ENERGYRETRO: BRIDGING BOND ENERGETICS AND REACTION KNOWLEDGE FOR MULTI-STEP RETROSYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrosynthesis planning aims to identify synthetic routes by recursively transforming a target molecule into accessible starting materials. Although single-step prediction and multi-step search have advanced considerably, bond energetics and prior reaction knowledge are rarely integrated to guide both local disconnection decisions and complete-route planning. To address this gap, we propose **EnergyRetro**, a multi-agent framework that bridges bond dissociation energy (BDE)-derived information with reaction knowledge for multi-step retrosynthesis. EnergyRetro coordinates specialized agents to combine two complementary sources of information: bond energetics and reaction knowledge. Reaction knowledge retrieved from the reaction knowledge graph (RKG) provides precedents for plausible disconnections, while BDE-derived energy profiles assess their energy risk. The two signals are then integrated into reaction costs that guide multi-step planning toward complete synthetic routes. Round-trip validation provides an additional consistency check, while memory enables validated reactions and completed subroutes to be reused. EnergyRetro[1] solves 185 of 190 USPTO-190 targets; removing energy information or RKG retrieval reduces the solve rate by 15.29 and 34.24 percentage points, respectively. It further achieves solve rates of 97.36% and 97.18% on PaRoutes n1 and n5. Finally, we apply EnergyRetro to synthesis planning for a drug candidate targeting SARS-CoV-2 Mpro, demonstrating its applicability to structure-based drug design.

1 INTRODUCTION

Retrosynthesis planning identifies synthetic routes by recursively transforming a target molecule into readily accessible starting materials (Szymkuc et al., 2016). It connects molecular design to the preparation of compounds for drug discovery and natural-product research (Mikulak-Klucznik et al., 2020). Computational planners couple single-step reactant prediction (Tetko et al., 2020) with iterative search (Segler et al., 2018; Chen et al., 2020) to navigate the combinatorial space of possible disconnections. Each local choice introduces precursors whose synthesis must also be resolved, linking prediction quality to the complete routes recovered (Torren-Peraire et al., 2024).

Agent-based planners use language models to coordinate prediction, search, and intermediate planning states (Liu et al., 2025; Song et al., 2026; Zhu et al., 2026). The conventional workflow in Figure 1(a) illustrates an agent querying a template-guided predictor, selecting reactants, and invoking search tools in a predict–select–expand cycle. At each expansion, the planner must judge both whether a candidate disconnection is plausible and whether the resulting partial route should be explored further (Choe et al., 2025). However, the information used to evaluate individual reactions is often not incorporated into subsequent route-level decisions. This creates a gap between local reaction assessment and the construction of complete synthetic routes.

Reaction knowledge and bond energetics provide two complementary sources of information for bridging this gap. Reaction precedents show how related molecules have been transformed in known reactions and provide evidence for candidate disconnections (Coley et al., 2017; Xie et al.,

[1]Code is available at `https://anonymous.4open.science/r/EnergyRetro`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

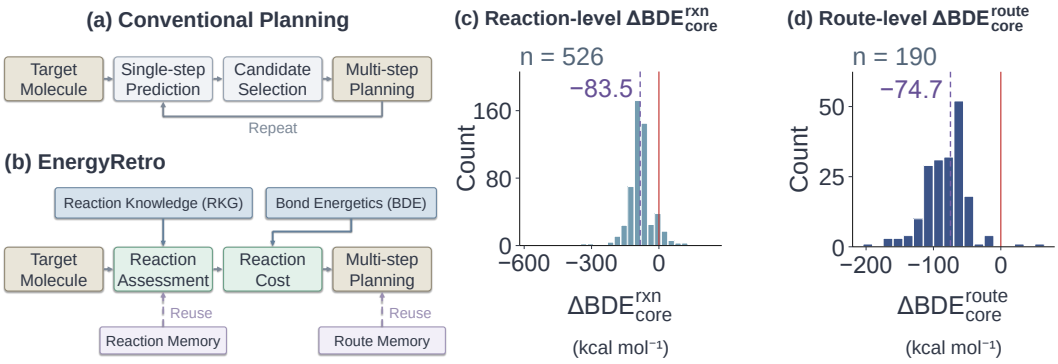

Figure 1: **Motivation and overview of EnergyRetro.** **(a)** Conventional Planning connects the target molecule to single-step prediction, candidate selection, and multi-step planning through a repeat loop. **(b)** EnergyRetro uses reaction knowledge for Reaction Assessment and bond energetics for Reaction Cost, guiding Multi-step Planning. Reaction Memory and Route Memory support reuse in assessment and planning, respectively. **(c)** Reaction-level $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ and **(d)** route-level $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{route}}$ on USPTO-190. Route-level values are averaged over reaction steps containing retained-skeleton edits; purple dashed lines mark medians and red lines mark zero.

2023). Bond dissociation energies (BDEs) characterize bond strengths in their molecular environments (St. John et al., 2020), and bond-energy features have improved reaction-center prediction (Zhao et al., 2025). While reaction knowledge provides evidence from known transformations, bond energetics characterizes the bond changes involved in a candidate reaction. At the reaction level, Figure 1(c) shows a median retained-skeleton $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ of -83.5 kcal mol^{-1} across 526 reference reactions. After step normalization, the 190 reference routes in Figure 1(d) have a median $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{route}}$ of -74.7 kcal mol^{-1}. Both distributions remain broad, showing substantial variation in bond-energy changes across reactions and routes. These observations motivate combining BDE-derived energy information with reaction precedents when assessing candidate disconnections. Corresponding PaRoutes distributions are provided in Appendix C.

Motivated by these complementary sources of information, we propose **EnergyRetro**, a multi-agent framework that bridges BDE-derived bond energetics with reaction knowledge for multi-step retrosynthesis (Figure 2). Reaction knowledge retrieved from a reaction knowledge graph (RKG) provides precedents for candidate disconnections, while BDE-derived energy profiles characterize their bond changes and support energy-risk assessment. Specialized agents combine this information to propose, evaluate, and refine candidate reactions. The resulting energy risk and RKG support, together with round-trip consistency, are integrated into reaction costs that guide multi-step planning by prioritizing partial routes and constructing complete synthetic routes. Reaction Memory provides reusable single-step precedents, while Route Memory reuses completed subroutes within each target's search. EnergyRetro solves 185 of 190 USPTO-190 targets, and removing energy information or RKG retrieval reduces the solve rate by 15.29 and 34.24 percentage points, respectively. It further achieves solve rates of 97.36% and 97.18% on PaRoutes n1 and n5, and plans a four-step synthetic route for a generated candidate targeting SARS-CoV-2 Mpro (Jin et al., 2020). The main contributions are as follows:

- We propose **EnergyRetro**, a multi-agent framework that bridges reaction knowledge with BDE-derived bond energetics to jointly support the proposal, assessment, and refinement of retrosynthetic disconnections.
- We connect local reaction assessment with multi-step planning by integrating energy risk, RKG support, and round-trip consistency into reaction costs. These costs are used in a constraint-guided AND-OR planner to prioritize partial routes and construct complete synthetic routes, while memory enables reaction and subroute reuse.
- Experiments on USPTO-190, ChEMBL-1000, and PaRoutes demonstrate the effectiveness of EnergyRetro, with controlled ablations validating the contributions of both energy information and RKG retrieval and an Mpro case study demonstrating its applicability to structure-based drug design.

2 Related Work

Reaction knowledge and bond energetics. RetroSim uses molecular similarity to retrieve reaction precedents, while RetroKNN retrieves local reaction templates to improve single-step prediction (Coley et al., 2017; Xie et al., 2023). KnowRetro incorporates hierarchical chemical knowledge from molecules, substructures, and functional groups (Chen et al., 2026). Bond energetics provides a complementary source of chemical information: Retro-MTGR incorporates bond-energy features for reaction-center prediction (Zhao et al., 2025). In a different line of work, energy-based models use learned energy functions to formulate or rerank single-step predictions (Sun et al., 2021; Lin et al., 2022). These methods improve single-step retrosynthesis, but stronger local predictions do not necessarily translate into better route recovery, since every generated precursor must ultimately be resolved into available starting materials (Torren-Peraire et al., 2024). EnergyRetro bridges reaction knowledge and bond energetics by using RKG evidence and BDE-derived energy profiles to refine candidate reactions, and carrying their assessments into reaction costs for multi-step planning.

Multi-step retrosynthesis planning. Neural-guided MCTS and Retro* couple single-step prediction with tree search (Segler et al., 2018; Chen et al., 2020), while EG-MCTS learns from search experience and PDVN uses learned estimates of synthesizability and synthesis cost to guide planning (Hong et al., 2023; Liu et al., 2023). LLM-based methods add route refinement, tool interaction, and structured memory (Wang et al., 2025; Liu et al., 2025; Zhu et al., 2026); LARC uses tool-grounded judgments to enforce constraints (Baker et al., 2026). Graph-based planners also exploit intermediate reuse: RetroGraph shares intermediate molecules across routes, while AOT* combines retrieved synthesis examples with AND-OR search and intermediate reuse (Xie et al., 2022; Song et al., 2026). However, these planners mainly prioritize routes through search heuristics, learned costs, constraints, or reuse, without explicitly combining bond energetics and reaction-knowledge support at the reaction level. EnergyRetro instead incorporates BDE-derived energy risk, RKG support, and round-trip consistency into reaction costs that directly guide constraint-guided AND-OR search, with Route Memory reusing completed subroutes.

3 Method

Problem formulation. Given a target molecule m_0 and a stock $\mathcal{B}$ of purchasable starting materials, retrosynthesis planning seeks complete synthetic routes to m_0. We represent a candidate reaction as $r = (m, \mathcal{P}_r)$, where m is the product and $\mathcal{P}_r$ contains all required precursor molecules. A synthetic route R is rooted at m_0: each expanded molecule selects one reaction, and every precursor of that reaction must be supplied by stock or a further reaction. A partial route retains unresolved precursor leaves; a complete route has all terminal molecules in $\mathcal{B}$. Within a depth limit $d_{\max}$ and an expansion limit $E_{\max}$, EnergyRetro collects up to J distinct complete routes and ranks them by accumulated reaction cost, $g(R) = \sum_{r \in R} c(r)$.

