RXNHAYSTACK: BEYOND NEEDLES IN LONG-CONTEXT STRUCTURED SCIENTIFIC DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) deployed in AI-for-science increasingly confront structured datasets that far exceed their context windows. Existing long-context benchmarks, however, rarely distinguish failures of dataset-scale access, systematic execution over structured objects, and domain-specific reasoning. We introduce RxnHaystack, a diagnostic benchmark over 122,456 cleaned USPTO reactions. Its 100 questions span exact retrieval, deterministic aggregation, reaction-level inference, and relational reasoning over synthesis graphs. We evaluate plain prompting, CodeAct, and Recursive Language Models (RLMs) across 21,500 question-level trajectories. Plain-prompting performance declines in every tier as the supplied context grows from 100 to 500 reactions. CodeAct remains stable on Tiers 1–2 but declines on the harder Tiers 3–4 as contexts extend to 1,000 reactions. Both interfaces are bounded by the model context window, preventing them from receiving very large corpora in full. Recently proposed RLMs address this access constraint through recursive processing beyond the prompt window. Their full-corpus setting exposes agents to all 122,456 reactions, enabling more demanding corpus-wide questions and revealing which challenges remain once the complete dataset is accessible. Across Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5, mechanically specified graph operations average 0.81 macro-F1, while mechanism classification averages 0.34 and route and multi-constraint tasks average 0.08. In controlled tests, chemistry-intensive performance falls as more reactions must be searched, even when every corpus contains one correct reaction. Providing the correct chemistry rule, but not the answers, improves classification, yet models still fail to apply it consistently across the entire corpus. Together with deterministic ground truth and inspectable traces, these controls turn aggregate performance into a diagnosis of why an agent fails, providing a reproducible testbed for agents that must operate systematically over large structured scientific data.

1 INTRODUCTION

Frontier LLMs now support context windows of a million tokens or more (Comanici et al., 2025), yet substantial reliability gaps remain. Retrieval accuracy falls when evidence appears in the middle of a long document (Liu et al., 2024); attention requires special handling over long sequences (Xiao et al., 2024); and models struggle to combine information spread across a large context even when each fact is individually accessible (Hsieh et al., 2024).

Scientific work introduces an additional challenge: many questions require a systematic operation over structured records. A biologist may compare genetic perturbations across cell states and experimental conditions (Dixit et al., 2016; Peidli et al., 2024); a materials scientist may filter structures by several composition and property constraints (Jain et al., 2013); and an adaptive experiment may aggregate a time series before selecting the next measurement (King et al., 2004; Kuroki et al., 2026). These analyses involve filtering, joining, aggregating, or traversing records, together with a domain rule that determines which records match or how they are linked. A textual encoding makes the records readable by an LLM, but reliable analysis still depends on executing the operation systematically and choosing a scientifically valid rule.

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.

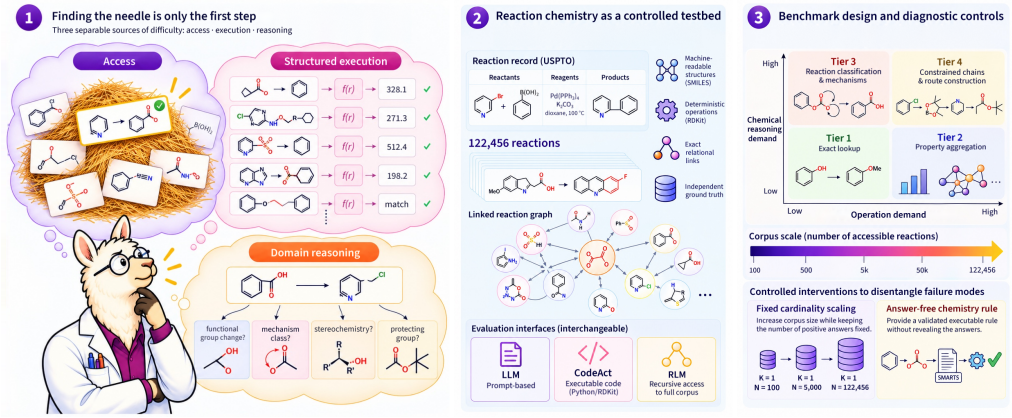

Figure 1: Overview of RxnHaystack. (1) Reasoning over large structured datasets separates access to relevant records, consistent execution, and domain reasoning. (2) Reaction chemistry provides 122,456 machine-readable records, deterministic operations, exact links, and independently computed answers; plain prompting, CodeAct, and RLM provide different evaluation interfaces. (3) The 100 questions span operation and chemistry-reasoning demand across corpus scale. Controls increase scale while retaining one correct reaction or supply an answer-free chemistry rule, distinguishing access, rule formulation, and execution.

A benchmark for this setting requires four ingredients: a corpus exceeding ordinary prompt budgets, exact relations between records, tasks that vary independently in operation and domain demands, and reference answers that can be computed and checked independently of the evaluated model. Chemical reaction corpora provide a controlled setting that meets all four requirements. Reaction SMILES (Weininger, 1988) encode reactants, reagents, and products as machine-readable molecular graphs; canonical molecular identity links products to downstream reactants, forming a synthesis graph; and cheminformatics tools can verify molecular properties, bond changes, reaction patterns, and graph relations. The USPTO 2023 corpus supplies realistic scale and chemical diversity (van der Lingen, 2023). Within this shared representation, we can vary dataset access, structured operation, and scientific reasoning while retaining independently computed answers. The tasks reflect practical reaction-data operations, including corpus filtering, reaction classification, and searches over synthesis graphs. They evaluate useful computational steps, not whether a synthesis would work in the laboratory.

We introduce RXNHAYSTACK, a benchmark that characterizes each question along two dimensions while varying dataset scale independently. *Operation demand* captures the structured computation needed to scan, aggregate, link, or traverse the records. *Chemistry-reasoning demand* captures the scientific knowledge and inference needed to formulate the relevant structural, mechanistic, or route-level rule. The benchmark tiers cover different combinations of these demands. For example, molecular-weight filtering and mechanism classification both require a corpus-wide scan, while the latter also requires a chemical account of the transformation. Graph traversal creates the opposite contrast: it can impose substantial operation demand even when every linking rule is stated explicitly.

Our contributions are:

- We construct 100 questions that vary dataset scale, operation demand, and chemistry-reasoning demand within one corpus, progressing from retrieval-dominated tasks to classification and multi-step relational reasoning. Section 3 defines these dimensions and the deterministic evaluation.
- We compare three common ways of letting language models work with large structured datasets: plain prompting, executable CodeAct (Wang et al., 2024), and recursive access through RLMs (Zhang et al., 2025). This comparison shows how the way models access and process the data changes performance across models, dataset scales, and task demands (Section 4).

- We increase corpus size while keeping one correct reaction, and separately provide the correct chemistry rule without revealing answers. These controls distinguish the effects of corpus scale, rule formulation, and consistent execution (Section 4.3).
- We connect aggregate scores to trace-level failure mechanisms, including incomplete data access, inconsistent execution, invalid domain rules, broken relational operations, and failed multi-step reasoning (Section 4.2 and Appendix C).
- We preserve task generators, scorers, execution traces, resource measurements, and reconstruction instructions so that every reported result can be audited and extended (Appendix E).

2 Related Work

Long-context evaluation. SCROLLS (Shaham et al., 2022), LongBench (Bai et al., 2025), and L-Eval (An et al., 2024) assemble long-document understanding tasks, while RULER (Hsieh et al., 2024) and InfiniteBench (Zhang et al., 2024) stress retrieval, aggregation, and multi-step inference at larger scales. OOLONG focuses specifically on classifying and aggregating information distributed throughout a long context (Bertsch et al., 2025). These benchmarks establish that nominal context length does not guarantee reliable use of distant information. Their inputs are predominantly natural-language sequences. RxnHaystack complements them with a scientific setting built around structured data, where success requires explicit operations over molecular records and relations.

Agents for data analysis and scientific discovery. ScienceAgentBench and DiscoveryBench evaluate executable workflows for data-driven scientific analysis and discovery (Chen et al., 2025; Majumder et al., 2025), while DSBench measures end-to-end data analysis and modeling on realistic datasets (Jing et al., 2025). BIRD studies semantic parsing and efficient execution over large relational databases (Li et al., 2023). These evaluations establish the practical importance of agents that can inspect data, write analyses, and execute structured queries. RxnHaystack addresses a complementary measurement question: it holds the corpus, scientific questions, and deterministic scoring rules fixed while varying dataset scale, structured operation, and chemistry reasoning. The resulting contrasts attribute performance changes to specific demands within a shared scientific environment instead of only measuring end-to-end task completion.

Molecular and reaction benchmarks. MoleculeNet evaluates molecular property prediction (Wu et al., 2018); USPTO-derived tasks support reaction-outcome prediction and retrosynthetic planning (Schwaller et al., 2019; Segler et al., 2018); and other benchmarks measure reaction yield (Ahneman et al., 2018) or broad chemistry knowledge in language models (Mirza et al., 2025). These resources have driven substantial progress on molecular representations and chemical prediction. Their standard instances provide a molecule, reaction, or short problem as the input. RxnHaystack instead asks a model to operate over a shared reaction corpus, making dataset access, exhaustive execution, and chemistry reasoning visible within the same evaluation.

Retrieval, tools, and recursive access. Retrieval-augmented generation (RAG) selects a compact evidence set before generation (Lewis et al., 2020); reaction fingerprints such as DRFP provide a domain-specific similarity representation for chemical retrieval (Probst et al., 2022). ReAct interleaves language-model reasoning with tool use (Yao et al., 2023), and CodeAct uses executable programs as the action space (Wang et al., 2024). RLMs expose an oversized context through recursive sub-calls (Zhang et al., 2025). These approaches offer different mechanisms for selecting, accessing, and processing external information. RxnHaystack holds the scientific questions and scoring rules fixed while comparing representative interfaces for corpus access and structured execution.

3 Benchmark

3.1 Dataset

All tasks are grounded in the USPTO 2023 dataset (van der Lingen, 2023) of 137,261 organic reactions from US patent filings. A cleaning pipeline discards malformed lines, rejects reactions with molecules that fail RDKit sanitization (Landrum et al., 2026), and removes duplicates under

Table 1: Benchmark tiers. Q = number of questions; OP/CR give the maximum operation and chemistry-reasoning demand within each coarse tier bundle. Dataset scale is varied independently through context construction.

Tier	Q	OP	CR	Representative tasks
1 Structural lookup	10	low	low	Find all reactions producing a given canonical SMILES.
2 Property aggregation	20	high	low	Filter by molecular-weight change, ring-count change, aromatic ring formation, and their conjunctions.
3 Reaction-level reasoning	35	high	high	Classify by bond changes, scaffolds, reagents, stereochemistry, SMIRKS functional-group templates, and mechanism templates.
4 Relational reasoning	35	high	high	Enumerate synthesis chains, hub molecules, protecting-group pairs, truncated route prefixes, SMIRKS-constrained 2-step chains.

the canonicalized (reactants, reagents, products) triple, leaving 122,456 reactions. Each reaction is one line of the form `index reactants>reagents>products` with dot-separated canonical SMILES fields. The dataset spans amide bond formation, palladium-catalyzed cross-couplings, protecting-group chemistry, ring-forming reactions, and functional-group interconversions, giving realistic chemical diversity.

3.2 Tier Design

We characterize tasks along two dimensions and evaluate them across a separate experimental scale axis. *Operation demand* is the computation a model must coordinate across entries once the task-relevant rule is available. *Chemistry-reasoning demand* is the scientific knowledge and inference needed to formulate the correct structural, mechanistic, or route-level rule. Context size controls dataset scale: the amount of the reaction corpus the model must access and traverse. We analyze these dimensions separately, without imposing a multiplicative model of difficulty. This organization creates direct comparisons among related tasks. Exact lookup stresses exhaustive access; descriptor filters add corpus-wide computation while keeping chemistry reasoning shallow; and reaction classification preserves the scan structure while requiring deeper chemical interpretation. Path queries provide a parallel contrast between purely structural traversal and chemically constrained routes.

The operation dimension also has a computational interpretation. For N reaction records, treating the cost of processing one record as fixed, exhaustive lookup, descriptor filtering, and per-reaction classification require $O(N)$ work. Relational tasks add a graph with V nodes and E links: ordinary traversal takes $O(V + E)$ time, while enumerating every constrained path is output-sensitive and can grow combinatorially with path length. Operation demand therefore records the computation required by a task; a tier need not correspond to one asymptotic class.

The task dimensions apply at subgroup granularity, and tier labels are coarse bundles whose table ratings denote their maximum demand. Table 1 summarizes the design; full definitions appear in Appendix A. The 100 questions are fixed across runs with deterministic RDKit-computed ground truth; randomness enters only through context sampling.

Tier 1 (10 questions) is exact-match retrieval: given a target product SMILES, return all reaction indices whose product side contains it after canonicalization. No chemistry knowledge is required; the challenge is accurate scanning of a large structured corpus.

Tier 2 (20 questions) requires computing a molecular descriptor for every reaction in context and filtering by threshold: molecular-weight increase beyond τ Da, ring-count increase of exactly r, formation of exactly a new aromatic rings, and conjunctions of weight and ring criteria. The chemistry is shallow but the computation is non-trivial at scale.

Tier 3 (35 questions) requires classifying individual reactions by chemical content. This is the work of a trained organic chemist reading a reaction table: recognizing what bonds have formed or broken, which functional groups have been transformed, what the reagents imply about the mechanism, and

whether stereochemistry has been set or preserved. Tasks fall into six sub-categories of increasing depth: bond-level connectivity changes (3Q), fused-heterocycle and ring-construction scaffolds (3Q), reagent classes such as transition-metal catalysts and amide-activating agents (2Q), *de novo* stereocenter creation and *E*-alkene identification (2Q), functional-group transformations defined by Rxn-INSIGHT SMIRKS templates (Dobbelaere et al., 2024) covering amide couplings, protecting-group manipulations, and named cross-couplings (15Q), and mechanism-level templates from a mechanistic reaction dataset (Joung, 2025) covering pathways such as Wittig olefination, Mitsunobu, and Staudinger reduction (10Q).

Tier 4 (35 questions) shifts to reasoning over the network of reactions, where chains are linked by exact canonical-SMILES equality between a product of step i and a reactant of step $i + 1$. This mirrors the logic of multi-step synthesis planning, where intermediates flow from one reaction to the next, and requires the model to hold a graph structure in mind across all reactions in context, which is qualitatively different from any per-reaction classification. Sub-groups: mechanical graph tasks covering fixed-length and longest-chain enumeration and hub-molecule identification (5Q); chemically constrained chains covering functional-group transformation chains, protecting-group install/remove pairs, and ring-construction chains (10Q); and two substantially harder groups (20Q): *truncated synthesis routes* (T4.16–T4.25; Appendix A), where the final step of a 5-reaction route to a named target is withheld, requiring the model to identify prefixes compatible with an unseen terminal step; and *SMIRKS-constrained 2-step chains*, requiring two independent reaction-template matches plus a cross-chain chemical-persistence condition simultaneously. The truncated-route questions identify each target by name without providing its structure, so they jointly test name-to-structure interpretation and route-prefix recovery.

3.3 Evaluation Setup

Ground truth construction. For each question, let G denote the complete set of correct answers over the $N = 122{,}456$ reactions, and let $|G|$ denote the number of answers. These are reaction indices for Tiers 1–3 and task-specific chains, chain indices, reaction pairs, or molecules for Tier 4. All ground-truth sets are precomputed offline using RDKit chemistry rules and exact canonical-SMILES rules for linking reactions into chains, then stored with each task, so evaluation is deterministic and independent of any inference-time chemistry tooling.

