C@@H](CCCC2CCCCC2)CC(=
    O)NOC2CCCCO2)n1($1.000)</c5>
    <c6 q.005 p.000 rt1 $1.000>CC(C)(C)OC(=O)c1noc([C@H](CCCC2CCCCC
    2)CC(=O)NOC2CCCCO2)n1($1.000)</c6>
    <c7 q.001 p.723 rt- $1.000>COC(=O)c1noc(C(CCCC2CCCCC2)CC(=O)NOC
    2CCCCO2)n1($1.000)</c7>
    <c8 q.001 p.993 rt-
    $.634>COC(=O)C[C@H](CCCC1CCCCC1)c1nc(C(=O)O)no1($.633) +
    NOC1CCCCO1*($.076)</c8>
    <c9 q.001 p1.000 rt2 $.521>C1=COCCC1*($.031) +
    O=C(C[C@@H](CCCC1CCCCC1)c1nc(C(=O)O)no1)NO($.520)</c9>
```

```
  d5   under r4 | depth 2
    <c0 q.504 p.024 rt1 $.546>CN(C)C(=O)c1noc([C@H](CCCC2CCCCC2)CC(
    =O)OC(C)(C)C)n1*($.546)</c0>
    <c1 q.394 p.999 rt1 $.646>COC(=O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O)
    N(C)C)no1($.646)</c1>
    <c2 q.047 p.933 rt1
    $.576>CCOC(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O)O)n1($.575) +
    CNC*($.000)</c2>
    <c3 q.046 p.985 rt1 $.589>CCOC(=O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O
    )N(C)C)no1($.589)</c3>
    <c4 q.005 p.979 rt1 $.667>CNC*($.000) +
    COC(=O)c1noc([C@H](CCCC2CCCCC2)CC(=O)O)n1($.667)</c4>
    <c5 q.001 p.999 rt-
    $.646>COC(=O)CC(CCCC1CCCCC1)c1nc(C(=O)N(C)C)no1($.646)</c5>
    <c6 q.001 p.998 rt1 $.653>CNC*($.000) +
    O=C(O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O)O)no1($.653)</c6>
    <c7 q.001 p1.000 rt1 $.575>CNC*($.000) +
    O=C(O)C[C@@H](CCCC1CCCCC1)c1nc(C(=O)Cl)no1($.575)</c7>
    <c8 q.001 p.024 rt- $.546>CN(C)C(=O)c1noc(C(CCCC2CCCCC2)CC(=O)O
    C(C)(C)C)n1*($.546)</c8>
    <c9 q.001 p.024 rt- $.546>CN(C)C(=O)c1noc([C@@H](CCCC2CCCCC2)CC
    (=O)OC(C)(C)C)n1*($.546)</c9>

CLOSED  ea*($.000) | px*($.000) | e7*($.076) | xs*($.014)
```

Turn 5 — assistant (analysis)
`done` on m2·c1, uy·c3, d5·c0: the route ends on the purchasable leaves ea, px, e7, xs. d5 is made at r4, not bought, so it is not a leaf.
Turn 5 — assistant (commentary)

```
{"actions": [{"type": "done", "choices": {"m2": 1, "uy": 3, "d5":
0}}]}
```

Reasoning Schema — MOLECULE / DISCONNECTIONS / DECIDING

Turn 8 — assistant (analysis)

MOLECULE
target: oxadiazole bearing a dimethylamide, a cyclohexyl-substituted carbon, and a hydroxamic acid; two ring systems, one aromatic.
this piece: the same oxadiazole and cyclohexane, plus a methyl ester, a free hydroxyl, and a tetrahydropyranyl-protected hydroxamate; the ester and the pyran are handles this route installed.
not in the target: a methyl ester and a tetrahydropyran ring.

DISCONNECTIONS OF g3
c0 | bond: forms C-N at the hydroxamate carbonyl carbon and the amine nitrogen; breaks C-O at the hydroxyl oxygen | reaction: Carboxylic acid with primary amine to amide | what it does: unbalanced, fragments of 24 and 8 heavy atoms; the ring skeleton changes.
c1 | bond: forms C-N at the hydroxamate carbonyl carbon and the amine nitrogen; breaks C-O at the ester oxygen | reaction: Aminolysis of esters | what it does: unbalanced, fragments of 25 and 8 heavy atoms; the ring skeleton changes.
c2 | bond: forms C-N at the hydroxamate carbonyl carbon and the amine nitrogen; breaks C-O at the hydroxyl oxygen | reaction: Carboxylic acid with primary amine to amide | what it does: unbalanced, fragments of 24 and 8 heavy atoms; the ring skeleton changes.
c6 | bond: forms C-O at the pyran carbon; changes the order of a C-C bond | reaction: oxa-Michael addition | what it does: unbalanced, fragments of 25 and 6 heavy atoms; the ring skeleton changes.

DECIDING g3
route 1 – c6
diverging from: c1, c2 | both are amide couplings of the pyran amine | this one closes the

pyran by oxa-Michael addition, the amide already in place.
ordered by: round-trip | value: 1 | it reads back to the product, which the amide cuts do not.
route 2 – c1
diverging from: r2 | same step, Aminolysis of esters, Ester with primary amine to amide | it re-forms the hydroxamate from the ethyl ester, a split the oxa-Michael route never makes.
ordered by: confidence | value: 0.320 | the two amide cuts read the same on the scored axes, so the single-step model's own reading separates them.
route 3 – c2
diverging from: r4 | same step, Carboxylic acid with primary amine to amide | it re-forms the hydroxamate from the free acid, the other side of the same amide bond.
ordered by: structure | value: none | it beats the head on no number; it is the acid-side coupling, the bond the head leaves untouched.

Turn 8 — assistant (commentary)