EnergyRetro consists of two foundational components and an agentic planning pipeline for multi-step retrosynthesis (Figure 2). The framework relies on two infrastructure components: the *Reaction Energy Tool* (Figure 2a) constructs BDE-derived energy profiles from mapped bond changes, while the *Reaction Knowledge Graph (RKG)* (Figure 2a) organizes reactions, molecules, fragments, and memory for knowledge-guided retrieval. Using the resulting energy profiles and retrieved reaction knowledge, the *Agentic Single-step Expansion* module (Figure 2b) coordinates specialized agents to propose and refine candidate reactant sets. These candidates are then assessed by *Energy-aware Evaluation* (Figure 2c) to assign reaction costs based on energy risk, RKG support, and round-trip consistency. Finally, the *Multi-step Retrosynthesis* planner (Figure 2d) performs constraint-guided AND-OR search, using these costs to iteratively expand partial routes and reuse completed subroutes until complete synthetic routes are obtained.

3.1 Reaction Energy Tool & Reaction Knowledge Graph

The framework relies on two foundational components: the reaction energy tool characterizes bond changes in candidate reactions, while the RKG retrieves relevant reaction precedents and supporting chemical context for the agents.

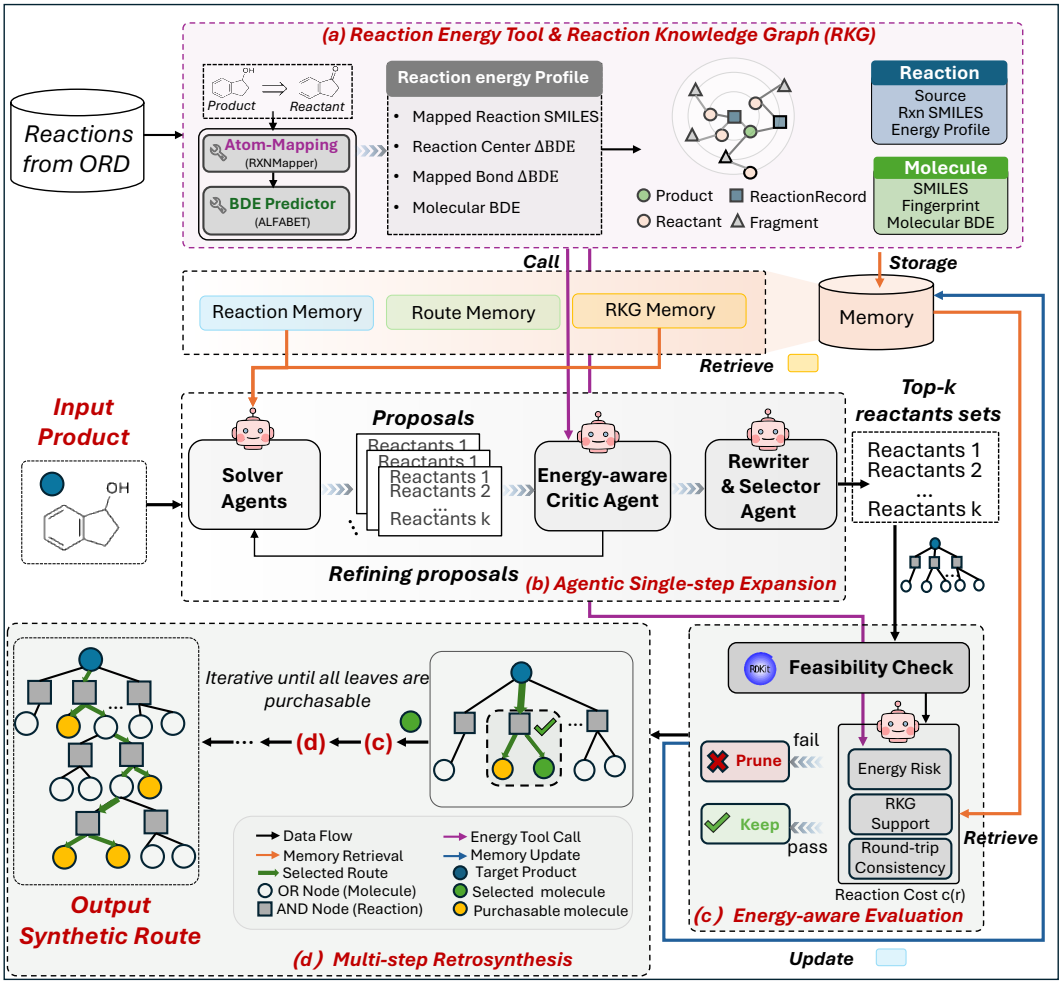

Figure 2: **Overview of the EnergyRetro framework.** **(a)** The Reaction Energy Tool constructs BDE-derived energy profiles, and the RKG organizes reaction knowledge and memory for retrieval. **(b)** Specialized agents propose and refine candidate reactant sets. **(c)** Energy-aware Evaluation computes reaction costs from energy risk, RKG support, and round-trip consistency. **(d)** Constraint-guided AND-OR search constructs complete synthetic routes with subroute reuse.

Reaction energy profiles. For a candidate $r = (m, \mathcal{P}_r)$, RXNMapper aligns reactant and product atoms to identify broken bonds $\mathcal{B}_b$, formed bonds $\mathcal{B}_f$, and bonds whose order changes $\mathcal{B}_c$. ALFABET predicts the corresponding bond dissociation energies (BDEs) (St. John et al., 2020). In the reactants-to-product direction, the reaction-level difference is

$$\Delta\text{BDE} = \sum_{e\in\mathcal{B}_b} D_R(e) - \sum_{e\in\mathcal{B}_f} D_P(e) + \sum_{e\in\mathcal{B}_c} \big[D_R(e) - D_P(e)\big], \tag{1}$$

where $D_R(e)$ and $D_P(e)$ denote BDEs in the reactant and product environments. The three change sets are disjoint; unchanged bonds are excluded. The tool returns an energy profile containing the mapped reaction, bond changes, predicted BDEs, and ΔBDE.

RKG structure and retrieval. The RKG connects Open Reaction Database (ORD) records (Kearnes et al., 2021) with participating molecules and fragments. *ReactionRecord* nodes store reaction source, reaction SMILES, and available reaction-level energy profiles; *Compound* nodes store molecular structures, canonical identifiers, fingerprints, and bond-level BDE information; *Fragment* nodes represent molecular substructures used for retrieval. The typed relations are *ReactionRecord–CONSUMES–Compound*, *ReactionRecord–PRODUCES–Compound*, and *Compound–HAS_FRAGMENT–Fragment*. For a query molecule, the RKG supports four com-

plementary retrieval strategies: exact SMILES matching, canonical-SMILES matching, fingerprint-based similarity retrieval, and fragment-based retrieval. Each matched product is traced through the typed relations to recover the associated reaction record and complete precursor set, providing reaction precedents for candidate generation and refinement. RKG construction, benchmark overlap filtering, graph schema, and retrieval queries are detailed in Appendix A.

RKG Memory. Reaction Memory stores validated single-step outcomes collected during training as *ReactionRecords* distinguished by source. It is frozen for evaluation and retrieved alongside ORD records. Route Memory stores completed synthetic routes and subroutes as connected *ReactionRecords* with all required precursor branches. It is initialized empty for each target and updated only within that target's search; completed subroutes can then be retrieved when the same intermediate reappears. Details of memory construction and reuse are provided in Appendix B.3.

3.2 Agentic Single-Step Expansion

Using the energy profiles and reaction precedents obtained from Section 3.1, this module coordinates specialized agents to propose and refine reactant sets for an unresolved molecule m (Figure 2b). It develops proposals through feedback and rewriting, then returns up to K reactant sets for evaluation.

Proposing disconnections. The input is checked while retaining original and canonical SMILES for retrieval. The Solver Agents use five complementary proposal strategies. Four correspond to the RKG retrieval strategies described above, while the fifth combines retrieved precedents with the reaction energy tool to incorporate bond-energy information during proposal generation. Each strategy returns an initial disconnection, its complete precursor set, and the supporting RKG evidence and/or mapped bond changes.

Refining proposals. The Energy-aware Critic Agent examines each proposal's molecular structures, retrieved reaction evidence, and BDE-derived energy profile. Molecular Transformer round-trip validation (Schwaller et al., 2019) supplies a forward-predicted product for comparison with m. Based on these complementary sources of information, the Critic provides proposal-specific feedback on the disconnection and precursor set. Proposals are iteratively retained or revised over the configured feedback rounds, after which the current proposals and accumulated feedback are passed to the Rewriter and Selector Agent.

Rewriting and selecting candidates. The Rewriter generates n variants of each current proposal using its retrieved reaction evidence, energy profile, and Critic feedback. Canonicalization merges duplicate reactant sets while retaining their evidence; rewritten structures receive updated molecular checks, energy profiles, and forward predictions. The Selector ranks the resulting candidate pool using structural consistency, retrieved reaction support, energy information, and Critic feedback, retaining up to K reactant sets. The selected candidates are then passed to the Energy-aware Evaluation module for reaction-cost computation.

3.3 Energy-aware Evaluation

The Energy-aware Evaluation module converts the chemical evidence associated with each retained candidate into a quantitative reaction cost $c(r)$ for downstream route planning (Figure 2c). For each structurally valid reaction, it receives the product and precursors, energy profile, retrieved *ReactionRecords*, and forward-prediction results. The Evaluation Agent assigns three normalized scores in $[0, 1]$, each accompanied by the evidence used for the assessment.

Three complementary assessments. Each candidate is evaluated from three complementary perspectives: 1) The **energy-risk score** $e(r)$ assesses the bond changes of the proposed reaction using the mapped broken, formed, and bond-order-changed bonds, their predicted BDEs, and $\Delta\mathrm{BDE}$, together with available precedents for comparable transformations. A higher value indicates greater assessed energy risk. 2) The **RKG-support score** $s(r)$ measures the support provided by retrieved reaction records, considering the relevance of the transformations, the correspondence of products and reactants, and the retrieval relations connecting the candidate to its precedents. Stronger reaction support yields a higher score. 3) The **round-trip consistency score** $s_{\mathrm{RT}}(r)$ measures the agreement between the forward-predicted product and the target molecule. An exact canonical match yields a score of one, while partial correspondence receives a lower score.