Context construction. Each question is presented with a context of x reaction lines. For index-return tasks, we deterministically select correct reactions in proportion to $|G|x/N$, with a task-specific minimum that prevents contexts with no positive example. For relational tasks, we select complete answers under the same proportional budget and include every reaction needed to form them. Random non-answer reactions fill the remaining positions. The selected answers form the context-specific ground-truth set $G_c \subseteq G$.

Every answer used for scoring is therefore already present in the context, so the candidate-selection step misses none of the scored positives. A chemical retriever such as DRFP (Probst et al., 2022) could change the distractors or supporting records, but a RAG system could not place more scored positives into the same context because none are missing from G_c. We therefore omit a separate RAG baseline and use the sampled contexts to measure what happens after retrieval: recognizing the relevant records, processing them correctly, and reporting the complete answer set.

Answer format and metrics. Models return all satisfying task-specific items, or -1 if none. Each evaluated context contains at least one scored answer, so G_c is never empty. We score each question as set retrieval: precision, recall, and F1 of the predicted set against G_c, assigning zero to an empty or unparsable prediction, plus exact-set-match as a binary indicator. We compute each metric separately for every question and then average the question scores, giving every question equal weight regardless of how many answers it contains (macro-averaging).

Human baseline. We collected a tool-assisted human baseline from three chemistry-trained volunteers. Since several questions require substantial corpus-wide computation even with tools, each volunteer received a stratified subset spanning all tiers and task families, with selected overlap for agreement. Together they submitted 52 responses covering 43 unique questions. We score the 51 responses that contain an answer; one response marked "cannot answer" is retained for completion

and timing summaries but excluded from accuracy. Language models or coding agents were used in 45 responses. Annotators could also use RDKit, scripts, notebooks, literature or web search, offline processing, and consultation, but remained responsible for checking the method and final answer. Full study details appear in Appendix D.

Inference paradigms. Three paradigms receive identical sampled corpora and questions. *Plain LLM*: the supplied corpus and question in one model call without tool or code execution. *CodeAct* (Wang et al., 2024): an agent that iteratively writes and executes Python (with RDKit) over the supplied reaction lines. *RLM* (Zhang et al., 2025): an agent that can inspect the supplied corpus programmatically and issue recursive sub-calls. For all three interfaces, $x \in \{100, 500\}$ is the number of supplied reaction lines; the additional RLM "full" condition exposes all 122,456 reactions. No provider-side code-execution tool was enabled for plain prompting. Plain-LLM and CodeAct requests used low reasoning effort in Tier 1 and high reasoning effort in Tiers 2–4 where the endpoint exposed that control. RLM requests used low reasoning effort through OpenRouter. Where a model exposed a separate hidden-thinking channel, we disabled it while retaining the visible agent loop. RLM supplies a full-corpus access mechanism, but access alone does not ensure exhaustive or chemically valid processing. These paradigms probe different access and execution mechanisms; the benchmark tasks do not depend on any one of them. We evaluate five models spanning open- and closed-weight families: DeepSeek V4 Flash (DeepSeek-AI, 2026); Qwen3.5-397B-A17B (Qwen Team, 2026); Google Gemini 3.7 Flash (Google, 2026); OpenAI GPT-5 mini (OpenAI, 2025); and Anthropic Claude Haiku 4.5 (Anthropic, 2025).

Each model–interface–context–task configuration is evaluated in five repetitions, with context-sampling variation as described above. Per-model error bars report one population standard deviation across those repetitions. Cross-model figures instead report the standard error across model-level means and state the number of contributing models in their captions.

4 RESULTS

Having separated dataset scale, operation demand, and chemistry reasoning in the benchmark design, we now examine how the three interfaces behave as more of the reaction corpus becomes available. The $x = 100$ and $x = 500$ conditions cover all five models, while the full-corpus condition exposes all 122,456 reactions through RLM. At $x = 1000$, the CodeAct and RLM extensions both cover Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, and GPT-5 mini. These selected extensions limit the added inference cost. Figure 2 shows the resulting tier means. At $x = 1000$, CodeAct remains strong on structural lookup and property aggregation (0.991 and 0.848) but falls to 0.470 on reaction-level reasoning and 0.394 on relational reasoning. RLM shows the same broad separation, with scores of 0.883, 0.906, 0.630, and 0.612 across the four tiers. Thus, executable interfaces sustain systematic lookup and descriptor computation as the context grows, while chemistry-intensive classification and relational tasks remain sensitive to scale. Full-corpus access extends the range of questions an agent can attempt, but it does not remove the reasoning and execution errors exposed by the harder tiers.

Tier 1: Structural lookup. Exact lookup mainly tests whether an interface can search the supplied records systematically. CodeAct is effectively insensitive to the increase from $x = 100$ to 500, moving from 0.988 to 0.990, and reaches 0.991 in the selected $x = 1000$ extension. RLM follows the same pattern and still scores 0.900 when given access to the full corpus. Plain prompting begins comparably well at 0.954 for $x = 100$ but falls to 0.771 at 500. Once lookup is implemented as a systematic operation, increasing the amount of data adds little difficulty; the larger loss under prompting comes from using the supplied records less reliably.

Tier 2: Property aggregation. Property aggregation makes the value of executable access especially clear. Every reaction must be processed with the same descriptor calculation, a task that plain prompting performs poorly as a whole: its mean falls from 0.227 at $x = 100$ to 0.147 at 500. CodeAct instead scores 0.764 and 0.840, while RLM remains between 0.906 and 0.910 from $x = 100$ through the full corpus. Gemini 3.7 Flash is an important exception, reaching 0.960 under plain prompting at $x = 100$ without provider-side code execution; the other four models score no higher

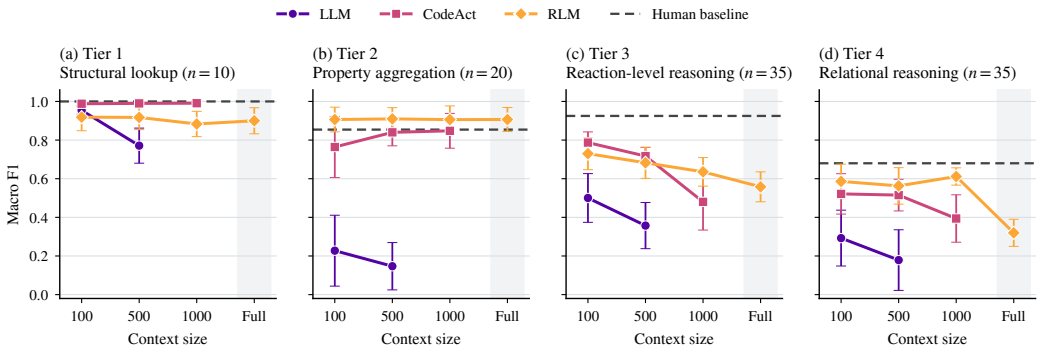

Figure 2: Cross-model scaling by tier. Points show mean macro-F1, and error bars show standard errors across model-level means. The $x = 100$ and $x = 500$ points use five models. At $x = 1000$, CodeAct and RLM use Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, and GPT-5 mini. Full-corpus RLM uses five models. Within each model, tier scores average questions, and unsuccessful trajectories score zero. Dashed horizontal lines show the tool-assisted human macro-F1 on the assigned question subsets. The shaded full-corpus setting is RLM-only.

than 0.080 in that condition. The general separation nevertheless shows that corpus-wide descriptor questions become tractable when models can express the calculation once and apply it consistently.

Tier 3: Reaction-level reasoning. Reaction-level questions change the picture because the model must first decide which chemical rule captures the query. Executable access still helps, but no longer makes performance stable as more reactions are exposed. CodeAct declines from 0.766 at $x = 100$ to 0.713 at 500 and reaches 0.470 in the selected $x = 1000$ extension. RLM declines from 0.712 to 0.679, reaches 0.630 at $x = 1000$, and scores 0.554 on the full corpus. Plain prompting follows the same direction, from 0.500 to 0.357. The model-specific task heatmaps (Figures 18, 21, 24, 27, and 30) expose substantial variation across chemical operations beneath these averages. A larger corpus also contains more correct answers, so the scaling curve alone cannot tell whether the loss comes from data access, an imprecise chemical rule, or incomplete execution. Section 4.3 separates these effects with controlled comparisons.

Tier 4: Relational reasoning. Relational reasoning reveals two very different regimes within a single tier. At $x = 100$ and 500, aggregate CodeAct performance is nearly unchanged (0.521 and 0.515), and RLM changes only modestly from 0.586 to 0.563. The selected $x = 1000$ RLM extension reaches 0.612, while full-corpus RLM scores 0.311. More importantly, the full-corpus average conceals a large internal divide: graph operations with an explicit linking rule reach 0.806, while route recovery and multi-constraint tasks reach only 0.076. The harder group does not simply require a larger answer set. It contains 30.7 scored items on average, compared with 979.6 for the chemically constrained graph questions that models answer much more successfully (Appendix A.5). None of the 225 full-corpus Tier 4 runs hit the 30 GiB memory limit. The low route scores therefore persist even when answer-set size and memory exhaustion are ruled out as sufficient explanations, pointing instead to the difficulty of preserving chemical and relational constraints across several steps. The cross-model comparison in Section 4.1 and the trace analysis in Section 4.2 examine this separation directly. In a separate three-target control, providing the target structure together with the broad final-transformation class raises route F1 from zero to 0.378 for Qwen3.5-397B-A17B and 0.133 for Claude Haiku 4.5, yet recovery remains far below that of explicitly linked graph tasks (Appendix A.8).

Human baseline. The tool-assisted human baseline calibrates difficulty on 51 responses covering 43 questions. Volunteers obtain 0.842 macro-F1 and 0.686 exact-set-match accuracy, with tier-level F1 scores of 1.000, 0.854, 0.925, and 0.680. Their largest loss occurs on the same relational tier that challenges the models. Annotators could use domain tools but remained responsible for checking their method and final answer; study design, tool use, timing, and agreement appear in Appendix D.

What does additional inference buy? RLM's additional inference primarily expands corpus reach: at matched context sizes its Tier-4 accuracy gain over CodeAct is modest, while it alone evaluates all 122,456 reactions. The corresponding accuracy–cost trade-offs and model-level resource profiles appear in Figures 13 and 14.

4.1 Which task demands remain difficult across models?

Tier averages mix several sources of difficulty. Table 2 therefore regroups the full-corpus RLM questions by what a model must do after it can access the reaction records. The comparison spans Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5, with unsuccessful trajectories scored as zero. Its purpose is to test whether task demand explains a common performance pattern across model families.

Two contrasts make that pattern clear. Within Tier 3, detecting bond and stereochemical changes reaches mean F1 scores of 0.852 and 0.828, while mechanism classification falls to 0.336. All three operate on individual reaction records, but mechanism classification requires the model to infer a deeper chemical account of the transformation. Tier 4 shows a corresponding progression over relations. When the linking rule is explicit, graph operations average 0.806. Adding chemical constraints lowers the mean to 0.532, and tasks that combine route recovery with several constraints fall to 0.076.

The model ranges in Table 2 show substantial variation in absolute performance. The ordering is nevertheless stable within each model: all five score higher on explicit-link graph operations than on mechanism classification or route and multi-constraint tasks. Across model families, successful corpus access and traversal of an explicitly defined graph therefore coexist with much lower performance when the model must infer the relevant chemistry or preserve it across several steps.

Table 2: Task-demand hierarchy under full-corpus RLM. Values summarize Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5. Each model's group score averages the constituent questions; the table reports the mean and range of those five scores.

Task group	What success requires	Mean F1 (model range)
Bond changes	Detect specified bond edits	0.852 (0.611–1.000)
Stereochemistry	Interpret specified stereochemical features	0.828 (0.588–1.000)
Mechanism classification	Infer a mechanistic class	0.336 (0.207–0.563)
Explicit-link graph operations	Execute a stated molecular-linking rule	0.806 (0.395–1.000)
Chemically constrained graph operations	Infer and enforce chemical constraints during traversal	0.532 (0.293–0.840)
Route and multi-constraint tasks	Coordinate target, graph, and multi-step constraints	0.076 (0.020–0.250)

4.2 What do failed trajectories reveal?

Recorded executions show how similar scores can arise from distinct failures. We assign a mechanism only when generated code, tool observations, strategy changes, or the returned answer provide direct evidence. Some agents never run the intended operation because a tool action, parser, or chemistry routine fails; others execute successfully but replace a structural rule with literal reagent names or narrow SMARTS. Full-corpus traces reveal later failures when recursive results are not preserved, common reagents create spurious graph edges, or route rules omit valid chains and generate invalid ones. These observations separate failures of execution, chemistry reasoning, relational operations, and route reasoning, while keeping infrastructure errors distinct. Appendix C quantifies the outcomes, maps the mechanisms to benchmark capabilities, and presents representative traces.

4.3 Is the bottleneck scale, chemistry reasoning, or execution?

The Tier-3 scaling curve in Figure 2c could fall simply because larger contexts contain more correct answers. We test this with RLM by holding the number of correct reactions at one while growing the corpus. GPT-5 mini remains at 0.961–1.000 on Tier 2 but falls from 0.671 to 0.212 on Tier 3; Qwen3.5-397B-A17B changes from 0.965 to 0.902 on Tier 2 and from 0.630 to 0.187 on Tier 3

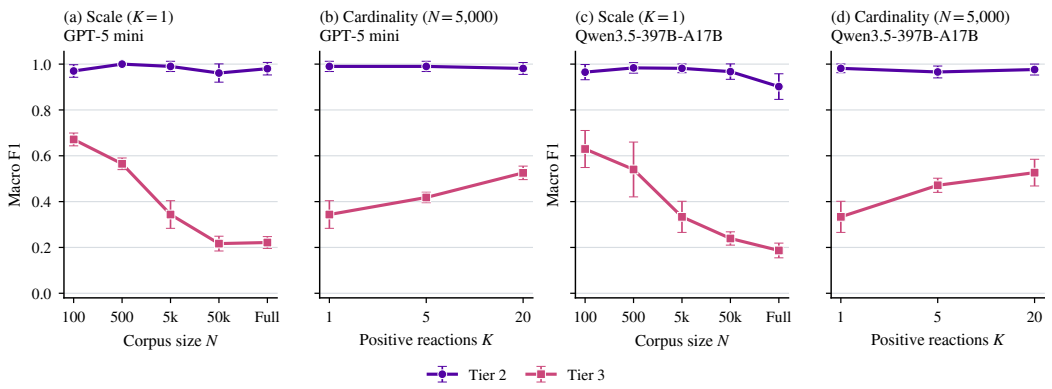

Figure 3: Matched-cardinality controls for GPT-5 mini (a–b) and Qwen3.5-397B-A17B (c–d). Scale panels increase corpus size N at $K = 1$; cardinality panels increase K at $N = 5{,}000$. Points are question-weighted means over five runs with one run-level standard deviation; unsuccessful runs score zero.

(Figure 3a,c). At fixed corpus size, increasing the number of correct reactions from one to 20 instead raises Tier-3 F1 from 0.337 to 0.525 and from 0.334 to 0.526, while Tier 2 remains at 0.965–0.990. The loss therefore follows exposure to a larger and more varied corpus, not simply the number of answers to return.