```
{"actions": [{"type": "rank", "mid": "g3", "order": [6, 1, 2],
"take": 1}]}
```


...

Turn 21 — assistant (analysis)

MOLECULE
target: an oxadiazole bearing a dimethyl amide, a cyclohexyl-substituted side chain, and a hydroxamic acid; two ring systems, one aromatic and one aliphatic, one stereocentre.
this piece: g3, the oxadiazole core with the cyclohexyl side chain, an ethyl ester in place of the dimethyl amide, and a tetrahydropyranyl-protected hydroxamate; three ring systems, one stereocentre.
not in the target: an ethyl ester and a tetrahydropyranyl protecting group, both handles to be consumed.

DISCONNECTIONS OF g3
c0 | bond: forms C-N at the hydroxamate carbonyl C; breaks C-O at the hydroxyl O | reaction: Carboxylic acid with primary amine to amide | what it does: splits it into fragments of 24 and 8 heavy atoms, one piece carrying most of it; the ring skeleton changes.
c1 | bond: forms C-N at the hydroxamate carbonyl C; breaks C-O at the ester O | reaction: Aminolysis of esters, Ester with primary amine to amide | what it does: splits it into fragments of 25 and 8 heavy atoms, one piece carrying most of it; the ring skeleton changes.
c5 | bond: forms C-N; breaks C-O | reaction: Aminolysis of esters, Ester with primary amine to amide | what it does: splits it into fragments of 26 and 8 heavy atoms, one piece carrying most of it; the ring skeleton changes.

DISCONNECTIONS OF ba
c0 | bond: breaks C-O at the sp3 C bearing the hydroxyl O | reaction: Cleavage of methoxy ethers to alcohols, Ester saponification (alkyl deprotection), Ester saponification (methyl deprotection) | what it does: one precursor of 30 heavy atoms; the ring skeleton is kept.
c8 | bond: forms C-N at the hydroxamate carbonyl carbon and the amine nitrogen; breaks C-O at the ester oxygen | reaction: Aminolysis of esters, Ester with primary amine to amide | what it does: unbalanced, fragments of 24 and 8 heavy atoms; the ring skeleton changes.

DISCONNECTIONS OF d5
c0 | bond: breaks C-O at the sp3 C bearing the hydroxyl O | reaction: Deprotection of carboxylic acid, Ester saponification (alkyl deprotection) | what it does: one precursor of 28 heavy atoms; the ring skeleton is kept.
c5 | bond: breaks C-O | reaction: Cleavage of methoxy ethers to alcohols, Ester saponification (alkyl deprotection), Ester saponification (methyl deprotection), Ester to carboxylate | what it does: one precursor of 25 heavy atoms; the ring skeleton is kept.

DECIDING g3
route 1 – c5
diverging from: r2 | same step, Aminolysis of esters, Ester with primary amine to amide |

this one cuts the hydroxamate side, so it opens a different branch than the amide aminolysis already expanded.

ordered by: structure | value: none | the tied leaders were separated by the single-step confidence and a missing round-trip, neither a scored axis, so the cut's own bond change is what it is kept for.

route 2 – c0

diverging from: r4 | same step, Carboxylic acid with primary amine to amide | this one keeps the ester and cuts the acid side, so it is the acid-side partner to the ester-side cut taken above.

ordered by: round-trip | value: 1 | the forward model recovers it cleanly, which the ester-side cut above, at rank two, does not.

DECIDING ba

route 1 – c8

diverging from: c2 | the same amide cut on the epimeric acid | c8 keeps the [C@H] centre the target carries, c2 inverts it.

ordered by: structure | value: none | against the plausibility-tied c2, the retained stereocentre decides.

DECIDING d5

route 1 – c5

diverging from: none | nothing on this board has yet cut this acid, so there is no earlier step to differ from.

ordered by: plausibility | value: .999 | the filter clears the step, and it is the only untried cut here that reaches the free acid.

Turn 21 — assistant (commentary)


```
{"actions": [{"type": "rank", "mid": "g3", "order": [5, 0], "take":
2},
             {"type": "rank", "mid": "ba", "order": [8], "take":
             1},
             {"type": "rank", "mid": "d5", "order": [5], "take":
             1}]}
```