Computing reaction costs. The executor validates the three scores and combines them into the reaction cost

$$c(r) = 1 + \lambda_E e(r) + \lambda_G\big[1 - s(r)\big] + \lambda_{\mathrm{RT}}\big[1 - s_{\mathrm{RT}}(r)\big], \tag{2}$$

with nonnegative weights fixed before planning. This formulation ensures that every reaction incurs a base step penalty of 1, implicitly encouraging the search algorithm to find shorter routes. The energy risk $e(r)$ directly inflates the cost, whereas the confidence metrics—RKG support and round-trip consistency—are inverted via $(1 - s)$ to penalize candidates lacking historical or structural validation. Appendix B details the fixed scoring configuration and evaluation cache.

3.4 Multi-Step Retrosynthesis

The multi-step planner assembles evaluated reactions into complete synthetic routes using constraint-guided AND-OR search (Figure 2d). In this search space, molecules act as **OR nodes** (offering alternative reaction pathways), while reactions act as **AND nodes** (requiring the successful synthesis of every precursor branch). Each search state represents a partial route that selects one reaction per expanded molecule and tracks all unresolved precursors.

Prioritizing partial routes. The planner maintains a queue of partial routes initialized with the target m_0. To determine the expansion priority, it employs an A*-like best-first priority function for a route R with unresolved leaf occurrences $\mathcal{L}(R)$: $\mathcal{L}(R)$:

$$F(R) = \sum_{r \in R} c(r) + \sum_{m \in \mathcal{L}(R)} h(m), \qquad h(m) = 1. \tag{3}$$

The first term accumulates the reaction costs defined in Section 3.3, while the second estimates the remaining cost of unresolved leaves. Because each unresolved leaf requires at least one additional reaction and every reaction has a base cost of at least 1, $h(m) = 1$ provides a simple lower bound on its remaining cost. Stock molecules are considered resolved and contribute zero remaining cost.

Expansion and constraints. At each step, the planner dequeues the partial route with the lowest $F(R)$ and selects an unresolved leaf. If no compatible stored subroute is found, it invokes the Agentic Single-step Expansion (Section 3.2) and Evaluation modules to generate new candidates, reusing cached outputs for recurring molecules. Each valid candidate generates a successor state by adding the required precursors, provided it satisfies molecular validity, depth limits ($d_{\max}$), and avoids cyclic ancestor matches. Because $c(r)$ incorporates energy risk, RKG support, and round-trip consistency, these reaction-level assessments directly influence the priority of each successor route.

Subroute reuse and termination. To prevent redundant exploration, the framework utilizes **Route Memory**, which starts empty for each new target. When a subroute is successfully completed, the memory agent extracts and stores its full structural topology and reaction costs. When the planner encounters the same intermediate again, it retrieves the lowest-cost compatible subroute from Route Memory and attaches it to the current partial route. Each new invocation of the complete single-step module counts toward the limit $E_{\max}$. The search terminates when J distinct complete routes have been found or the queue is exhausted, and the returned routes are ranked by their accumulated reaction cost $g(R)$. Appendix B.4 specifies leaf selection, queue ordering, and deduplication details.

4 Experiments

4.1 Experimental Setup

Implementation settings. For multi-step benchmarks, DeepSeek-v4-flash (temperature 0.2) powered five Solver Agents. Each produced one initial proposal; each retrieval strategy returned at most $k_{\mathrm{ret}} = 10$ rows. After up to 5 Critic rounds, the Rewriter produced 5 variants per proposal and the Selector retained up to 10 reactant sets. The search collected up to $J = 5$ distinct complete routes. The maximum route depth and expansion budget followed the corresponding benchmark settings, while the reaction-cost weights were fixed to $\lambda_E = 0.3$, $\lambda_G = 0.5$, and $\lambda_{\mathrm{RT}} = 0.5$.

Datasets. We evaluate EnergyRetro on three benchmarks covering planning coverage and reference-route recovery. USPTO-190 contains 190 targets evaluated under the Retro* protocol (Chen et al., 2020). ChEMBL-1000 contains 1,000 target molecules sampled from the ChEMBL bioactivity

database (Gaulton et al., 2017) without reference synthetic routes (Liu et al., 2025). PaRoutes n1 and n5 each provide 10,000 patent-derived reference routes, enabling evaluation of both target coverage and exact route recovery (Genheden & Bjerrum, 2022). We follow the official target sets and building-block stocks of each benchmark. Further details on baseline models and evaluation metrics are provided in Appendices B.5 and B.6, respectively.

Table 1: Multi-step planning results on USPTO-190 and ChEMBL-1000. Solve rates (SR, %) use method-specific planning budgets N; N_{LLM} counts external LLM calls. For USPTO-190, † marks route-level expansion for N; unmarked methods use single-step expansion. Average calls cover all or solved targets; tokens are totals in millions (M). A slash (/) denotes an unavailable result. For both solve rates and resource values, **best** and second-best results are shown in bold and underlined.

	USPTO-190								ChEMBL-1000		
	SR@N ↑			SR@N_{LLM} ↑		Avg. calls ↓		Token ↓	SR@N ↑		
Method	**100**	**300**	**500**	**100**	**300**	**All**	**Solved**	**(M)**	**100**	**300**	**500**
Greedy DFS	19.5	19.5	19.5	/	/	303.2	74.2	/	38.1	38.1	38.1
Retro*	50.0	74.7	79.5	/	/	138.1	102.3	/	69.0	73.2	74.4
EG-MCTS	64.7	75.3	81.1	/	/	61.7	41.3	/	74.2	78.3	79.3
PDVN	72.1	87.9	92.1	/	/	**52.3**	**36**	/	76.4	79.6	80.6
Retro-R1	77.2	84.5	87.0	74.7	79.5	157.5	87.1	**41.9**	80.2	82.6	83.7
RetroAgent	70.5	81.6	85.1	68.4	71.1	178.8	72.7	44.2	79.6	**83.2**	**84.4**
LLM(MCTS)	26.3	27.4	31.1	26.3	27.4	342.1	121.1	58.5	/	/	/
LLM(Retro*)	25.8	26.8	32.6	25.8	26.8	264.6	102.6	45.2	/	/	/
AOT*†	82.1	**92.6**	93.1	83.16	**90.53**	105.9	39.2	51.4	79.7	82.4	83.4
LLM-Syn-Planner†	64.7	91.1	92.1	52.6	63.1	213.2	92.9	99.0	77.1	80.7	82.8
EnergyRetro	**85.8**	92.1	**97.4**	**84.21**	88.95	132.6	68.4	48.3	**80.3**	82.4	84.0

4.2 Multi-Step Planning on USPTO-190 and ChEMBL-1000

We evaluated EnergyRetro against search- and LLM-based baselines on USPTO-190. As shown in Table 1, EnergyRetro achieves solve rates of 85.8%, 92.1%, and 97.4% under planning budgets of $N = 100, 300$, and 500, respectively. At $N = 500$, it achieves the highest solve rate, exceeding AOT* (93.1%) by 4.3 percentage points. Meanwhile, EnergyRetro achieves 92.1% solve rate when $N = 300$, exceeding PDVN (87.9%) and LLM-Syn-Planner (91.1%). We also validate the contribution of energy guidance through ablation studies by removing BDE-based energy (Section 4.4). EnergyRetro achieves a solve rate of 88.95% when limiting external LLM calls to $N_{\mathrm{LLM}} = 300$, demonstrating robust performance with limited reasoning calls. At $N_{\mathrm{LLM}} = 100$, it ranks first with 84.21%, exceeding AOT* (83.16%). Retro-R1 and RetroAgent use fewer tokens (41.9M and 44.2M), with lower solve rates of 87.0% and 85.1% at $N = 500$. In contrast, EnergyRetro achieves near-complete route coverage (97.4%) with moderate token consumption (48.3M tokens), outperforming AOT* (51.4M) and LLM-Syn-Planner (99.0M) in terms of solve performance. Overall, these results demonstrate that BDE-derived energy guidance, combined with knowledge-guided retrieval and agentic planning, enables EnergyRetro to identify complete synthetic routes with high target coverage and competitive token consumption.

We further evaluate EnergyRetro on ChEMBL-1000 to examine its planning performance on a larger target set. As shown in Table 1, EnergyRetro achieves solve rates of 80.3%, 82.4%, and 84.0% at $N = 100, 300, 500$, respectively. At $N = 100$, it achieves the highest solve rate, exceeding PDVN (76.4%), EG-MCTS (74.2%), and Retro* (69.0%). This result demonstrates effective route discovery under a limited planning budget. Additional search increases coverage by 3.7 percentage points, while the advantage over PDVN remains 3.4 percentage points at $N = 500$. Thus, EnergyRetro benefits from continued exploration while maintaining its coverage advantage as the search budget grows. At $N = 500$, it ranks second to RetroAgent (84.4%) and ahead of Retro-R1 (83.7%), AOT* (83.4%), and LLM-Syn-Planner (82.8%). Together with the USPTO-190 results, these findings support the effectiveness of EnergyRetro's combination of energy-guided reaction assessment and knowledge-guided planning across both target sets.

Table 2: Results (%) on PaRoutes n1/n5 (Genheden & Bjerrum, 2022). R@1/R@10: exact top-1/top-10 reference-route recovery. ‡ Results from the official PaRoutes 2.0 benchmark. LLM-based methods are evaluated in this work. **Best** and second-best results are bold and underlined.

	PaRoutes n1			PaRoutes n5		
Method	**SR**	**R@1**	**R@10**	**SR**	**R@1**	**R@10**
MCTS‡	97.16	23.72	54.14	96.89	12.37	40.56
Retro*‡	97.28	20.27	48.47	**97.29**	11.43	38.97
DFPN‡	77.86	17.05	24.60	67.30	7.53	11.51
Retro-R1	97.06	33.89	57.85	96.53	24.51	43.43
RetroAgent	96.08	31.02	54.24	94.85	29.99	45.06
AOT*	92.99	27.38	55.39	93.84	29.43	48.56
LLM-Syn-Planner	96.54	30.94	51.40	96.55	**30.80**	50.06
EnergyRetro	**97.36**	**34.11**	**59.12**	97.18	30.03	**51.23**

4.3 Generalization on PaRoutes

We evaluate EnergyRetro on PaRoutes using both target solve rate and reference-route recovery (Table 2). EnergyRetro achieves the highest R@10 on n1 and n5, reaching 59.12% and 51.23%, respectively. These scores exceed Retro-R1 and LLM-Syn-Planner, the strongest respective baselines, by 1.27 and 1.17 percentage points. On n5, EnergyRetro, Retro*, and MCTS all solve over 96% of targets, but Retro* and MCTS achieve R@10 scores of 38.97% and 40.56%. For the first proposed route, its recovery rates are 34.11% on n1 and 30.03% on n5, ranking first and second, respectively. Alongside solve rates of 97.36% and 97.18%, these results demonstrate broad target coverage and effective reference-route recovery across both subsets.