Precision and recall show what the larger corpus exposes. From 100 reactions to the full corpus, GPT-5 mini Tier-3 precision falls from 0.634 to 0.213 and recall from 0.800 to 0.684. For Qwen3.5-397B-A17B, precision falls from 0.600 to 0.200 and recall from 0.743 to 0.432 (Table 4). The precision loss is consistent with chemical rules that are broad enough to admit increasingly many near misses, while the recall loss shows that correct reactions are also missed during enumeration. Scale thus reveals both weaknesses. It does not act only through context access, nor does it reduce to answer cardinality. Qwen3.5-397B-A17B's unsuccessful runs are distributed across corpus sizes instead of clustering at the largest condition, so they do not explain the trend (Appendix A.6).

These controls clarify why performance falls as the corpus grows. Both models continue to execute the deterministic Tier-2 operations as the corpus grows, whereas Tier 3 exposes errors in defining and exhaustively applying chemical rules. Varying corpus size, answer count, and chemistry demand separately allows RxnHaystack to distinguish these failure sources instead of attributing every loss at long context to data access.

A complementary full-corpus intervention gives Qwen3.5-397B-A17B and Claude Haiku 4.5 the correct executable chemistry rule without revealing answers. Reaction-class F1 rises from 0.413 to 1.000 and from 0.297 to 0.800; mechanism F1 rises from 0.458 to 0.960 and from 0.332 to 0.720. Deterministic execution reaches 1.000, but graph-task effects remain mixed: a correct rule reduces the formulation burden without ensuring consistent application (Appendix A.7).

5 Conclusion

RxnHaystack shows that full-corpus access does not ensure reliable scientific analysis. Across five models, explicitly defined graph operations substantially outperform mechanism classification and route construction. Controlled scaling, answer-free chemistry rules, and execution traces expose failures in both rule formulation and systematic application: the same model can traverse a corpus on one task yet fail to define or execute the operation required by another.

Chemical reaction corpora make this decomposition measurable through compact structural representations, deterministic cheminformatics operations, and exact links between records. Other scientific environments raise the same design question when agents must traverse structured observations, apply systematic operations, and infer the domain rule defining a valid result. Whether these failures transfer remains empirical. RxnHaystack establishes the framework in reaction data, where the factors can be controlled and answers checked exactly.

AI USE STATEMENT

Generative AI tools were used to assist with software development, experiment execution and debugging, aggregation and visualization of experimental results, and language editing. The authors reviewed AI-assisted code, analyses, figures, and text; checked cited sources, tested software changes, and verified numerical claims and figures against the underlying experimental artifacts. The authors take responsibility for the final content of this work, including all claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The anonymous supplement includes all 100 questions and reference answers, answer-generation code, context builders and seeds, scorers, experiment definitions, dataset reconstruction instructions and file fingerprints, and a version-pinned software specification. The code and benchmark will be released publicly upon acceptance. Appendix E describes the release and contamination policy. The appendix also reports model calls, token use, latency, wall time, memory, and API cost in USD by tier and interface.

REFERENCES

Derek T. Ahneman, Jesús G. Estrada, Shishi Lin, Spencer D. Dreher, and Abigail G. Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, 360(6385):186–190, 2018. doi: 10.1126/science.aar5169.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14388–14411, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.776. URL https://aclanthology.org/2024.acl-long.776/.

Anthropic. Claude Haiku 4.5 system card. https://www-cdn.anthropic.com/7aad69bf12627d42234e01ee7c36305dc2f6a970.pdf, 2025. Accessed 2026-09-24.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.183. URL https://aclanthology.org/2025.acl-long.183/.

Amanda Bertsch, Adithya Pratapa, Teruko Mitamura, Graham Neubig, and Matthew R. Gormley. Oolong: Evaluating long context reasoning and aggregation capabilities. *arXiv preprint arXiv:2511.02817*, 2025. doi: 10.48550/arXiv.2511.02817.

Sebastian Bordt, Suraj Srinivas, Valentyn Boreiko, and Ulrike von Luxburg. How much can we forget about data contamination? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4998–5016, 2025. URL https://proceedings.mlr.press/v267/bordt25a.html.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6z4YKr0GK6.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

DeepSeek-AI. DeepSeek-V4-Flash update. https://api-docs.deepseek.com/updates/, 2026. API documentation update dated 2026-07-31; accessed 2026-09-24.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

Maarten R. Dobbelaere, István Lengyel, Christian V. Stevens, and Kevin M. Van Geem. Rxn-INSIGHT: fast chemical reaction analysis using bond-electron matrices. *Journal of Cheminformatics*, 16(1):37, 2024. doi: 10.1186/s13321-024-00834-z.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021. doi: 10.1145/3458723.

Google. Introducing Gemini 3.7 Flash. https://blog.google/innovation-and-ai/models-and-research/gemini-models/introducing-gemini-3-7-flash/, 2026. Accessed 2026-09-24.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. doi: 10.1063/1.4812323.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. DSBench: How far are data science agents from becoming data science experts? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DSsSPr0RZJ.

Joon F. Joung. Mechanistic dataset. https://github.com/jfjoung/Mechanistic_dataset, 2025. GitHub repository, accessed 2026-07-06.

Ross D. King, Kenneth E. Whelan, Ffion M. Jones, Philip G. K. Reiser, Christopher H. Bryant, Stephen H. Muggleton, Douglas B. Kell, and Stephen G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004. doi: 10.1038/nature02236.

Ryo Kuroki, Amin Mansouri, and Philippe Schwaller. Time-series as feedback: Evaluating adaptive reasoning in LLM agents. In *1st ICLR Workshop on Time Series in the Age of Large Models*, 2026. URL https://openreview.net/forum?id=N3vDM8CafN.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, Sereina Riniker, Peter Gedeck, et al. RDKit: Open-source cheminformatics, version 2025.09.6. Zenodo, 2026. URL https://doi.org/10.5281/zenodo.18797641.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing*

Systems, volume 36, 2023. doi: 10.52202/075280-1835. URL https://openreview.net/forum?id=dI4wzAE6uV.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0d70af566e69f1dfb687791ecf955e28-Abstract-Conference.html.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Mehrdad Asgari, Juliane Eberhardt, Amir Mohammad Elahi, Hani M. Elbeheiry, María Victoria Gil, Christina Glaubitz, Maximilian Greiner, Caroline T. Holick, Tim Hoffmann, Abdelrahman Ibrahim, Lea C. Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole C. Roesner, Johanna Schreiber, Ulrich S. Schubert, Leanne M. Stafast, A. D. Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, 2025. doi: 10.1038/s41557-025-01815-x.

OpenAI. GPT-5 Mini model. https://developers.openai.com/api/docs/models/gpt-5-mini, 2025. Accessed 2026-09-24.

Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scPerturb: Harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024. doi: 10.1038/s41592-023-02144-y.

Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint drfp. *Digital Discovery*, 1(2):91–97, 2022. doi: 10.1039/D1DD00006C.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026. Model release and model card; accessed 2026-09-24.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023. doi: 10.18653/v1/2023.findings-emnlp.722.

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. doi: 10.1021/acscentsci.9b00576.

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

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. SCROLLS: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 12007–12021, 2022. doi: 10.18653/v1/2022.emnlp-main.823.

Rik van der Lingen. Reaction SMILES USPTO year 2023, 2023. URL https://figshare.com/articles/dataset/Reaction_SMILES_USPTO_year_2023/24921555.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*,

pp. 50208–50232, 2024. URL https://proceedings.mlr.press/v235/wang24h.html.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018. doi: 10.1039/C7SC02664A.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Alex L. Zhang, Tim Kraska, and Omar Khattab. Recursive language models. *arXiv preprint arXiv:2512.24601*, 2025. doi: 10.48550/arXiv.2512.24601. URL https://arxiv.org/abs/2512.24601.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.814. URL https://aclanthology.org/2024.acl-long.814/.

A Full Task Definitions

This appendix gives the complete task definitions summarized in Table 1.

A.1 Tier 1: Structural Lookup (10 questions)

T1.1 Product Identification (10 questions): Each question names a target product by its canonical SMILES. The model must find all reaction indices in the provided context where that SMILES appears on the product side. Ten target molecules are used, each with multiple matching reactions seeded into context.

A.2 Tier 2: Computed Property Aggregation (20 questions)

T2.1 Molecular Weight Change (6 questions): Find all reactions where the molecular weight of the heaviest product exceeds that of the heaviest reactant by more than τ Da, for $\tau \in \{150, 184, 218, 252, 286, 320\}$. This captures reactions that substantially build up molecular mass, such as fragment couplings and macrocyclizations.

T2.2 Ring Count Change (5 questions): Find all reactions where the maximum ring-count increase across all reactant-product component pairs equals r, for $r \in \{1, 2, 3, 4, 5\}$. This identifies ring-forming reactions like cyclizations, annulations, and ring-closing metatheses by degree of ring construction.

T2.3 Aromatic Ring Formation (5 questions): Find all reactions where exactly a new aromatic rings appear in the products relative to the reactants, for $a \in \{1, 2, 3, 4, 5\}$. This captures aromatization events and reactions that introduce aromatic fragments.

T2.4 Combined MW and Ring Change (4 questions): Find all reactions satisfying both a molecular weight change threshold and a ring-count change simultaneously, for $(\tau, r) \in \{(100, 1), (100, 2), (150, 1), (150, 2)\}$. The conjunction tests the model's ability to apply multiple filters consistently.

A.3 TIER 3: REACTION-LEVEL REASONING (35 QUESTIONS)

Bond-Level Tasks (3 questions). These tasks require tracking atomic connectivity changes between the reactant and product sides of each reaction, without needing to name the reaction class.

T3.1 Reactions that introduce at least one new nitrogen atom: product-side nitrogen count exceeds reactant-side count, as occurs when a nitrogen-containing fragment is incorporated (e.g., nitration, or amine coupling where nitrogen comes from a reagent).

T3.2 Reactions that form at least one C–N bond and break at least one C–O bond simultaneously, as in amide formation from acids or esters and Mitsunobu-type substitution.

T3.3 Reactions that form exactly one C–C bond with no other C–C bond changes: characteristic of selective C–C coupling reactions such as aldol additions or Grignard additions.

Scaffold-Based Tasks (3 questions). These tasks require ring-system recognition in product molecules, distinguishing reactions that construct new rings from those that merely elaborate existing ones.

T3.4 Reactions producing a product containing at least one of the fused heterocycles indole, benzofuran, benzothiazole, or benzimidazole: benzo-fused five-membered heterocycles frequent in drug scaffolds.

T3.5 Reactions that construct a new ring system in the product, identified by the appearance of an SSSR component absent from all reactant molecules.

T3.6 Reactions that build a fused bicyclic or polycyclic ring system from non-fused reactants: the product must contain two or more rings sharing at least one bond while no reactant does.

Reagent-Based Tasks (2 questions). These tasks require inspecting the middle (reagent) field of the reaction SMILES and recognizing specific reagent classes.

T3.7 Reactions whose reagent slot contains a transition metal (d-block elements, groups 3–12, excluding lanthanides and actinides), signaling metal-catalyzed processes such as palladium cross-couplings, ruthenium metatheses, or copper-mediated reactions.

T3.8 Reactions whose reagent slot contains HATU (as the uronium cation with PF_6^-) or T3P: two amide-bond activating agents widely used in peptide and medicinal chemistry synthesis.

Stereochemistry-Based Tasks (2 questions). These tasks inspect explicit tetrahedral and double-bond stereochemistry in the molecular representation.

T3.9 Reactions where the product contains at least one tetrahedral stereocenter with explicit SMILES stereo annotation (`@`/`@@`) and none of the reactants contain any: reactions that create stereocenters *de novo*, such as asymmetric reductions or chiral auxiliary-controlled additions.

T3.10 Reactions producing a product with an *E*-configured C=C double bond (explicit CIP annotation, non-aromatic): typically eliminations, Wittig-type olefinations, or geometrically selective cross-metatheses. The criterion identifies an *E*-alkene in the product; it does not require that the alkene be absent from every reactant.

Functional-Group Tasks (15 questions). These tasks use SMIRKS reaction templates from the Rxn-INSIGHT library (Dobbelaere et al., 2024), so identification requires verifying that the correct bond-forming and bond-breaking events occurred in the right atomic context.

T3.11–T3.14 Amide coupling variants (4 questions): Four mechanistically distinct routes to an amide bond: acyl chloride with primary amine (Schotten–Baumann); carboxylic acid with primary amine (condensation); ester aminolysis with primary amine; ester aminolysis with secondary amine.

T3.15–T3.19 Functional-group transformations (5 questions): Grignard addition to ketone giving tertiary alcohol; nitrile reduction to primary amine; nitro reduction to amine; alcohol conversion to organic azide; primary alcohol oxidation to carboxylic acid.

T3.20–T3.25 Protecting-group and named-reaction transformations (6 questions): Protecting groups (2 questions): Boc deprotection of a secondary amine; Boc protection of a primary amine. Named reactions (4 questions): Suzuki coupling with a boronic acid; Mitsunobu reaction on a sulfonamide nucleophile; Buchwald–Hartwig/Ullmann–Goldberg N-arylation of a primary amine; Stille coupling with an aryl stannane.

Reaction Mechanism Tasks (10 questions). These tasks use mechanism-level templates from a mechanistic reaction dataset (Joung, 2025), requiring that the full bond-making and bond-breaking sequence of a named mechanistic pathway be matched.

T3.26–T3.35 Named mechanistic pathways (10 questions): Ester hydrolysis via hydroxide-promoted acyl substitution; Mitsunobu reaction; Wittig olefination; Knoevenagel condensation/aldol dehydration; azide-alkyne Huisgen [3+2] cycloaddition; keto alpha-alkylation (enolate C-alkylation); base-catalyzed transesterification; Grignard carbonyl addition; Staudinger reduction; alpha-ketone electrophilic bromination.

A.4 Tier 4: Pairwise and Relational Reasoning (35 questions)

Throughout, chains are ordered reaction sequences where a product of step i equals a reactant of step $i + 1$ under exact canonical-SMILES equality.

Mechanical Graph Tasks (5 questions).

T4.1–T4.2 Synthetic chain enumeration (2 questions): Find all ordered synthesis chains of fixed length (3 and 4 reactions) starting from a given reaction index.

T4.3–T4.4 Longest synthesis chain (2 questions): Given a target product SMILES, find the longest chain of reactions with ascending indices that terminates in that product; ascending index is an ordering convention, not a claim about chronology.

T4.5 Hub molecule identification (1 question): Find all molecules that appear as the product of exactly one reaction and as a reactant in at least three distinct higher-index reactions. Index order is a deterministic convention and carries no chronological meaning.

Chemically Constrained Chain Tasks (10 questions).

T4.6–T4.9 Functional-group transformation chains (4 questions): Find all 7-reaction chains where the entry reaction carries a specified source functional group among its reactants and the exit reaction produces a specified target functional group. The four source-to-target pairs are: primary alcohol $\rightarrow$ carboxylic acid; alkyl halide $\rightarrow$ tertiary amine; ester $\rightarrow$ tertiary amide; nitrile $\rightarrow$ primary amide.

T4.10–T4.11 Protecting-group pairs (2 questions): Find all reaction pairs $(r_{\text{install}}, r_{\text{remove}})$ with $r_{\text{install}} < r_{\text{remove}}$ where the same protecting group is installed on a functional group in one reaction and removed in a later reaction on the same canonical molecular scaffold. Two protecting groups are queried: Boc on nitrogen; benzyl on oxygen/nitrogen.

T4.12–T4.15 Ring-construction chains (4 questions): Find all reaction chains of a fixed length where an acyclic precursor is progressively elaborated into a product containing a specified fused ring system: quinoline, indole, benzothiazole, or benzimidazole.

Prospective and Multi-Constraint Reasoning Tasks (20 questions). Both task groups require interacting constraints across multiple reactions and therefore extend beyond a single linear scan with one filter per reaction.

T4.16–T4.25 Truncated synthesis routes (10 questions): A ground-truth 5-reaction synthesis route leading to a named target molecule is mined from the full dataset. At evaluation time, only the first four reactions of each route appear in context; the final step producing the target is withheld. The model must find all valid 4-reaction prefixes of complete 5-step routes to that target. Because the target never appears as a product in context, the model must identify prefixes compatible with an unseen final step. The prompt supplies only the

target name, so performance also depends on name-to-structure interpretation. Ten target molecules span heterocyclic, biaryl, peptidic, spirocyclic, and lactam scaffolds.

T4.26–T4.35 SMIRKS-constrained 2-step synthesis chains (10 questions): Find all ordered 2-reaction chains $[r_0, r_1]$ where step 1 matches a specified SMIRKS template, step 2 matches a second template, the product of r_0 and a reactant of r_1 share the same canonical SMILES, and an additional chemical-persistence condition is satisfied. The 10 reaction-type pairs include Boc protection $\rightarrow$ Buchwald–Hartwig, nitro reduction $\rightarrow$ amide coupling, Suzuki coupling $\rightarrow$ carbonyl reduction, Wittig olefination $\rightarrow$ alkene hydrogenation, alcohol oxidation $\rightarrow$ reductive amination, azide formation $\rightarrow$ Staudinger reduction, and bromination/boronation $\rightarrow$ coupling sequences.

A.5 Full-Corpus Answer-Set Cardinality

Table 3 reports the number of elements in the ground-truth set scored by F1 at full scale. For the longest-chain task, the scored elements are reaction indices in the returned chain; for the other Tier 4 tasks they are complete chains, reaction pairs, or molecules. The route and multi-constraint subgroup has larger sets than the mechanical subgroup, so cardinality could contribute to that pairwise contrast. It nevertheless has only 3.1% as many answers on average as the chemically constrained subgroup, despite a five-model full-corpus RLM mean of 0.076 versus 0.532, ruling out answer-set cardinality as a sufficient explanation of the subgroup ordering.

Table 3: Full-corpus Tier 4 ground-truth cardinality, computed from the stored task constants. Statistics are across questions.

Subgroup	Q	Mean $\|G\|$	Median $\|G\|$	Range
Mechanical graph	5	8.8	8.0	8–10
Chemically constrained	10	979.6	191.5	1–5,378
Truncated-route / multi-constraint	20	30.7	12.0	2–190

A.6 Matched-Cardinality Diagnostics

The completed GPT-5 mini control contains 725 jobs and 2,175 question-level trajectories. The terminal Qwen3.5-397B-A17B replication contains the same 725 jobs: 673 succeeded and 52 terminal failures, which contribute zero. All 52 occur in Tier 3. The fixed-K conditions at $N = 100$, 500, 5,000, 50,000, and full contain 8, 8, 6, 8, and 10 failures, respectively; the two fixed-N cardinality conditions contain six each. Their distribution does not increase monotonically with corpus size. Figure 4 exposes the task-level heterogeneity behind the tier means in Figure 3a–b. For GPT-5 mini, the fixed-K degradation is concentrated in chemistry-intensive Tier-3 families, while deterministic Tier-2 filters remain close to perfect. Figure 5 reports the corresponding calls, input and output tokens, API cost, latency, tool time, wall time, and peak memory.

A.7 Chemistry-Rule Control by Task

Table 5 disaggregates the full-corpus control in the main text. Supplying the executable chemistry rule strongly improves reaction-class and mechanism recognition for both models. The effect varies by operation: Qwen3.5-397B-A17B's stereochemistry score falls from 1.000 to 0.000, and the graph-task effects are mixed. Because deterministic exhaustive execution of each supplied rule scores 1.000,

Table 4: Tier-3 matched-cardinality decomposition at fixed $K = 1$. Values are question-weighted means over five repetitions; terminal failures contribute zero.

Model	Corpus size	Macro-F1	Precision	Recall
GPT-5 mini	100	0.671	0.634	0.800
GPT-5 mini	Full	0.212	0.213	0.684
Qwen3.5-397B-A17B	100	0.630	0.600	0.743
Qwen3.5-397B-A17B	Full	0.187	0.200	0.432

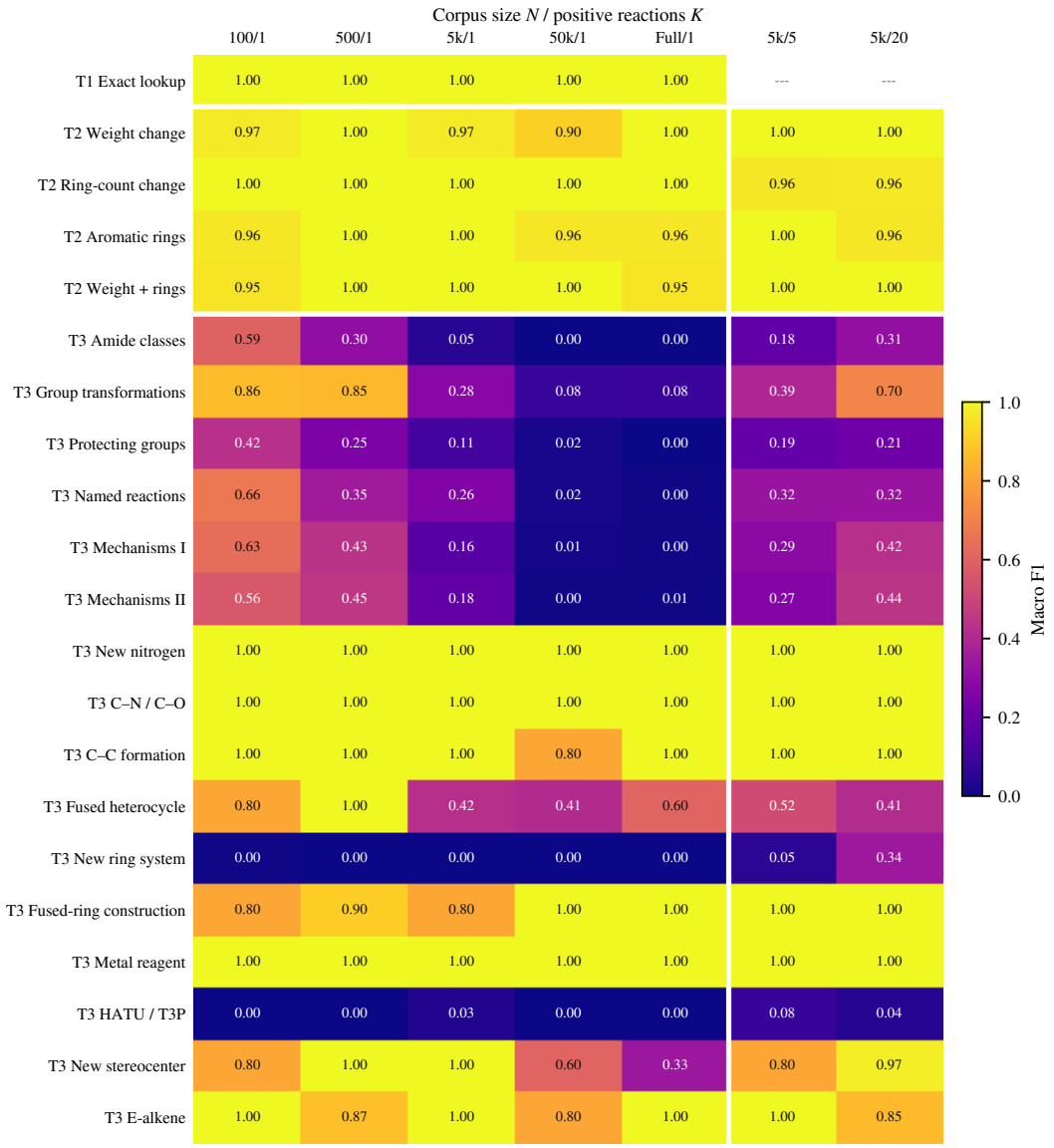

Figure 4: GPT-5 mini matched-cardinality macro-F1 by task. The first five columns increase corpus size N with one positive reaction ($K = 1$); the final two hold $N = 5{,}000$ and increase K. Each cell averages five repetitions. Tier 1 was evaluated only in the fixed-K scale sweep.

these residual errors cannot be attributed to incorrect rules or scoring. They show that an agent may fail to apply even a correct local rule consistently across the corpus or within a larger graph-search procedure.

Table 5: Full-corpus macro-F1 for ordinary RLM (Ord.), RLM supplied with the executable chemistry rule (Rule), and deterministic exhaustive execution (Exec.). Values are question-weighted means over five runs. Q is the number of questions in each task configuration.

		Qwen3.5-397B-A17B		Claude Haiku 4.5		
Task	Q	Ord.	Rule	Ord.	Rule	Exec.
T3 amide couplings (T3.11–14)	4	0.413	1.000	0.297	0.800	1.000
T3 mechanisms I (T3.26–30)	5	0.458	0.960	0.332	0.720	1.000
T3 new stereocenter (T3.9)	1	1.000	0.000	0.599	0.603	1.000
T4 group-constrained paths (T4.6–9)	4	0.692	0.556	0.243	0.204	1.000
T4 protecting-group paths (T4.10–11)	2	0.556	0.533	0.247	0.467	1.000

Figure 6 gives the complete task-by-context breakdown. Figures 7 and 8 report the associated resource profiles. Supplying the local rule reduces calls and tokens for Qwen3.5-397B-A17B at the two sampled contexts, but the full-corpus graph tasks remain expensive and the performance effect depends on the task. Claude Haiku 4.5 shows the same task dependence, with the largest full-corpus improvements on reaction-class and mechanism recognition.

A.7.1 Deterministic executor ceiling.

All 15 executor jobs—five task configurations at three context sizes—reach macro-F1 and exact-set-match accuracy of 1.000 with zero model calls, tokens, and API cost. Figure 9 shows that deterministic runtime grows at full-corpus scale, particularly for the two graph operations, while peak memory remains below 341 MiB. This verifies the supplied rules, exhaustive execution path, and scorers independently of model behavior.

A.8 Truncated-Route Control by Model and Target

Figure 10 complements Table 6 with precision, recall, and exact-route accuracy. Figure 11 reveals that the aggregate gain from the final-step class is target-dependent: Qwen3.5-397B-A17B improves on the lactam dipeptide and benzamide–pyrazole targets, while Claude Haiku 4.5's gain is concentrated on benzamide–pyrazole. Neither model recovers the pyrimidine–piperazine route from the structure-plus-class prompt. Figure 12 reports the computational profile of all 30 jobs and 90 target-level trajectories.

Table 6: Truncated-route control. Values are mean macro-F1 $\pm$ one run-level standard deviation over three targets and five repetitions. Every reaction producing the exact target is excluded from the searchable corpus.

Model	Target name	Target structure	Structure + final-step class
Qwen3.5-397B-A17B	0.000 ± 0.000	0.000 ± 0.000	0.378 ± 0.169
Claude Haiku 4.5	0.052 ± 0.117	0.000 ± 0.000	0.133 ± 0.183

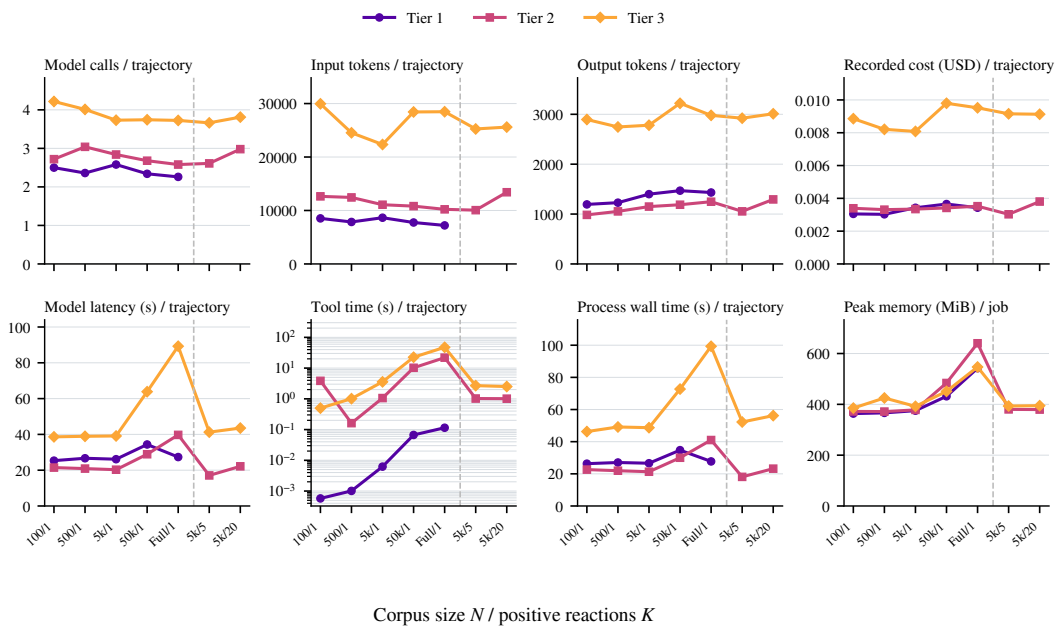

Figure 5: GPT-5 mini matched-cardinality resource profile. Additive quantities are normalized per successfully scored question trajectory; peak combined host-process-tree plus Docker memory is averaged per job. The dashed divider separates the fixed-K scale sweep from the fixed-N cardinality sweep. Input and output tokens are reported separately.

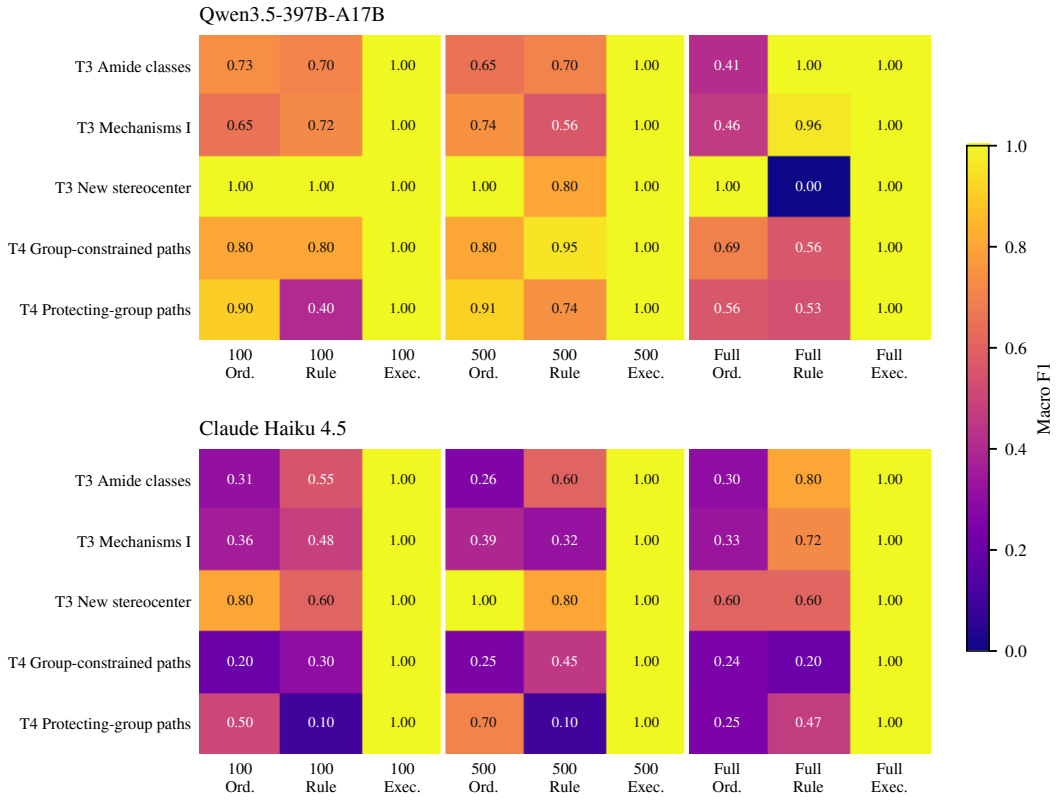

Figure 6: Chemistry-rule control by task, context, and model. Ordinary RLM (Ord.) and rule-supplied RLM (Rule) are averaged over five repetitions. Deterministic exhaustive execution (Exec.) applies the same rule to every record and provides the verified ceiling.