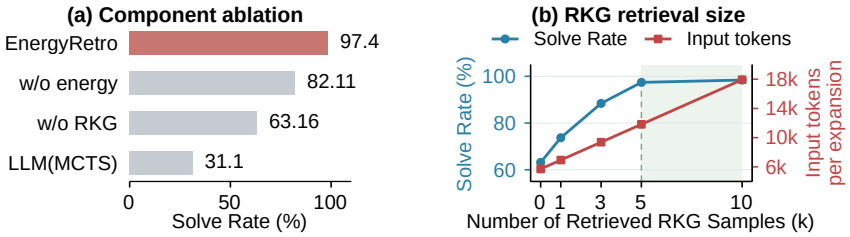

Figure 3: Component ablation and RKG retrieval-budget analysis on USPTO-190. **(a)** Solve rates of EnergyRetro, its *w/o energy* and *w/o RKG* variants, and the LLM(MCTS) reference baseline. **(b)** Solve rate (left axis) and mean input tokens per single-step expansion (right axis) as the per-strategy RKG retrieval limit k_{ret} varies.

4.4 Ablation Study

Energy and RKG ablation. We ablate energy guidance and RKG retrieval on USPTO-190 while keeping all other components unchanged. As shown in Figure 3(a), removing energy guidance or RKG retrieval reduces the solve rate from 97.4% to 82.11% and 63.16%, respectively, corresponding to drops of 15.29 and 34.24 percentage points. The results show that energy guidance improves planning with retrieved precedents, while the larger drop without RKG highlights the importance of reaction knowledge. LLM(MCTS) achieves 31.1%.

RKG retrieval budget. We vary the per-strategy retrieval limit k_{ret} to study the trade-off between solve rate and input-token cost. Figure 3(b) shows that the solve rate increases from 63.16% at $k_{\text{ret}} = 0$ to 88.4% at 3 and 97.4% at 5. Increasing k_{ret} to 10 maintains the same solve rate but raises mean input tokens per expansion from 11,822 to 17,932 (+51.7%), indicating that five retrieved records provide comparable coverage at lower cost.

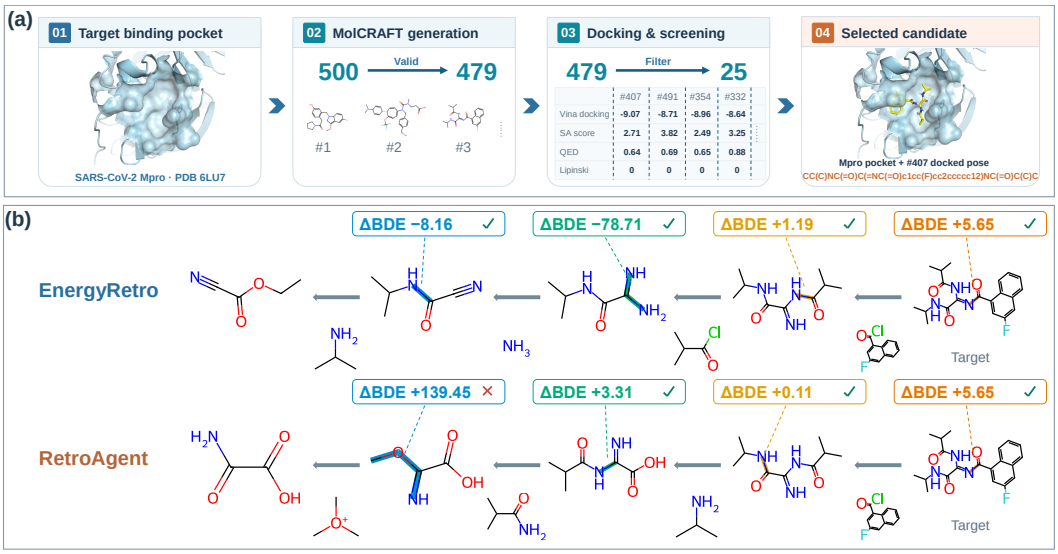

Figure 4: Structure-based candidate generation and retrosynthetic planning for SARS-CoV-2 Mpro. **(a)**: Generation and screening pipeline leading to the selected candidate (#407). **(b)**: Four-step retrosynthetic routes from EnergyRetro and RetroAgent, where colored bonds and ΔBDE values highlight the energetic characteristics of successive disconnections, and check marks and crosses indicate the round-trip validation results.

4.5 Case Study

We examine EnergyRetro in a structure-based drug design scenario targeting SARS-CoV-2 Mpro. As shown in Figure 4, MolCRAFT (Qu et al., 2024) generates 500 candidates from the N3-bound Mpro pocket (PDB ID: 6LU7) (Jin et al., 2020), and 479 valid molecules proceed to screening. Based on docking score, synthetic accessibility, drug-likeness, and scaffold diversity, 25 candidates are retained and #407 is selected for retrosynthetic planning due to its excellent profile (Vina docking score of -9.07, SA score of 2.71, QED of 0.64, and 0 Lipinski violations). EnergyRetro identifies a four-step route with forward-synthesis ΔBDE values of -8.16, -78.71, $+1.19$, and $+5.65$ kcal/mol. Round-trip validation also achieves exact top-1 product recovery for all four reactions, confirming consistency with the forward reaction model. In contrast, RetroAgent proposes an alternative route (with forward-synthesis ΔBDE values of $+139.45$, $+3.31$, $+0.11$, and $+5.65$ kcal/mol) that achieves exact recovery for three reactions; its initial step fails round-trip validation and exhibits an extreme positive ΔBDE of $+139.45$ kcal/mol. This comparison illustrates how energetic profiles and forward-prediction consistency jointly guide multi-step route exploration, helping EnergyRetro prioritize chemically plausible transformations rather than only score completed routes.

5 Conclusion

We presented EnergyRetro, an energy-guided multi-agent framework that integrates reaction energetics with structured reaction knowledge for multi-step retrosynthesis planning. EnergyRetro combines BDE-derived energy profiles with reaction precedents retrieved from the RKG to evaluate candidate disconnections, while round-trip validation and memory reuse further support reliable and efficient route construction. These signals are incorporated into a constraint-guided AND–OR search, enabling reaction-level chemical evidence to directly guide complete-route planning. Experiments across three benchmarks demonstrate the effectiveness and generalization of EnergyRetro, with ablations confirming the complementary contributions of energy information and RKG retrieval. The case study further demonstrates its applicability to synthesis planning for generated drug candidates. Overall, EnergyRetro couples chemically informed reaction evaluation with global retrosynthetic search, providing a unified knowledge- and energy-aware framework for multi-step synthesis planning.

AI USE STATEMENT

Large language models (LLMs) were used in EnergyRetro to propose and assess candidate reactions during retrosynthetic planning. Retro-R1, RetroAgent, AOT*, and LLM-Syn-Planner are LLM-based retrosynthesis planning methods; we reproduced them as baselines for comparison. The authors take responsibility for the reported results and final manuscript.

REFERENCES

Frazier N. Baker, Daniel Adu-Ampratwum, Reza Averly, Botao Yu, Huan Sun, and Xia Ning. LARC: Towards human-level constrained retrosynthesis planning through an agentic framework. In *Proceedings of the AI for Accelerated Research Symposium*, volume 3 of *EPiC Series in Technology*, pp. 153–176, 2026.

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, 2020.

Yujie Chen, Tengfei Ma, Zhou Yu, Jiayi Zhang, Ajie Lin, Shu Wu, Dongsheng Cao, and Yiping Liu. Beyond reaction data: Learning chemical knowledge from large-scale molecules for retrosynthesis. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 10681–10692. ACM, 2026.

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, 2025.

Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science*, 3(12):1237–1245, 2017.

Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. The ChEMBL database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954, 2017. doi: 10.1093/nar/gkw1074.

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.

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.

Zhenming Jin, Xiaoyu Du, Yechun Xu, Yongqiang Deng, Meiqin Liu, Yao Zhao, Bing Zhang, Xiaofeng Li, Leike Zhang, Chao Peng, Yinkai Duan, Jing Yu, Lin Wang, Kailin Yang, Fengjiang Liu, Rendi Jiang, Xinglou Yang, Tian You, Xiaoce Liu, Xiuna Yang, Fang Bai, Hong Liu, Xiang Liu, Luke W. Guddat, Wenqing Xu, Gengfu Xiao, Chengfeng Qin, Zhengli Shi, Hualiang Jiang, Zihe Rao, and Haitao Yang. Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. *Nature*, 582(7811):289–293, 2020.

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

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22266–22276, 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, pp. 70709–70737, 2025.

Barbara Mikulak-Klucznik, Patrycja Gołębiowska, Alison A. Bayly, Oskar Popik, Tomasz Klucznik, Sara Szymkuć, Ewa P. Gajewska, Piotr Dittwald, Olga Staszewska-Krajewska, Wiktor Beker, Tomasz Badowski, Karl A. Scheidt, Karol Molga, Jacek Mlynarski, Milan Mrksich, and Bartosz A. Grzybowski. Computational planning of the synthesis of complex natural products. *Nature*, 588(7836):83–88, 2020.

Yanru Qu, Keyue Qiu, Yuxuan Song, Jingjing Gong, Jiawei Han, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. MolCRAFT: Structure-based drug design in continuous parameter space. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41749–41768, 2024.

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

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.

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, 2026.

Peter C. St. John, Yanfei Guan, Yeonjoon Kim, Seonah Kim, and Robert S. Paton. Prediction of organic homolytic bond dissociation enthalpies at near chemical accuracy with sub-second computational cost. *Nature Communications*, 11(1):2328, 2020.

Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Towards understanding retrosynthesis by energy-based models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10186–10194, 2021.

Sara Szymkuc, Ewa P. Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michal Startek, Michal Bajczyk, and Bartosz A. Grzybowski. Computer-assisted synthetic planning: The end of the beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, 2016.

Igor V. Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11 (1):5575, 2020.

Amol Thakkar, Thierry Kogej, Jean-Louis Reymond, Ola Engkvist, and Esben Jannik Bjerrum. Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain. *Chemical Science*, 11(1):154–168, 2020. doi: 10.1039/C9SC04944D.

Paula Torren-Peraire, Alan Kai Hassen, Samuel Genheden, Jonas Verhoeven, Djork-Arné Clevert, Mike Preuss, and Igor V. Tetko. Models matter: The impact of single-step retrosynthesis on synthesis planning. *Digital Discovery*, 3(3):558–572, 2024.

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, 2025.

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. ACM, 2022.

Shufang Xie, Rui Yan, Junliang Guo, Yingce Xia, Lijun Wu, and Tao Qin. Retrosynthesis prediction with local template retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5330–5338, 2023.

Peng-Cheng Zhao, Xue-Xin Wei, Qiong Wang, Qi-Hao Wang, Jia-Ning Li, Jie Shang, Cheng Lu, and Jian-Yu Shi. Single-step retrosynthesis prediction via multitask graph representation learning. *Nature Communications*, 16(1):814, 2025.

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 Third Conference on Language Modeling*, 2026.

APPENDIX

A REACTION KNOWLEDGE GRAPH DETAILS

A.1 CONSTRUCTION AND STATISTICS

We construct the Reaction Knowledge Graph (RKG) from 2,351,064 reaction records across 546 datasets in the Open Reaction Database (ORD), linking 1,948,388 compounds and 85 RDK fragment types. Table 3 summarizes the nodes and typed relations supporting reaction retrieval. *ReactionRecords* store reaction identity, source, and dataset metadata. *Compound* nodes carry molecular structures, canonical SMILES, InChIKeys, Morgan fingerprints, and *Fragment* annotations. *ReactionRecords* connect to reactant and product *Compounds* through CONSUMES and PRODUCES relations, respectively. The HAS-FRAGMENT relations connect *Compounds* nodes to *Fragment* nodes. The energy tool computes molecular BDEs and reaction-center energy profiles during candidate evaluation. The ORD data repository is available at `https://github.com/open-reaction-database/ord-data`.

Table 3: Statistics of node and relation types in the Reaction Knowledge Graph (RKG).

Type	Label or relation	Count
Node	*ReactionRecord*	2,351,064
Node	*Compound*	1,948,388
Node	*Fragment*	85
Edge	CONSUMES	11,277,661
Edge	PRODUCES	2,554,007
Edge	HAS-FRAGMENT	7,334,204

Compound retrieval uses exact SMILES, canonical SMILES, folded Morgan vectors, and retained *Fragment* anchors. The query specification is provided below in Sections A.3–A.6.

Table 4 reports all 85 *Fragment* types and their coverage in the RKG. *Compound* counts denote distinct *Compound* nodes connected to each *Fragment*; product-compound counts denote those appearing in at least one PRODUCES relation. Producing-reaction counts denote distinct *ReactionRecords* producing those *Compounds*.

Table 4: *Fragment* types and coverage in the RKG. Comp., Prod. comp., and Prod. rxn. denote *Compounds*, product *Compounds*, and producing *ReactionRecords*, respectively.

Fragment	**Comp.**	**Prod. comp.**	**Prod. rxn.**
`fr_Al_COO`	113,760	81,370	153,587
`fr_Al_OH`	236,279	174,536	290,564
`fr_Al_OH_noTert`	197,119	144,982	243,349
`fr_ArN`	143,602	102,213	174,249
`fr_Ar_COO`	75,725	54,079	86,568
`fr_Ar_N`	37,734	30,597	71,473
`fr_Ar_NH`	176,096	131,411	215,228
`fr_Ar_OH`	111,782	79,365	135,748
`fr_COO`	188,401	134,575	238,602
`fr_COO2`	190,345	134,952	240,639
`fr_C_O`	43,839	33,641	94,092
`fr_C_O_noCOO`	39,443	30,360	76,259
`fr_C_S`	18,392	13,372	20,037
`fr_HOCCN`	15,577	12,198	20,439
`fr_Imine`	46,251	33,941	51,947

Continued on next page

Table 4 (continued)

Fragment	Comp.	Prod. comp.	Prod. rxn.
fr_NH0	60,153	47,516	108,471
fr_NH1	44,038	34,161	75,137
fr_NH2	276,231	193,915	325,930
fr_N_O	21,202	15,862	28,065
fr_Ndealkylation1	84,299	67,260	105,435
fr_Ndealkylation2	192,684	152,472	241,296
fr_Nhpyrrole	176,096	131,411	215,228
fr_SH	8,819	4,667	7,867
fr_aldehyde	48,977	33,820	62,250
fr_alkyl_carbamate	65,808	43,928	68,640
fr_alkyl_halide	277,424	198,339	319,389
fr_allylic_oxid	104,215	72,808	118,208
fr_amide	24,738	19,828	43,828
fr_amidine	29,215	22,513	32,938
fr_aniline	26,644	21,565	46,963
fr_aryl_methyl	486,027	367,779	607,010
fr_azide	8,986	6,548	10,084
fr_azo	1,568	997	1,598
fr_barbitur	548	403	591
fr_benzene	61,627	47,917	112,852
fr_benzodiazepine	2,307	1,535	2,413
fr_bicyclic	35,540	28,993	67,682
fr_diazo	185	114	255
fr_dihydropyridine	2,685	1,880	2,992
fr_epoxide	7,781	5,041	8,949
fr_ester	365,047	250,390	409,128
fr_ether	37,178	28,466	66,223
fr_furan	38,611	29,302	49,355
fr_guanido	10,740	8,311	14,205
fr_halogen	30,355	23,478	115,543
fr_hdrzine	20,311	14,277	22,122
fr_hdrzone	13,367	10,371	15,091
fr_imidazole	146,975	112,195	178,063
fr_imide	37,478	27,722	44,741
fr_isocyan	2,883	1,243	2,532
fr_isothiocyan	1,947	1,058	1,788
fr_ketone	160,741	112,418	190,300
fr_ketone_Topliss	125,849	87,272	149,452
fr_lactam	12,298	8,396	16,531
fr_lactone	25,655	18,850	32,215
fr_methoxy	408,609	298,775	488,836
fr_morpholine	46,199	37,950	60,004
fr_nitrile	123,861	94,986	151,985
fr_nitro	103,606	72,902	126,402
fr_nitro_arom	81,838	57,037	98,899
fr_nitro_arom_nonortho	42,898	29,489	51,784
fr_nitroso	1,043	766	1,211
fr_oxazole	27,070	20,940	30,513
fr_oxime	22,134	17,102	29,398
fr_para_hydroxylation	227,354	173,191	282,771
fr_phenol	88,070	63,062	108,363
fr_phenol_noOrthoHbond	86,029	61,608	105,879
fr_phos_acid	2,061	1,352	3,073
fr_phos_ester	2,003	1,315	2,175
fr_piperdine	179,175	141,009	223,359

Continued on next page

Table 4 (continued)

Fragment	Comp.	Prod. comp.	Prod. rxn.
fr_piperzine	92,057	74,834	110,396
fr_priamide	39,722	30,884	48,561
fr_prisulfonamd	8,842	6,462	10,821
fr_pyridine	417,859	326,050	528,888
fr_quatN	5,028	3,739	8,980
fr_sulfide	98,545	72,302	115,498
fr_sulfonamd	99,513	79,494	130,311
fr_sulfone	45,753	33,760	47,779
fr_term_acetylene	12,161	8,597	17,527
fr_tetrazole	13,471	10,109	16,879
fr_thiazole	74,470	57,924	91,148
fr_thiocyan	743	496	800
fr_thiophene	82,170	62,344	97,173
fr_unbrch_alkane	125,309	90,979	148,300
fr_urea	65,034	53,746	79,205

A.2 Benchmark Overlap Filtering

During RKG construction, we excluded the ORD reaction records matching benchmark reactions in USPTO-190, PaRoutes-n1, and PaRoutes-n5. Table 5 reports the overlap identified in the raw ORD corpus before this filtering step.

Matching was performed directly against the original ORD protocol-buffer files (`.pb.gz`). For each reaction, we removed atom-mapping labels, canonicalized the molecular representations, and sorted and concatenated the reactants and products separately to form a reaction signature. A match required equality of both sides of the signature. On the ORD side, reactants comprised inputs labeled `REACTANT` and inputs with an unspecified reaction role; inputs labeled as reagents, solvents, or catalysts were excluded from the signature.

Table 5: Benchmark overlap with raw ORD before RKG construction filtering. Reaction overlap is the percentage of benchmark reactions with a matching ORD signature. Matched routes contain at least one matching reaction. The corresponding ORD records were excluded during RKG construction.

	Reactions			Synthetic routes	
Benchmark	Total	Matched	Overlap (%)	Total	Matched
USPTO-190	1,268	161	12.70	190	104
PaRoutes-n1	28,951	4,639	16.02	10,000	3,597
PaRoutes-n5	37,243	5,377	14.44	10,000	3,913

Comparison with the 5,007 USPTO50K test reactions after filtering yielded zero exact matches under atom-map-stripped canonical reaction signatures. Canonical products matched for 243 test records (4.85%); zero test reactant sets were fully contained in a product-matched graph reaction.

A.3 Graph Schema and Query Inputs

The Cypher listings describe retrieval over the three-node schema comprising *ReactionRecord*, *Compound*, and *Fragment*. Both reaction edges point from *ReactionRecord* to *Compound*: PRODUCES identifies products, and CONSUMES identifies reactants. A single retrosynthetic step traverses from a product against PRODUCES, then follows CONSUMES to its precursors. *Compound* nodes connect to RDKit functional-group *Fragment* nodes through HAS-FRAGMENT, whose `count` property records the number of occurrences.