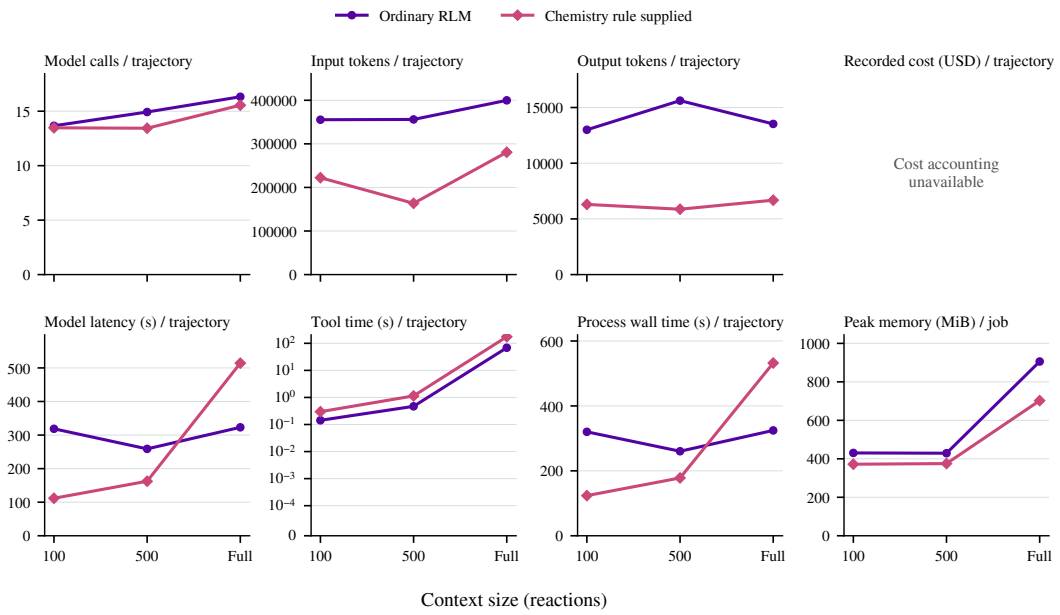

Figure 7: Qwen3.5-397B-A17B chemistry-rule resource profile. Calls, tokens, recorded model latency, tool time, and process wall time are normalized per successfully scored question trajectory; peak memory is averaged per job. API-cost accounting is unavailable for the ordinary-RLM runs and is therefore not compared.

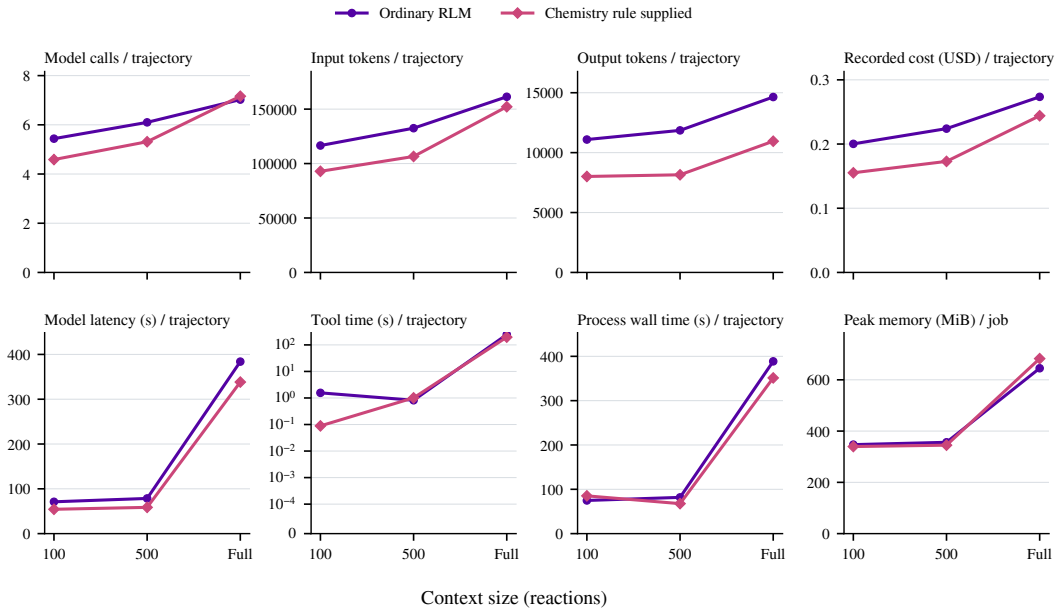

Figure 8: Claude Haiku 4.5 chemistry-rule resource profile, with the same aggregation and normalization as Figure 7.

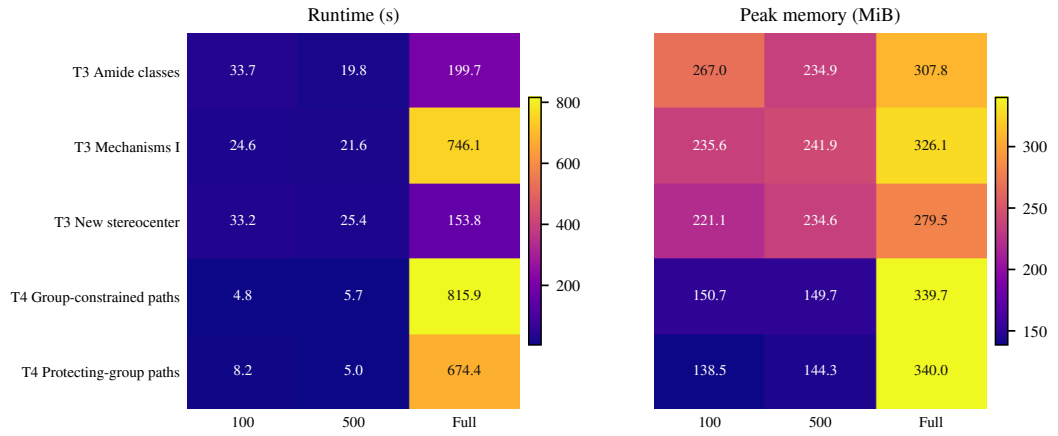

Figure 9: Runtime and peak memory for deterministic exhaustive execution of the five chemist-validated rules. Every cell scores macro-F1 and exact-set-match accuracy of 1.000 and uses no language-model calls.

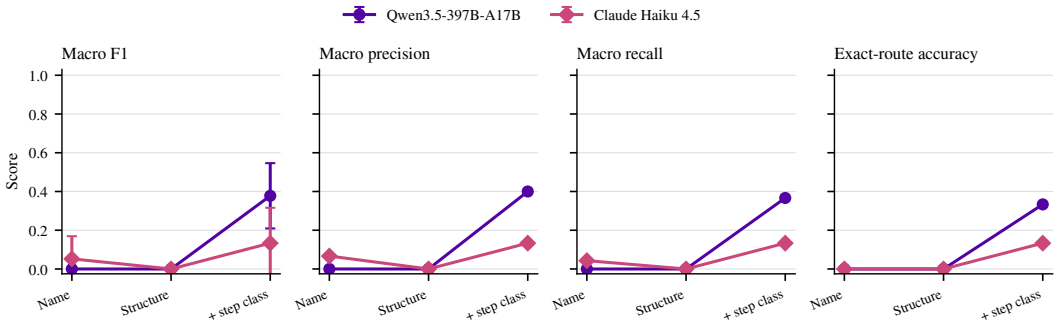

Figure 10: Aggregate truncated-route control across three targets and five repetitions. Error bars on macro-F1 show one run-level standard deviation. Exact-route accuracy requires the complete predicted chain set to equal the reference set.

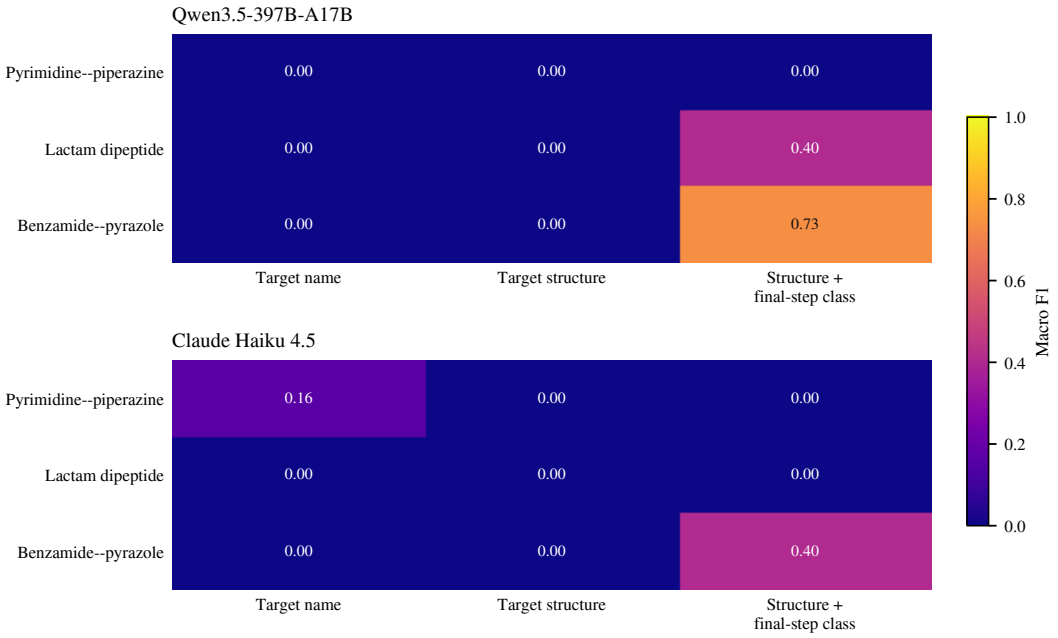

Figure 11: Truncated-route macro-F1 separated by model, target, and prompt condition. Each cell averages five repetitions after removing every reaction that directly produces the target.

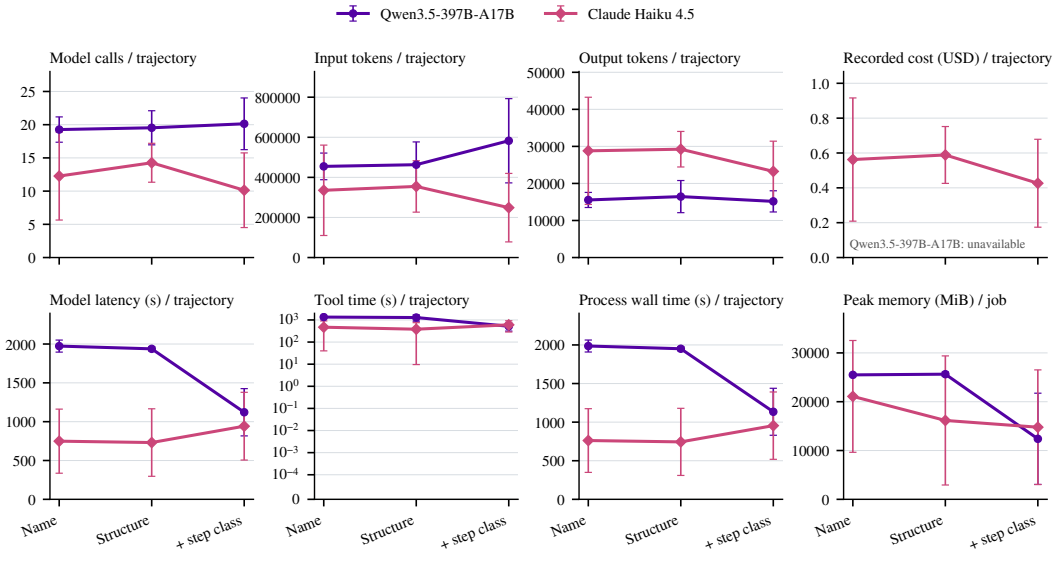

Figure 12: Truncated-route resource diagnostics. Calls, tokens, API cost, model latency, tool time, and wall time are normalized per target-level trajectory; peak combined memory is averaged per job. Points show means over five jobs and error bars show one standard deviation. Qwen3.5-397B-A17B cost accounting is unavailable and is omitted.

B PERFORMANCE AND RESOURCE PROFILES

Figure 13 shows the accuracy–cost trade-off across Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5, the three models with complete paid-API accounting for the core benchmark. Figure 14 separates the same observations by model. The profiles that follow report performance, cost, tokens, time, memory, and task-level variation from the frozen per-run records.

For these three models at $x = 100$, CodeAct costs \$0.009–\$0.042 per question trajectory across the four tiers, compared with \$0.018–\$0.088 for RLM. At the same $x = 500$ context on Tier 4, RLM reaches 0.578 F1 at \$0.108 per trajectory, while CodeAct reaches 0.553 at \$0.076. RLM alone extends this interface to all 122,456 reactions; the three-model mean then costs \$0.168 per trajectory and scores 0.332. Recursive inference therefore enables access to a corpus that cannot fit in the prompt, but does not by itself provide the chemical rules or multi-step reasoning required by the hardest questions.

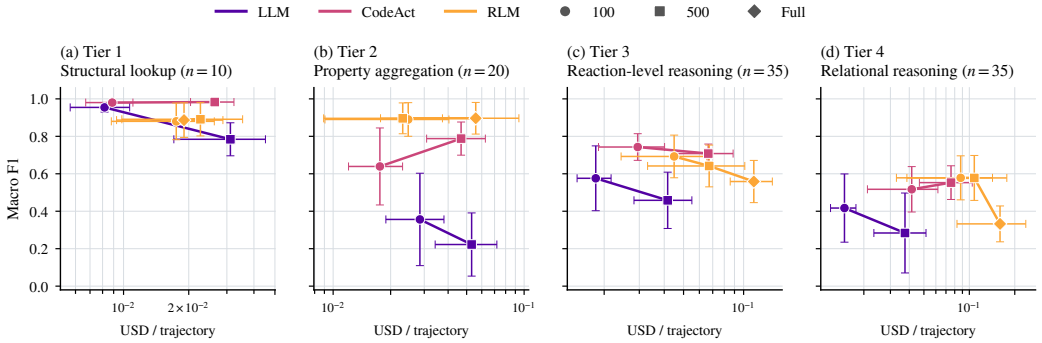

Figure 13: Cross-model accuracy–cost trade-off by tier for Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5. Points show means across the three model-level values; horizontal and vertical bars show standard errors across models. Cost is recorded in USD per question trajectory on a logarithmic axis.

B.1 EFFICIENCY BEYOND DOLLAR COST

Figure 15 characterizes RLM efficiency using 2,246 successful jobs, comprising 7,460 question-level trajectories from Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5. For each model, calls, tokens, recorded latency, tool time, and process wall time are normalized per successful trajectory, while peak memory is averaged per successful job. The plotted points are unweighted means across the five models, with standard errors capturing model-to-model variation. At Tier 4, moving RLM from $x = 500$ to full-corpus access raises the across-model mean from 11.79 to 13.68 calls, 354k to 411k tokens, 238 to 704 seconds of recorded latency, 8.22 to 319 seconds of tool execution, 241 to 708 seconds of process wall time per successful trajectory, and 413 MiB to 3.50 GiB peak memory per job. The latency, tool-time, and process-wall counters are independently recorded and represent separate measurements. Every trajectory in this analysis has a complete record for all six measures.

B.2 QWEN3.5-397B-A17B PROFILE

Figures 16–18 report the Qwen3.5-397B-A17B benchmark profile. The main evaluation contains 1,050 terminal jobs and 3,500 question-level trajectories: plain LLM and CodeAct at $x \in \{100, 500\}$, and RLM at $x \in \{100, 500, \text{full}\}$. The extended-scale evaluation adds CodeAct and RLM at $x = 1000$, with 150 successful jobs and 500 trajectories per interface. One main-matrix RLM job ended without a valid answer and contributes zero to macro-F1. Additive resource measures use successfully scored question trajectories, and peak memory is averaged over successful jobs.

B.3 COMPLETE DEEPSEEK V4 FLASH PROFILE

Figures 19–21 report the complete DeepSeek V4 Flash benchmark profile. The main evaluation contains 1,050 terminal jobs and 3,500 question-level trajectories: plain LLM and CodeAct at

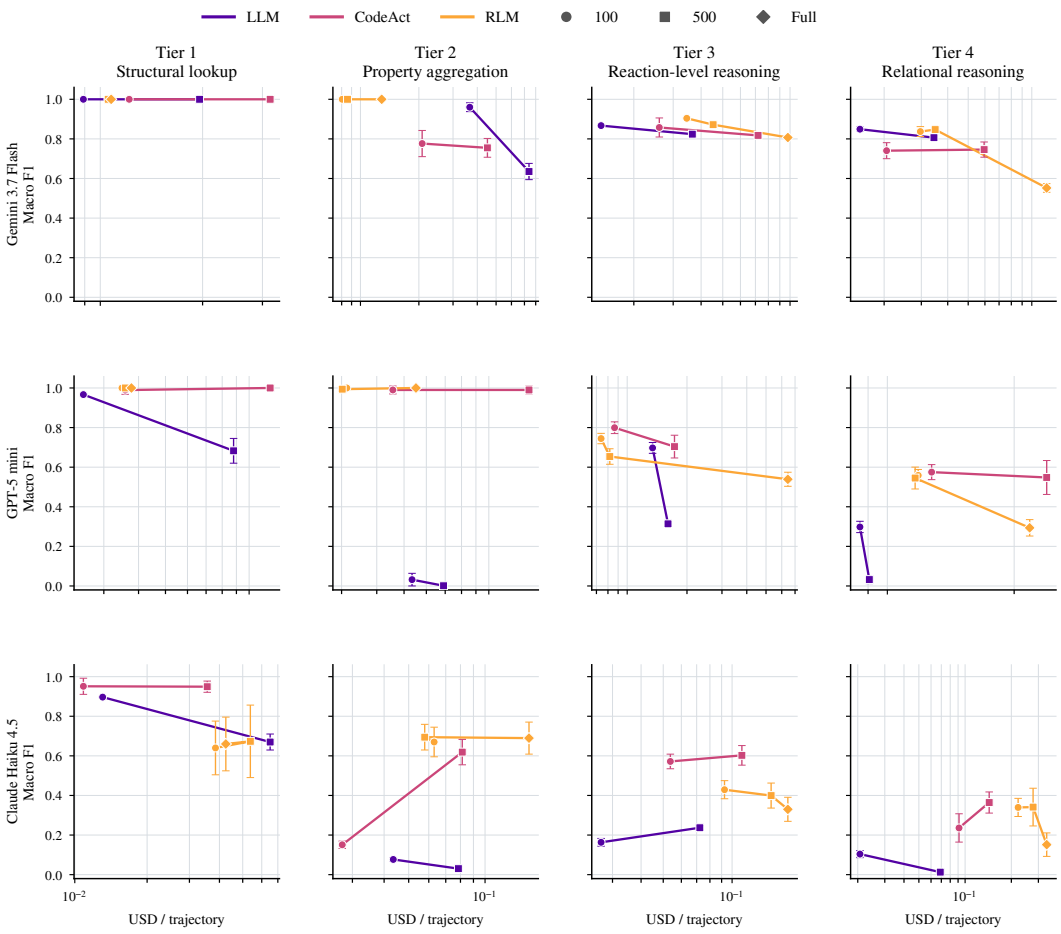

Figure 14: Model-separated accuracy–cost frontiers for Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5. Points show means over five repetitions; vertical bars show one population standard deviation.

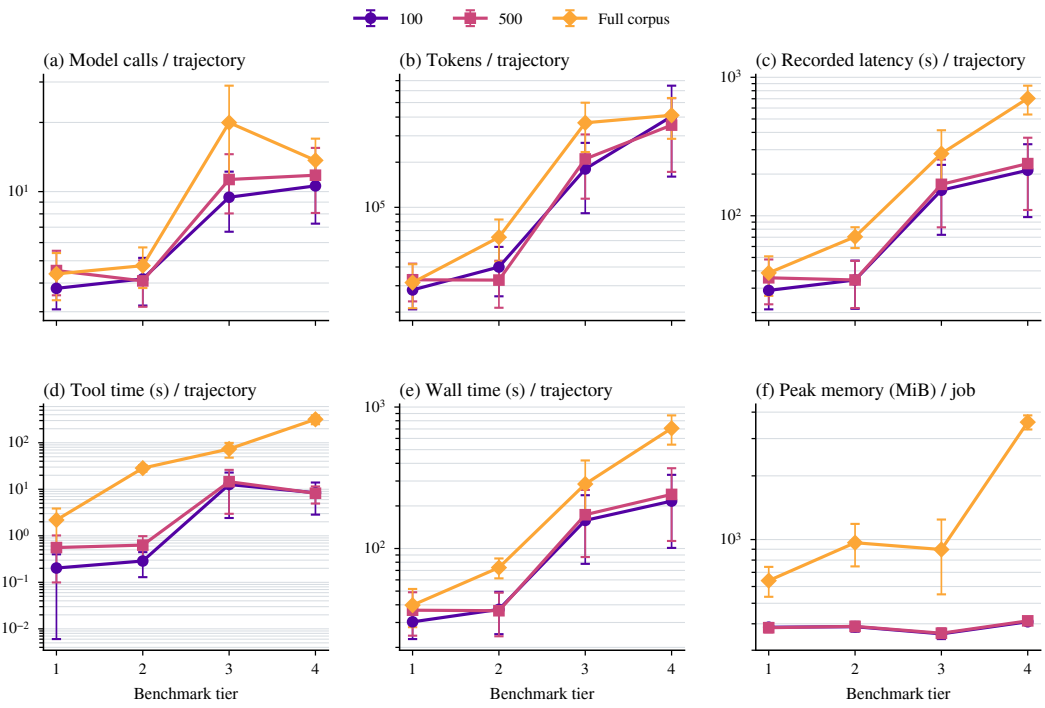

Figure 15: RLM efficiency beyond API cost. Curves show unweighted means and standard errors across Qwen3.5-397B-A17B, DeepSeek V4 Flash, Gemini 3.7 Flash, GPT-5 mini, and Claude Haiku 4.5. Calls, tokens, recorded latency, tool time, and wall time are normalized per successful scored question trajectory; peak combined host-process-tree plus Docker memory is averaged per successful job.

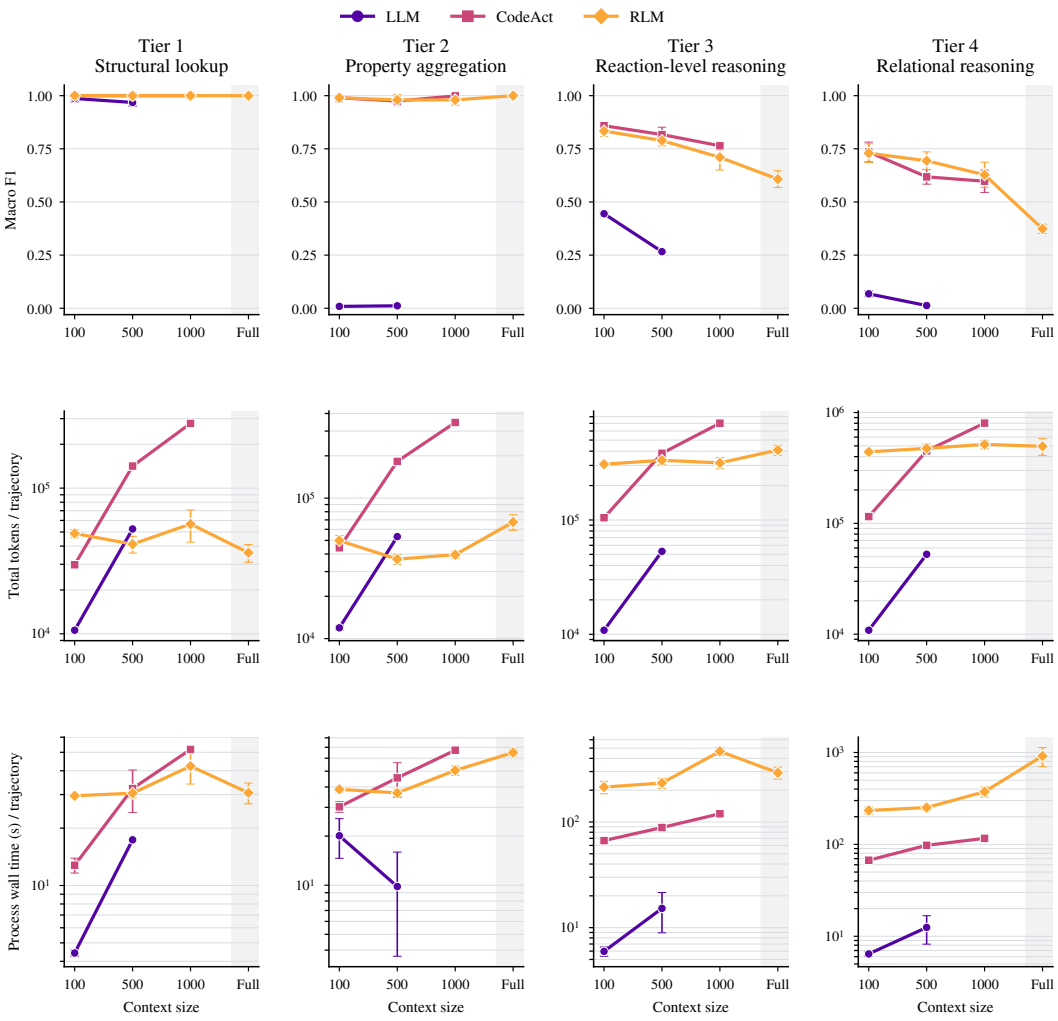

Figure 16: Qwen3.5-397B-A17B performance and efficiency by tier. Bars show one population standard deviation across five repetitions. Token and process-wall-time means are normalized per successfully scored question trajectory; incomplete API-cost accounting is omitted. The $x = 1000$ extensions cover CodeAct and RLM, while full-corpus access is RLM-only.

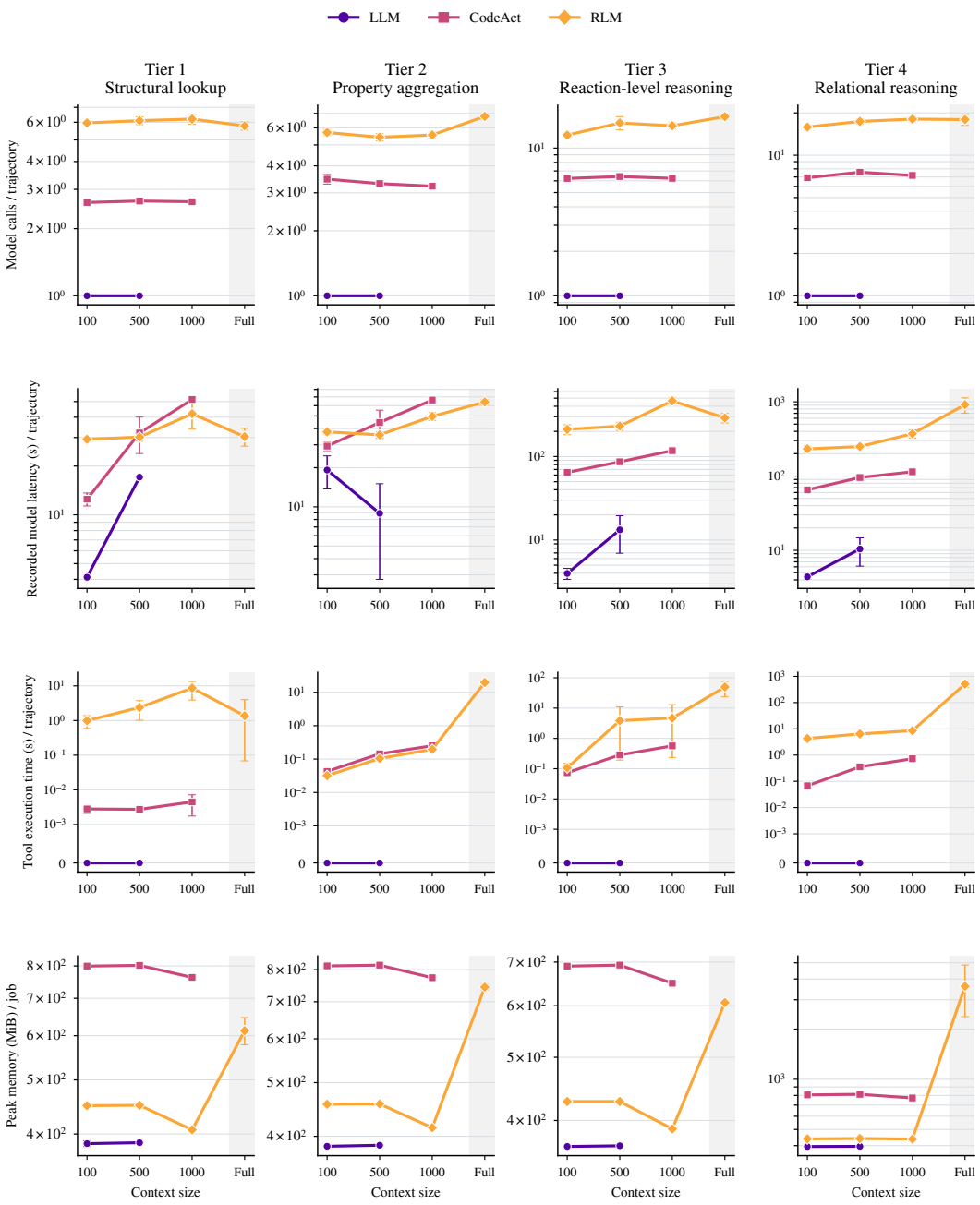

Figure 17: Qwen3.5-397B-A17B resource diagnostics by tier. Model calls count successful language-model API calls, recorded latency accumulates model-call latency, and tool time records local Python/RDKit execution. These additive measures are normalized per successfully scored question trajectory. Peak memory is the mean per-job maximum of the combined host process tree and Docker processes.