Compound identifiers use `smiles:<SMILES>`; corpus reaction identifiers use `ord:<reaction_id>`. Different original SMILES nodes may share a canonical SMILES

representation. Reaction Memory stores validated training outcomes as *ReactionRecords*, with `source` distinguishing these records from ORD reactions. Routes are reconstructed from *ReactionRecords* connected through shared *Compounds*.

Query parameters are prepared using the same RDKit feature procedure as graph construction. The original input is retained as `target_smiles`; molecule parsing followed by canonical SMILES generation supplies `canonical_smiles`. Canonical matching uses equality of these canonical strings. The retrieval limit k_{ret} is passed as `top_k`; K limits final candidates and n specifies rewrites per proposal.

A radius-2, 2048-bit Morgan fingerprint is folded into a 256-dimensional binary vector by mapping each active bit index b to $b \bmod 256$. The corresponding vector entry is set to one when any active bit maps to it. The vector is supplied as `morgan_fp_dense` to the cosine vector index over folded Morgan fingerprints. The index name is supplied through `$vector_index` in Listing 3. Fragment retrieval considers all 85 types listed in Table 4. Fragment parameters contain up to five nonzero fragment types ordered by decreasing target occurrence count and then increasing fragment code. The maximum number of selected anchors is the smaller of two and the number of supplied fragment types.

A.4 Four Retrieval Strategies and Cypher Queries

Each query returns the matched *Compound* identifier, *ReactionRecord* identifier, precursor SMILES, original reaction source, and strategy-specific retrieval score. Precursor collection includes inputs whose `reaction_role` is `REACTANT` or absent, preserving unannotated records for downstream candidate assessment. Each strategy applies a final `LIMIT $top_k` to its compound–reaction rows. The four strategies return at most $4k_{\text{ret}}$ rows in total; application-level deduplication by `reaction_id` forms a unique reaction set. Retrieval scores retain their strategy labels as evidence for the Agents, which assign the final evaluation scores described in the main Method section.

Exact SMILES matching. Listing 1 matches the original SMILES string and assigns score 1.0. For each matched *Compound*, producing reactions are selected in identifier order; the final limit applies across all returned compound–reaction pairs. Identifier ordering provides deterministic selection.

Listing 1: Exact matching of the original product SMILES.

```
MATCH (c:Compound {smiles: $target_smiles})
        CALL (c) {
            MATCH (r:ReactionRecord)-[:PRODUCES]->(c)
            RETURN r
            ORDER BY r.uid
            LIMIT $top_k
        }
        WITH c, r, 1.0 AS score
        WITH c, r, score
OPTIONAL MATCH (r)-[rel:CONSUMES]->(precursor:Compound)
WHERE rel.reaction_role IS NULL OR rel.reaction_role = 'REACTANT'
WITH c, r, score, collect(DISTINCT precursor.smiles) AS precursors
RETURN
    c.uid AS compound_id,
    r.uid AS reaction_id,
    precursors AS precursors,
    r.source AS source,
    score AS score
ORDER BY reaction_id, compound_id
LIMIT $top_k
```

Canonical SMILES matching. Listing 2 matches only the RDKit canonical SMILES of stored products. Each match receives the fixed retrieval-tier score 0.95. Producing reactions are selected by identifier, and the final result is ordered by retrieval score and reaction identifier before applying the result limit.

Listing 2: Matching products by canonical SMILES only.

```
MATCH (c:Compound)
        WHERE c.canonical_smiles = $canonical_smiles
        CALL (c) {
            MATCH (r:ReactionRecord)-[:PRODUCES]->(c)
            RETURN r
            ORDER BY r.uid
            LIMIT $top_k
        }
        WITH c, r, 0.95 AS score
        WITH c, r, score
OPTIONAL MATCH (r)-[rel:CONSUMES]->(precursor:Compound)
WHERE rel.reaction_role IS NULL OR rel.reaction_role = 'REACTANT'
WITH c, r, score, collect(DISTINCT precursor.smiles) AS precursors
RETURN
    c.uid AS compound_id,
    r.uid AS reaction_id,
    precursors AS precursors,
    r.source AS source,
    score AS score
        ORDER BY score DESC, reaction_id
LIMIT $top_k
```

Molecular fingerprint retrieval. In Listing 3, the vector index first retrieves up to k_{ret} *Compounds* by cosine-index score using the folded Morgan vectors. For each retrieved *Compound*, up to three producing reactions are selected in identifier order. *Compounds* with no producing reaction contribute no result rows. The combined compound–reaction rows are ordered by the index score and reaction identifier, then limited to k_{ret}. The score retained in the output is the value returned by the vector index.

Listing 3: Fingerprint retrieval with the cosine vector index.

```
CALL db.index.vector.queryNodes(
            $vector_index,
            $top_k,
            $morgan_fp_dense
        )
        YIELD node AS c, score
        CALL (c) {
            MATCH (r:ReactionRecord)-[:PRODUCES]->(c)
            RETURN r
            ORDER BY r.uid
            LIMIT 3
        }
        WITH c, r, score
        WITH c, r, score
OPTIONAL MATCH (r)-[rel:CONSUMES]->(precursor:Compound)
WHERE rel.reaction_role IS NULL OR rel.reaction_role = 'REACTANT'
WITH c, r, score, collect(DISTINCT precursor.smiles) AS precursors
RETURN
    c.uid AS compound_id,
    r.uid AS reaction_id,
    precursors AS precursors,
    r.source AS source,
    score AS score
        ORDER BY score DESC, reaction_id
LIMIT $top_k
```

Fragment retrieval. Listing 4 counts the number of distinct *Compounds* connected to each supplied fragment type and selects up to two anchors with the lowest database coverage. Matched *Compounds* are ranked by the number of distinct selected anchors they contain, with *Compound* identifiers breaking ties. For each selected product *Compound*, one producing reaction is retained in identifier order. The retrieval score is the fraction of selected anchor types matched by the *Compound*. Target fragment counts determine the initial fragment ordering; matching uses distinct anchor types. An empty set of available anchors yields an empty result.

Listing 4: Retrieval through low-coverage Fragment anchors.

```
UNWIND $top_fragments AS fragment_code
MATCH (f:Fragment {fragment_code: fragment_code})<-[:HAS_FRAGMENT]-(fragment_hit:Compound)
WITH fragment_code, count(DISTINCT fragment_hit) AS fragment_compound_count
ORDER BY fragment_compound_count ASC, fragment_code
LIMIT $substructure_anchor_k
WITH collect(fragment_code) AS anchor_fragments
UNWIND anchor_fragments AS fragment_code
MATCH (f:Fragment {fragment_code: fragment_code})<-[:HAS_FRAGMENT]-(c:Compound)
WITH anchor_fragments, c, count(DISTINCT fragment_code) AS matched_fragment_count
WHERE matched_fragment_count > 0
  AND EXISTS {
      MATCH (:ReactionRecord)-[:PRODUCES]->(c)
  }
ORDER BY matched_fragment_count DESC, c.uid
LIMIT $top_k
CALL (c) {
    MATCH (r:ReactionRecord)-[:PRODUCES]->(c)
    RETURN r
    ORDER BY r.uid
    LIMIT 1
}
OPTIONAL MATCH (r)-[rel:CONSUMES]->(precursor:Compound)
WHERE rel.reaction_role IS NULL OR rel.reaction_role = 'REACTANT'
WITH
    c,
    r,
    anchor_fragments,
    matched_fragment_count,
    collect(DISTINCT precursor.smiles) AS precursors
RETURN
    c.uid AS compound_id,
    r.uid AS reaction_id,
    precursors AS precursors,
    r.source AS source,
    toFloat(matched_fragment_count) / toFloat(size(anchor_fragments)) AS score,
    matched_fragment_count
ORDER BY matched_fragment_count DESC, score DESC, reaction_id
LIMIT $top_k
```

A.5 Retrieval Latency

We measured retrieval latency on 200 deterministically selected product *Compounds* with RDKit-parseable SMILES, distinct original and canonical representations, and at least one *Fragment* connection. Each strategy executed one query per target with a result limit of $k_{\text{ret}} = 100$ compound–reaction pairs. Timing covered the Neo4j query and result consumption, with RDKit input features prepared beforehand.

Table 6: Retrieval latency over 200 targets at $k_{\text{ret}} = 100$. All values are in milliseconds; P50 and P95 denote the 50th and 95th percentiles.

Strategy	**Mean** $\downarrow$	**P50** $\downarrow$	**P95** $\downarrow$
Exact matching	722.7	719.8	759.3
Canonical matching	2.1	2.0	2.9
Similarity retrieval	22.4	18.0	36.0
Substructure retrieval	687.9	652.1	1,130.5

Canonical matching and similarity retrieval had mean latencies of 2.1 and 22.4 ms, respectively (Table 6). Substructure retrieval, which selects low-coverage *Fragment* anchors and retrieves associated product reactions, had a median latency of 652.1 ms.

A.6 Reaction Records and Bounded Path Retrieval

Retrieving a complete reaction record. The returned `reaction_id` is the *ReactionRecord*'s `uid` and can be used for a complete record lookup. Listing 5 aggregates the two sides independently, retaining all input roles, stoichiometric coefficients, product yields, and desired-product annotations when present. Candidate construction applies the precursor-role filter to the returned inputs.

Listing 5: Retrieving reaction properties and both molecular sides.

```
MATCH (r:ReactionRecord {uid: $reaction_id})
CALL (r) {
    OPTIONAL MATCH (r)-[rel:CONSUMES]->(c:Compound)
    RETURN collect(CASE WHEN c IS NOT NULL THEN {
        compound_id: c.uid,
        smiles: c.smiles,
        role: rel.reaction_role,
        coefficient: rel.coefficient
    } END) AS inputs
}
CALL (r) {
    OPTIONAL MATCH (r)-[rel:PRODUCES]->(c:Compound)
    RETURN collect(CASE WHEN c IS NOT NULL THEN {
        compound_id: c.uid,
        smiles: c.smiles,
        coefficient: rel.coefficient,
        yield_percent: rel.yield_percent,
        is_desired_product: rel.is_desired_product
    } END) AS products
}
RETURN r.uid AS reaction_id, properties(r) AS reaction,
       inputs, products
LIMIT 1
```

Retrieving a bounded reaction chain. A path query repeats the directed product–reaction–precursor unit. Each repetition contains one *ReactionRecord* and two graph edges, so reaction depth equals half the graph-path length. Listing 6 uses the Neo4j quantified-path syntax to retrieve one acyclic chain of at most three reaction steps. It returns the reaction identifiers and *Compound* SMILES in traversal order, preferring fewer reaction steps and then reaction identifiers.