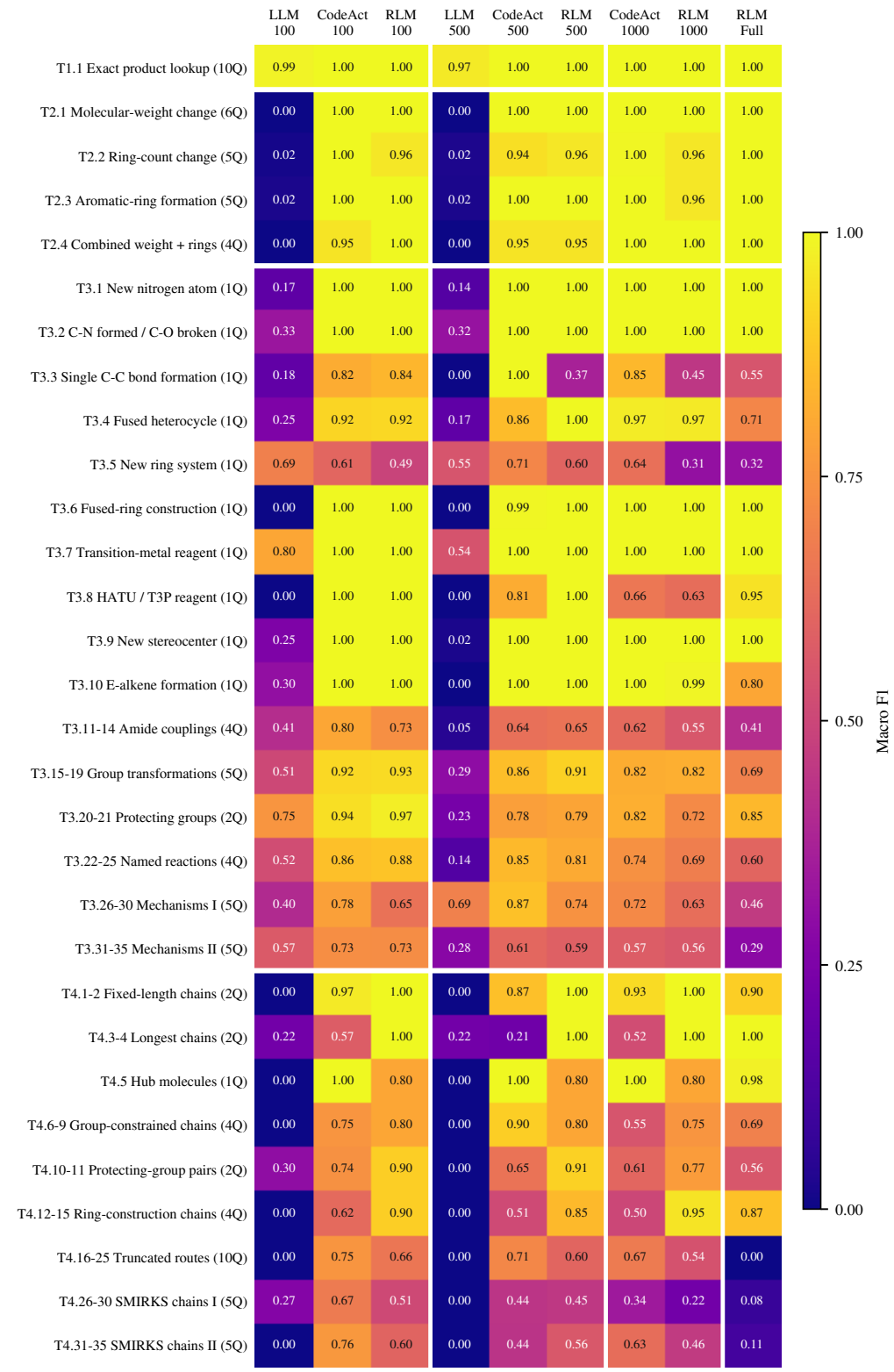

Figure 18: Qwen3.5-397B-A17B task-level macro-F1, averaged across five repetitions. The terminal failure contributes zero. The heatmap preserves all 30 task configurations and includes the $x = 1000$ CodeAct and RLM extensions; full-corpus access is RLM-only.

$x \in \{100, 500\}$, and RLM at $x \in \{100, 500, \text{full}\}$. The extended-scale evaluation adds CodeAct and RLM at $x = 1000$, with 150 jobs and 500 trajectories per interface. The standard and extended CodeAct arms contain 31 and 65 terminal failures; these contribute zero to macro-F1. All 450 standard RLM jobs and all 150 RLM $x = 1000$ jobs succeeded. Additive resource measures use successfully scored question trajectories, and peak memory is averaged over successful jobs.

B.4 Complete Claude Haiku 4.5 Profile

Figures 22–24 report the complete Claude Haiku 4.5 benchmark profile. The evaluation contains 1,050 successful jobs and 3,500 question-level trajectories: plain LLM and CodeAct at $x \in \{100, 500\}$, and RLM at $x \in \{100, 500, \text{full}\}$. Each point in the tier summaries averages five repetitions. Macro-F1 is question-weighted within each repetition; additive resource measures are normalized by the number of successfully scored question trajectories, and peak memory is averaged over successful jobs.

B.5 Complete Gemini 3.7 Flash Profile

Figures 25–27 report the complete Gemini 3.7 Flash benchmark profile. The main evaluation contains 1,050 terminal jobs and 3,500 question-level trajectories: plain LLM and CodeAct at $x \in \{100, 500\}$, and RLM at $x \in \{100, 500, \text{full}\}$. The extended-scale evaluation adds CodeAct and RLM at $x = 1000$, with 150 jobs and 500 trajectories per interface. Terminal failed jobs contribute zero to macro-F1. Additive resource measures use successfully scored question trajectories, and peak memory is averaged over successful jobs.

B.6 GPT-5 mini Profile

Figures 28–30 report the GPT-5 mini benchmark profile. The main evaluation contains 1,050 successful jobs and 3,500 question-level trajectories: plain LLM and CodeAct at $x \in \{100, 500\}$, and RLM at $x \in \{100, 500, \text{full}\}$. Extended-scale CodeAct and RLM evaluations each add 150 jobs and 500 trajectories at $x = 1000$. Terminal failed jobs contribute zero to macro-F1. Additive resource measures use successfully scored question trajectories, and peak memory is averaged over successful jobs.

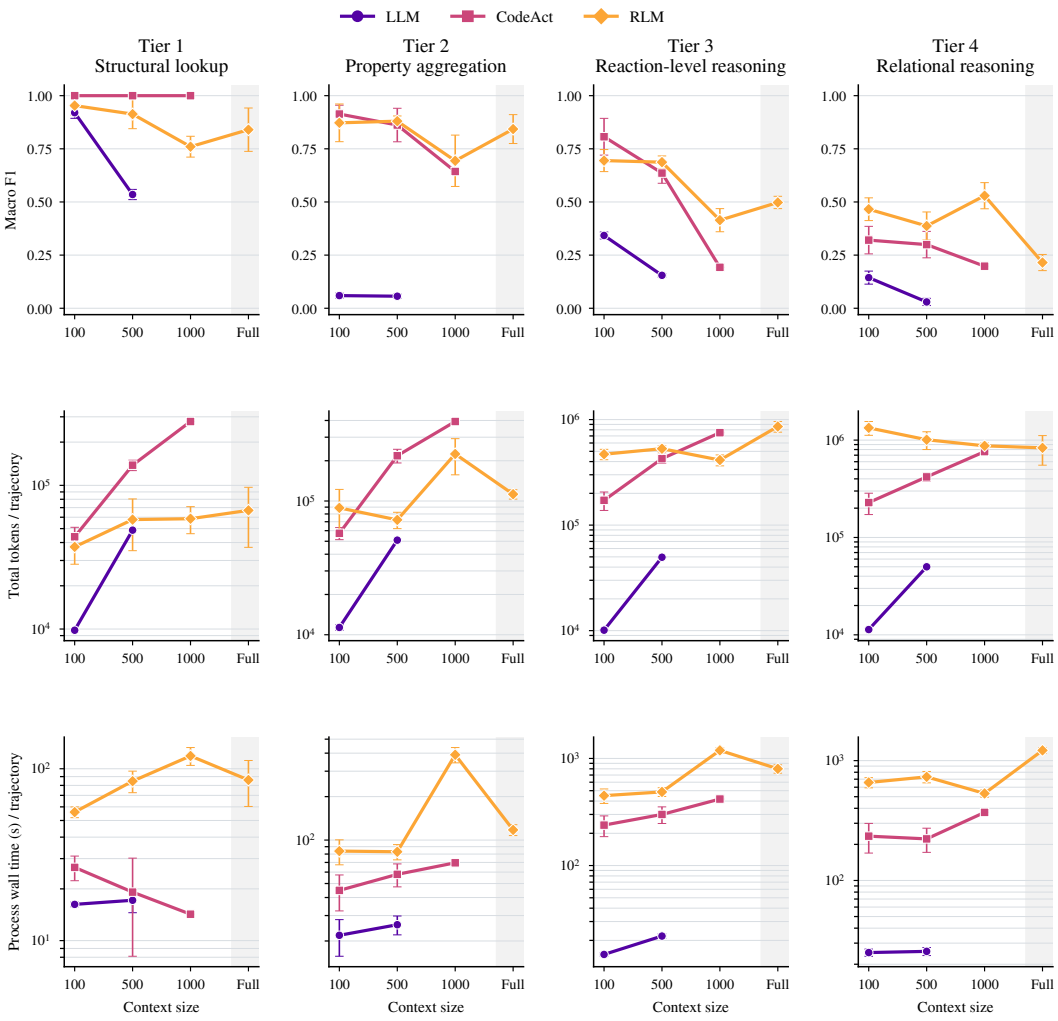

Figure 19: DeepSeek V4 Flash performance and efficiency by tier. Bars show one population standard deviation across five repetitions. Token and process-wall-time means are normalized per successfully scored question trajectory; incomplete API-cost accounting is omitted. The $x = 1000$ extension covers CodeAct and RLM, while full-corpus access is RLM-only.

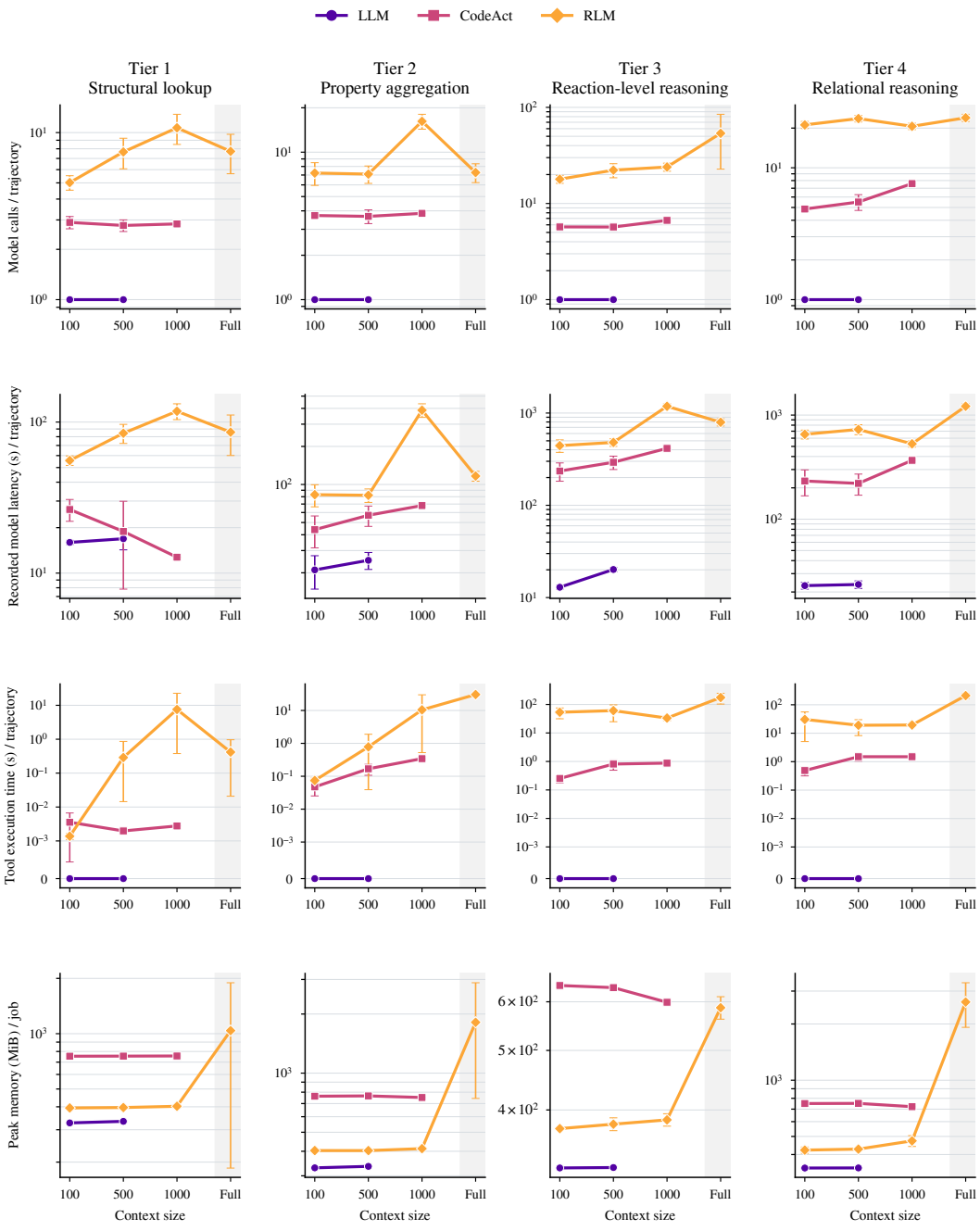

Figure 20: DeepSeek V4 Flash resource diagnostics by tier. Model calls count successful language-model API calls, recorded latency accumulates model-call latency, and tool time records local Python/RDKit execution. These additive measures are normalized per successfully scored question trajectory. Peak memory is the mean per-job maximum of the combined host process tree and Docker processes.