Listing 6: One acyclic reaction chain with at most three steps.

```
MATCH p = (target:Compound {uid: $target_uid})
    ((product:Compound)<-[:PRODUCES]-(r:ReactionRecord)
      -[input:CONSUMES]->(precursor:Compound)
      WHERE input.reaction_role IS NULL OR input.reaction_role = 'REACTANT'
    ){1,3}
    (leaf:Compound)
WHERE all(v IN nodes(p) WHERE single(w IN nodes(p) WHERE w = v))
RETURN p,
       length(p) / 2 AS reaction_steps,
       [v IN nodes(p) WHERE v:ReactionRecord | v.uid] AS reaction_ids,
       [v IN nodes(p) WHERE v:Compound | v.smiles] AS compound_smiles
ORDER BY reaction_steps ASC, reaction_ids
LIMIT 1
```

For a general depth limit d_{ret}, a validated positive integer supplies the pattern's repetition bound. All molecule identifiers remain query parameters, and `path_k` may replace the final literal limit to control the number of returned chains. The starting `target_uid` comes from an exact or canonical match when retrieving recorded transformations for the target itself. A starting identifier obtained through fingerprint or Fragment retrieval refers to the matched analogue and supplies its recorded transformations as evidence.

The endpoint can be constrained by `leaf.uid = $end_uid` for a specific *Compound* or `leaf.uid IN $stock_uids` for a supplied stock. A chain ending at a *Compound* with no further recorded transformation has reached a graph endpoint; stock membership is checked separately by the planner. To reconstruct a complete synthetic route, the planner retrieves every selected reaction's full reactant set and resolves all precursor branches in the AND-OR tree. The chain query supplies one connected branch for this process, and the record query supplies the additional partners at each step.

Four-step organic-precursor retrieval. We used depth-first search to retrieve a four-step reaction branch for target P_0 (Listing 7). At each depth, retrieval follows PRODUCES backwards from the product to its *ReactionRecords*, then follows CONSUMES to candidate precursors. The search recursively follows an organic precursor at each step to recover a connected four-step branch.

Table 7 identifies the four *ReactionRecords* connecting P_0 through P_4. The five molecular structures are listed below, providing the selected precursor at each step. Additional co-reactants are obtained

through the complete-record query; the multi-step planner resolves their branches when constructing a synthetic route.

Table 7: Four-step reaction branch retrieved through organic precursors. Each row connects a product to the selected precursor through one *ReactionRecord*.

Step	Product $\rightarrow$ precursor	*ReactionRecord* identifier
1	$P_0 \rightarrow P_1$	`ord:ord-18765e8d08314601b6d4cede53efec69`
2	$P_1 \rightarrow P_2$	`ord:ord-84fb141af9a14f8ab8616a1ed0fc70ff`
3	$P_2 \rightarrow P_3$	`ord:ord-3ed119e974ba44c78b5ddeb51f0189b3`
4	$P_3 \rightarrow P_4$	`ord:ord-a688bdf22bb0415f9f334dbd7991dd4f`

Listing 7: Molecular SMILES for the target and four selected organic precursors.

```
P0 = CCCC[C@@H](C(=O)N1CCC[C@H]1C(=O)O)[C@@H](F)C(=O)OC

P1 = CCCC[C@@H](C(=O)N1CCC[C@H]1C(=O)OC(C)(C)C)[C@@H](F)C(=O)OC

P2 = CCCC[C@@H](C(=O)N1CCC[C@H]1C(=O)OC(C)(C)C)[C@H](O)C(=O)OC

P3 = CCCCC(C(=O)N1CCC[C@H]1C(=O)OC(C)(C)C)C1OC(OC)(c2ccccc2)OC1=O

P4 = CC(C)(C)OC(=O)[C@@H]1CCCN1
```

Execution settings. The vector index is checked for the `ONLINE` state and the expected 256-dimensional configuration before querying. SMILES queries can use property indexes on the corresponding *Compound* fields. The query examples use driver-supplied parameters and a 30-second query timeout as specified in the retrieval guide. Result limits control returned rows; the timeout separately bounds query execution, including path enumeration and sorting. These retrieval settings and source identifiers accompany the returned evidence for later Agent evaluation.

B Multi-Step Planning Details

B.1 Reaction Evidence, Energy, and Round-Trip Scores

For each final candidate reaction, the Evaluation Agent receives the target, proposed reactants, molecular checks, mapped energy profile, retrieved *ReactionRecords*, and forward prediction results. Tools provide the chemical observations; the Agent evaluates them under a shared scoring rubric and returns three scores in $[0, 1]$ with supporting explanations. The executor validates the score fields and applies the reaction-cost formula in the main Method section.

Energy-risk score. The Agent assigns $e(r)$ using the broken and formed bonds, bond-order changes, predicted BDEs, reaction-center ΔBDE, and available precedents for comparable transformations. The score increases with the assessed energy risk of the proposed bond changes relative to comparable reaction precedents. The Agent cites the bond changes and BDE values used in its assessment.

RKG-support score. The Agent assigns $s(r)$ by inspecting the records returned through exact, canonical, fingerprint, and Fragment retrieval. It considers the relevance of the recorded transformation, the correspondence of both reaction sides, and the typed relations connecting the target to its precedents. Directly applicable reaction records support higher scores, whereas partial analogies or mismatching transformations support lower scores. The output cites the supporting *ReactionRecord* identifiers and explains the relationship to the candidate. When retrieval returns no supporting records, the Agent reports zero RKG support and records this observation.

Round-trip consistency score. The Agent receives the forward-predicted product and canonical match flag, and assigns $s_{\mathrm{RT}}(r)$ from the recovered connectivity, substituents, and reaction center. An exact canonical match supports a score of one; partial structural correspondence receives a lower

score, and a clearly incompatible product supports a score near zero. For structurally valid candidates, discrepancies with the target increase the round-trip cost penalty.

Score aggregation and consistency. The Agent returns $e(r)$, $s(r)$, and $s_{\mathrm{RT}}(r)$ together with their evidence identifiers and short justifications. The executor computes $c(r) = 1 + \lambda_E e(r) + \lambda_G[1 - s(r)] + \lambda_{\mathrm{RT}}[1 - s_{\mathrm{RT}}(r)]$ using fixed nonnegative weights. The rubric, model configuration, and weights are held constant during a planning run. Evaluation records are cached by canonical candidate under this configuration. Rewritten candidates receive updated evaluations; incomplete tool results and score fields are completed before search insertion.

B.2 Branch Eligibility and Partial-Route States

After rewriting, each candidate receives molecular-validity and forward-prediction checks. The planner inserts candidates whose required precursors parse successfully and satisfy the depth and ancestor-cycle constraints at that tree position. Energy risk, RKG support, and round-trip consistency determine the reaction cost.

A search state is a partial synthetic route: it selects one reaction at every expanded OR node and retains all precursor children of every selected AND node. Its accumulated cost counts each selected reaction occurrence once. Repeated molecules retain separate ancestor contexts at each tree position and can share cached single-step outputs. Each attached subroute contributes its full cost at that position.

The planner ranks states by accumulated cost plus the number of unresolved leaf occurrences. An unresolved non-stock molecule needs at least one further reaction, and every reaction has cost at least one. Replacing such a leaf by a reaction therefore adds that reaction's cost and the estimates for all new unresolved precursors. Queue ties are broken by fewer unresolved leaves and then insertion order; leaf-selection ties use canonical SMILES and tree position. Partial states and completed routes are deduplicated by canonical rooted route structure, retaining reaction choices and precursor multiplicity.

B.3 Memory, Caching, and Termination

Figure 5 illustrates how Reaction Memory and Route Memory share the RKG representation. The blue highlight identifies an individual *ReactionRecord*, R_2, retained in Reaction Memory; the green region groups $[R_1, R_2, R_3]$ into a Route Memory entry through their participating *Compounds*. Each *ReactionRecord* links to its required reactants through CONSUMES and to its product through PRODUCES; *Compounds* retain their HAS-FRAGMENT relations for substructure retrieval. The surrounding records R_0 and R_4 illustrate how stored memory remains connected to other reaction precedents in the same graph.

The memory agent extracts key reactions from validated single-step outcomes and complete reusable routes or subroutes from solved search results. It produces structured reaction and route records; the executor stores these records, performs lookup, and checks compatibility at reuse. During training, validated single-step outcomes populate Reaction Memory as *ReactionRecords* distinguished from ORD records by their source attribute. These records supply reaction precedents for candidate generation, revision, and reference scoring. For Route Memory, extraction preserves every reaction and required precursor branch of the completed route, together with its stock leaves and costs.

Reaction Memory is fixed before evaluation and participates in retrieval and reference scoring. Route Memory is initialized empty for each target, permitting reuse between branches of that target's search. Only stock-complete subroutes whose reactions pass the configured checks are inserted. Entries retain their canonical root, *ReactionRecord* sequence, participating *Compounds*, terminal stock molecules, and constituent reaction costs.

To attach an entry, the planner checks its stock compatibility, its height against the remaining depth, and all of its internal paths against the current ancestors. Among compatible entries, the entry with the smallest summed reaction cost is used; canonical route order breaks ties. The continuation is attached with its full reaction structure and cost. The planner marks its terminal compounds solved and propagates this state through the attached reactions and their ancestors. An AND node is solved

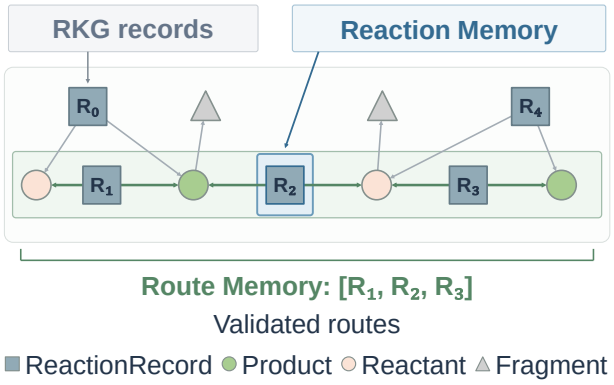

Figure 5: Reaction Memory and Route Memory within the RKG. Squares denote *ReactionRecords*, circles denote participating *Compounds* with product or reactant roles, and triangles denote *Fragments*. The blue highlight marks a single-step memory record; the green region and bracket identify the connected reaction records of a validated synthetic route. Shared *Compounds* connect successive reactions and preserve access to surrounding reaction and *Fragment* evidence.

when every precursor is solved; an OR node is solved when it is in stock or has a solved reaction alternative. Newly completed subroutes are inserted into Route Memory after each such update.

Single-step outputs are cached by canonical target molecule under the fixed model, retrieval, and evaluation configuration. The cache stores the final candidate structures, evaluations, and reaction costs. Cached candidates are checked against the depth and ancestors at each occurrence. When a memory entry becomes available after a state was queued, the planner attaches it upon selecting the affected leaf and reinserts the state with its updated cost.

The algorithm uses an expansion limit $E_{\max}$, counting one expansion for each new invocation of the complete single-step module. Cache lookup, memory attachment, and tree-state updates reuse existing results. After $E_{\max}$ expansions, queued states continue using cached candidates and completed subroutes. A route is reported only after every terminal molecule belongs to stock. The search returns the requested number of distinct complete routes, or the complete routes collected when the queue is exhausted, ordered by their accumulated costs.

B.4 Search Algorithm

For a partial route R, $\mathcal{L}(R)$ contains its unresolved leaf occurrences, $g(R) = \sum_{r \in R} c(r)$, and $F(R) = g(R) + |\mathcal{L}(R)|$. Algorithm 1 initializes the priority queue $\mathcal{Q}$, canonical-molecule candidate cache $\mathcal{C}$, Route Memory $\mathcal{M}$, and complete-route set $\mathcal{S}$ (line 1). The search selects the partial route with the smallest F and records it when all leaves are solved (lines 3–5). After selecting an unresolved leaf, the planner attaches the lowest-cost compatible memory subroute, propagates solved states, and reinserts the route (lines 7–11). Compatibility accounts for the stock, remaining depth, and ancestors at the selected leaf.

For an uncached molecule, the planner checks the expansion limit $E_{\max}$ before invoking the **Agentic Single-step Expansion** and **Evaluation** modules (encapsulated as EVALUATE(SINGLESTEP)) to cache new candidates (lines 13–17). Each new invocation of this agentic pipeline increments the expansion counter. If $E_{\max}$ is exceeded, uncached branches are pruned (line 15), naturally forcing the remaining search to rely solely on cached candidates and stored memory subroutes. For each cached candidate satisfying the depth and ancestor-cycle constraints, the planner creates a successor partial route (lines 19–20). RESOLVE handles stock leaves and compatible memory subroutes; UPDATE propagates solved states and stores newly completed subroutes (lines 21–22). PUSH deduplicates successor routes and orders them by F (line 23). The search stops when the queue is empty or J complete routes have been collected, returning those routes ordered by their true cost g (lines 2, 26).

Algorithm 1 EnergyRetro constraint-guided AND-OR search

Require: Target m_0, stock $\mathcal{B}$, RKG $\mathcal{G}$, Reaction Memory $\mathcal{M}_{\mathrm{rxn}}$, K, $d_{\max}$, $E_{\max}$, J
Ensure: Up to J complete synthetic routes, ordered by g

$R_0 \leftarrow \mathrm{Init}(m_0, \mathcal{B})$; $\mathcal{Q} \leftarrow \{R_0\}$; $\mathcal{C}, \mathcal{M}, \mathcal{S} \leftarrow \varnothing$; $i \leftarrow 0$
while $\mathcal{Q} \neq \varnothing$ **and** $|\mathcal{S}| < J$ **do**
  $R \leftarrow \mathrm{PopMin}_F(\mathcal{Q})$
  if $\mathcal{L}(R) = \varnothing$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{R\}$; **continue**
  end if
  $m \leftarrow \mathrm{SelectLeaf}(\mathcal{L}(R))$
  if $\mathcal{M}(m \mid R) \neq \varnothing$ **then**
    $\rho^* \leftarrow \arg\min_{\rho \in \mathcal{M}(m|R)} g(\rho)$
    $R \leftarrow \mathrm{Attach}(R, m, \rho^*)$; $(R, \mathcal{M}) \leftarrow \mathrm{Update}(R, \mathcal{M})$
    $\mathrm{Push}_F(\mathcal{Q}, R)$; **continue**
  end if
  if $m \notin \mathrm{dom}(\mathcal{C})$ **then**
    if $i \geq E_{\max}$ **then**
      continue {Prune if limit reached and no cache available}
    end if
    $\mathcal{C}[m] \leftarrow \mathrm{Evaluate}(\mathrm{SingleStep}(m, \mathcal{G}, \mathcal{M}_{\mathrm{rxn}}, K))$; $i \leftarrow i + 1$
  end if
  for each $r \in \mathcal{C}[m]$ with $\mathrm{Valid}(r, R, d_{\max})$ **do**
    $R' \leftarrow \mathrm{Attach}(R, m, r)$
    $R' \leftarrow \mathrm{Resolve}(R', \mathcal{B}, \mathcal{M})$
    $(R', \mathcal{M}) \leftarrow \mathrm{Update}(R', \mathcal{M})$
    $\mathrm{Push}_F(\mathcal{Q}, R')$
  end for
end while
return $\mathrm{Sort}_g(\mathcal{S})$

B.5 Baseline Models

We compared EnergyRetro with search-based and LLM-based methods on the corresponding benchmarks. Search-based baselines included Greedy DFS, Retro*, EG-MCTS (Hong et al., 2023), and PDVN (Liu et al., 2023), with MCTS and DFPN additionally included on PaRoutes. LLM-based baselines comprised AOT* (Song et al., 2026), Retro-R1, RetroAgent (Zhu et al., 2026), and LLM-Syn-Planner (Wang et al., 2025). On USPTO-190, LLM(MCTS) uses the AiZynthFinder search framework (Genheden et al., 2020; Thakkar et al., 2020), with its single-step retrosynthesis predictor replaced by an LLM. LLM(Retro*) uses the Retro* search framework (Chen et al., 2020) with the same single-step predictor replacement.

B.6 Evaluation Metrics

Solve rate measures the percentage of targets for which planning returns a route whose terminal molecules all belong to the building-block stock. We report this metric as Success Rate on ChEMBL-1000 at each specified planning budget. On PaRoutes, R@1 and R@10 measure exact reference-route recovery among the first one and ten returned routes, respectively. PaRoutes benchmark resources are available at `https://github.com/MolecularAI/PaRoutes`. A *single-step Agent expansion* invokes the complete workflow for one frontier molecule and increments the expansion counter in Algorithm 1. For USPTO-190 and ChEMBL-1000, N denotes the planning budget measured in each method's own planning unit. These units follow the method's planning procedure, which may expand an individual molecule or generate a complete synthetic route. The separate budget N_{LLM} limits external LLM requests. LLM calls count external model requests for Solver proposals, Critic feedback, rewriting, selection, retries, and tool-use rounds.

C BDE DISTRIBUTIONS ON PAROUTES

Figure 6 presents retained-skeleton BDE distributions for PaRoutes n1 and n5 separately. The reaction-level $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ distributions contain 21,438 and 26,167 records, with medians of -87.9 and -88.3 kcal mol^{-1}, respectively. Each route-level distribution contains 10,000 routes and shows $\sum \Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ over the route's steps; the corresponding medians are -175.9 and -214.1 kcal mol^{-1}. The supplied histogram bins have widths of 25 kcal mol^{-1} for reactions and 50 kcal mol^{-1} for route sums, and include the full observed ranges. The USPTO-190 route panel in the main text reports $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{route}}$, the average over steps with retained-skeleton edits.

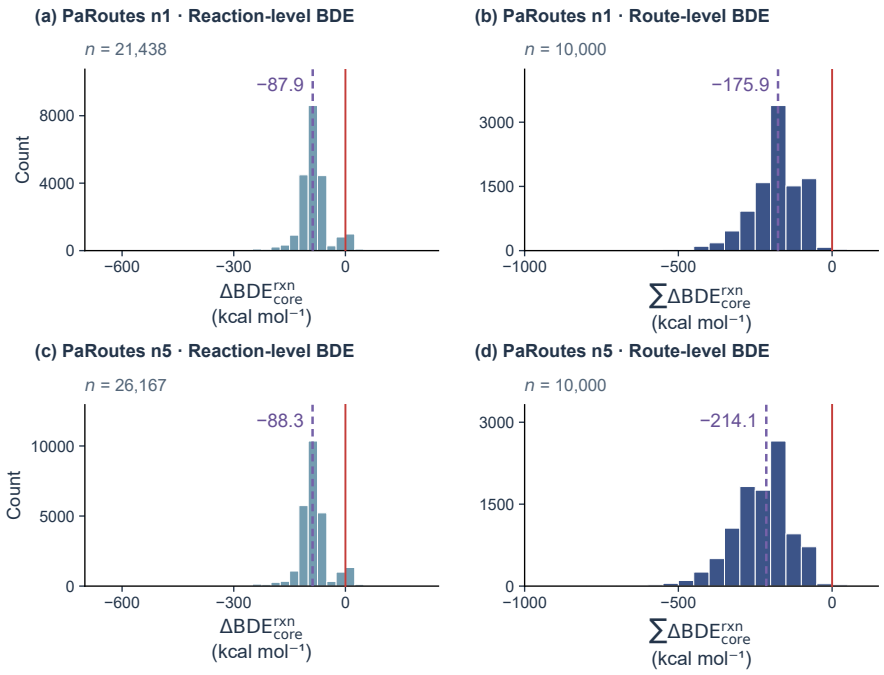

Figure 6: Retained-skeleton BDE distributions on PaRoutes. **(a,b)** Reaction-level $\Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ and route sums $\sum \Delta\mathrm{BDE}_{\mathrm{core}}^{\mathrm{rxn}}$ for n1. **(c,d)** The corresponding distributions for n5. Counts are shown separately for each dataset and level. BDE differences use the forward-synthesis direction. Purple dashed lines mark medians from the source summary statistics; red lines mark zero.