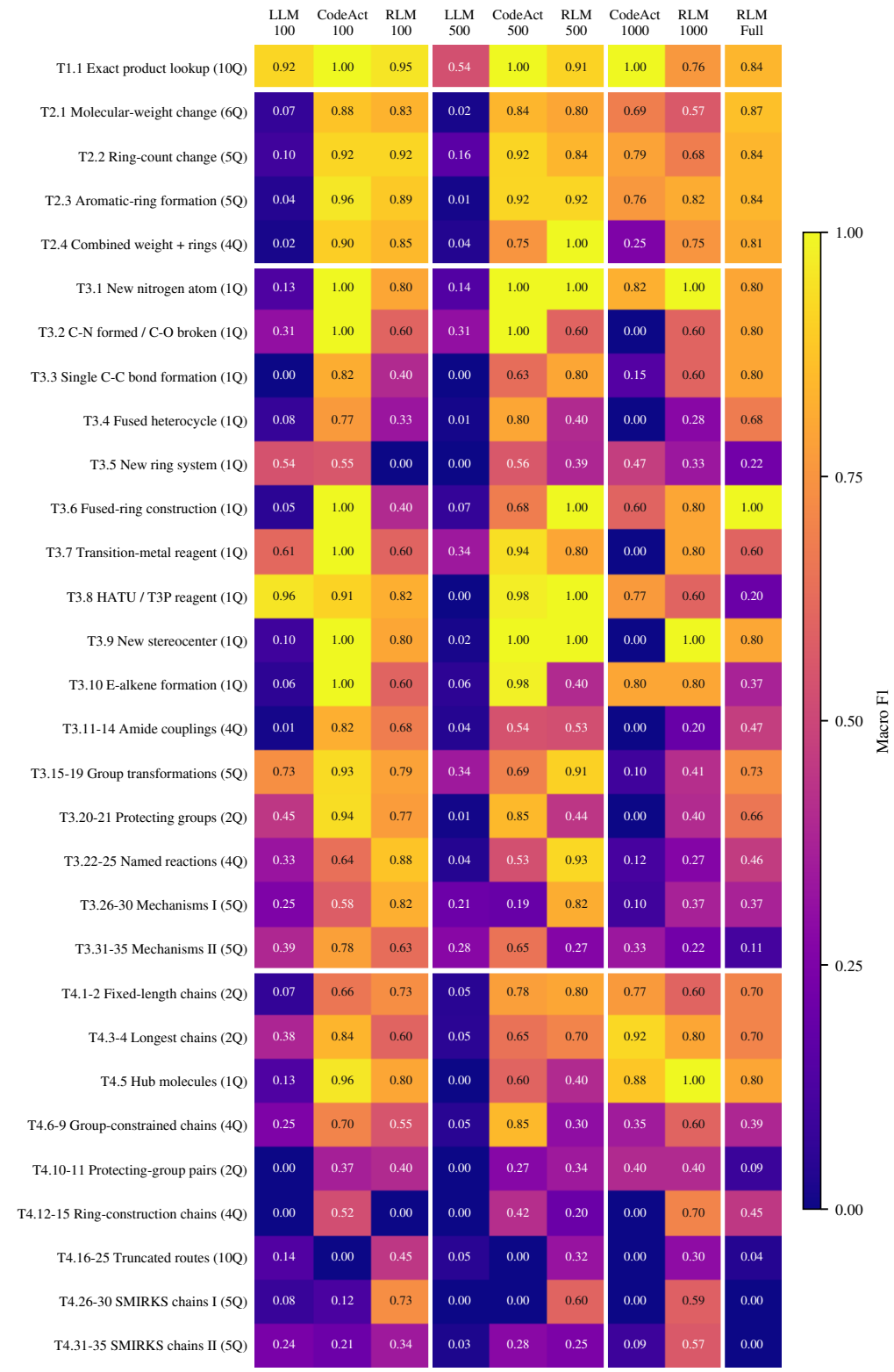

Figure 21: DeepSeek V4 Flash task-level macro-F1, averaged across five repetitions. Terminal failures contribute zero. The heatmap preserves all 30 task configurations and includes the $x = 1000$ CodeAct and RLM extensions; full-corpus access is RLM-only.

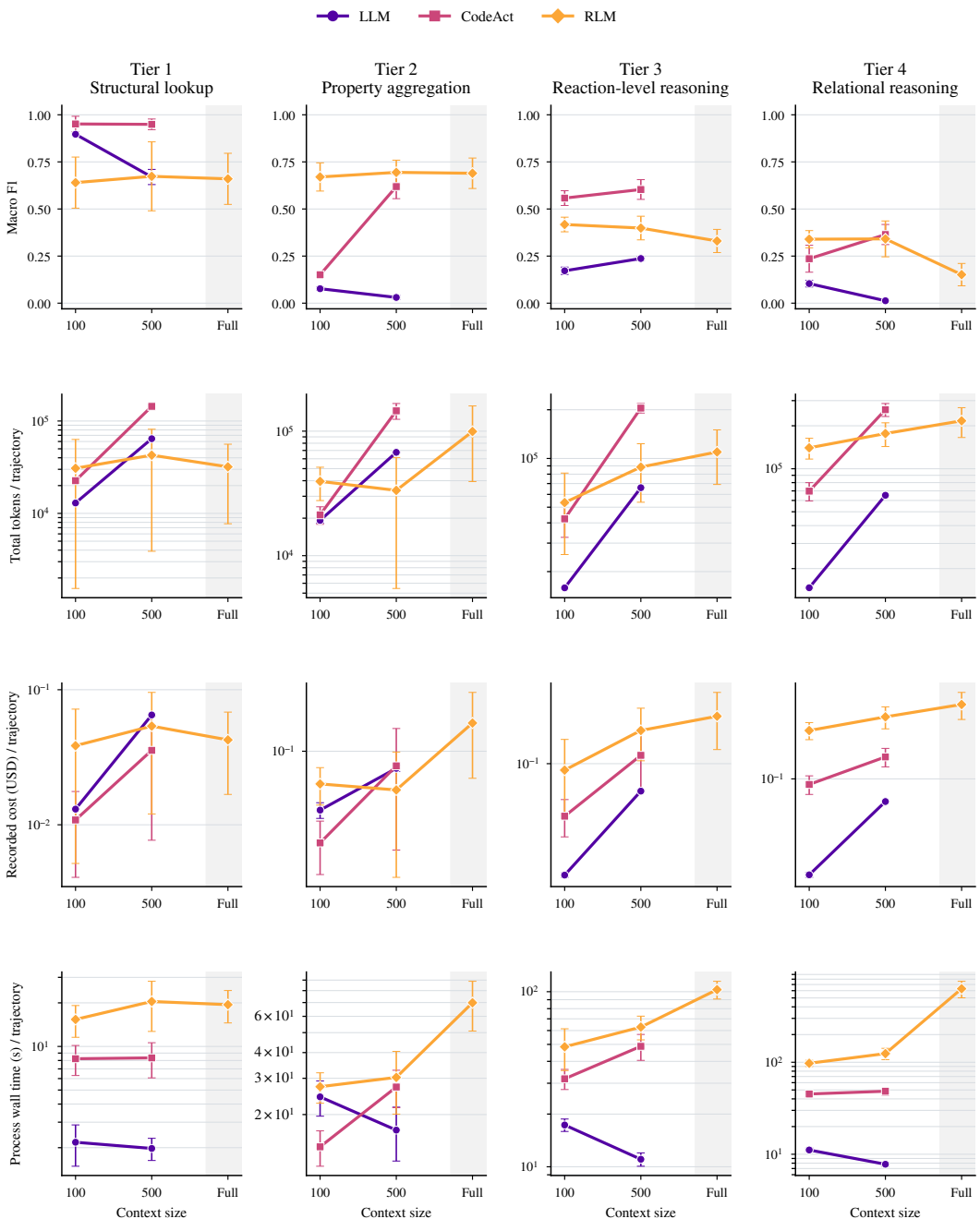

Figure 22: Claude Haiku 4.5 performance and efficiency by tier. Bars show one population standard deviation across five repetitions. API cost is reported in USD; token and process-wall-time means are normalized per successfully scored question trajectory. Full-corpus access is evaluated only for RLM.

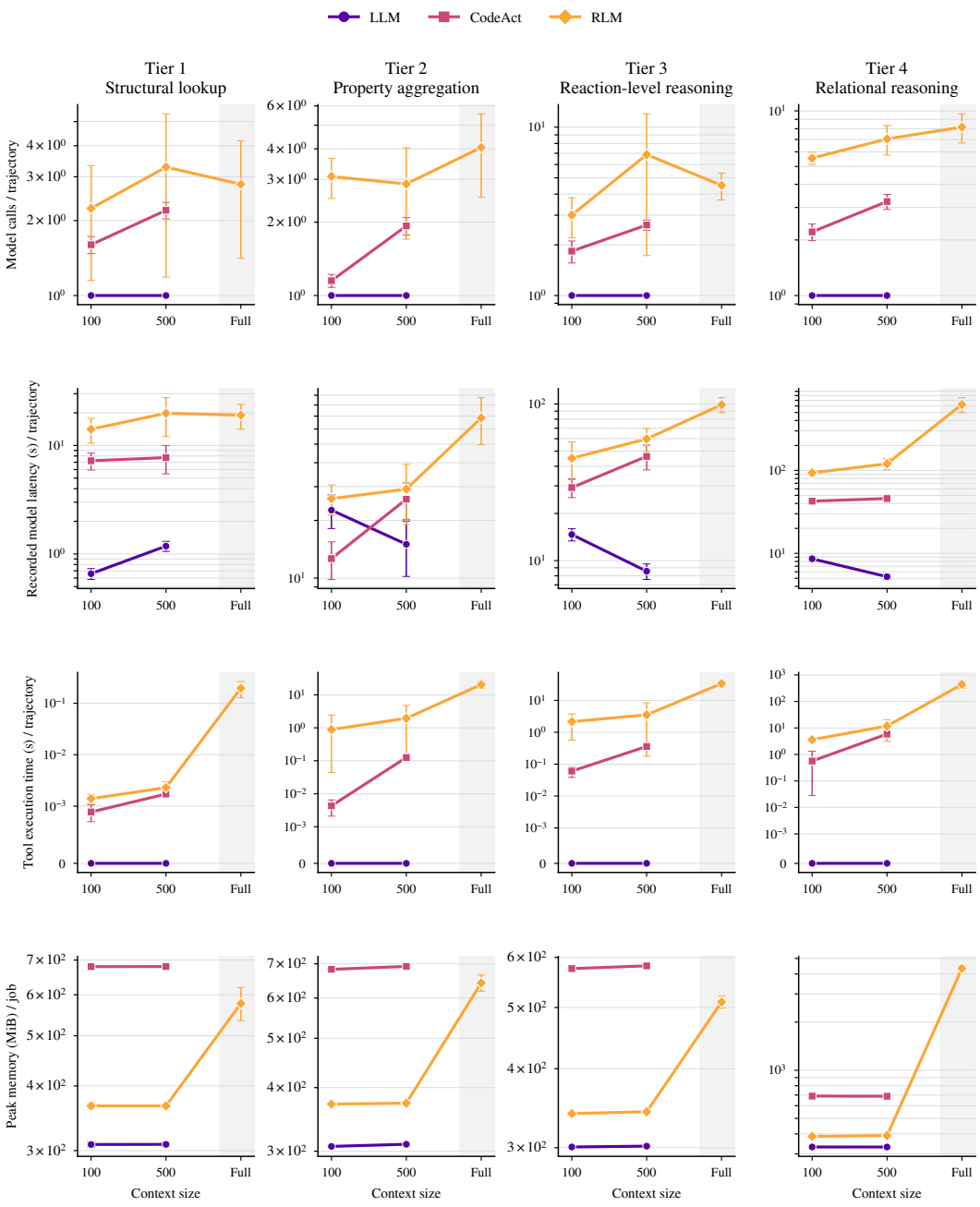

Figure 23: Claude Haiku 4.5 resource diagnostics by tier. Model calls count successful language-model API calls, recorded latency accumulates model-call latency, and tool time records local Python/RDKit execution. These additive measures are normalized per successfully scored question trajectory. Peak memory is the mean per-job maximum of the combined host process tree and Docker processes.

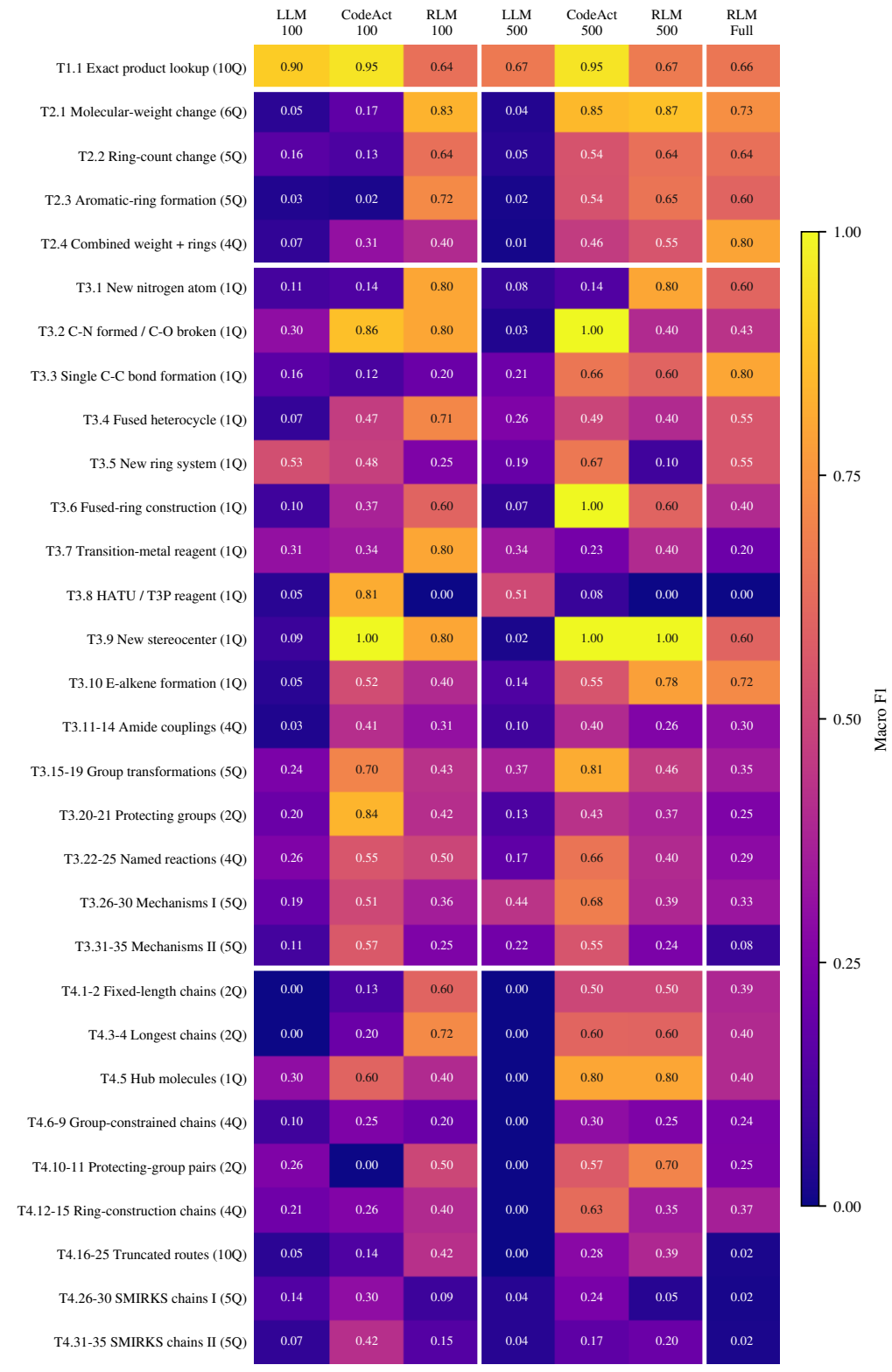

Figure 24: Claude Haiku 4.5 task-level macro-F1, averaged across five repetitions. The heatmap preserves the individual task configurations hidden by tier-level aggregation. Full-corpus access is RLM-only.

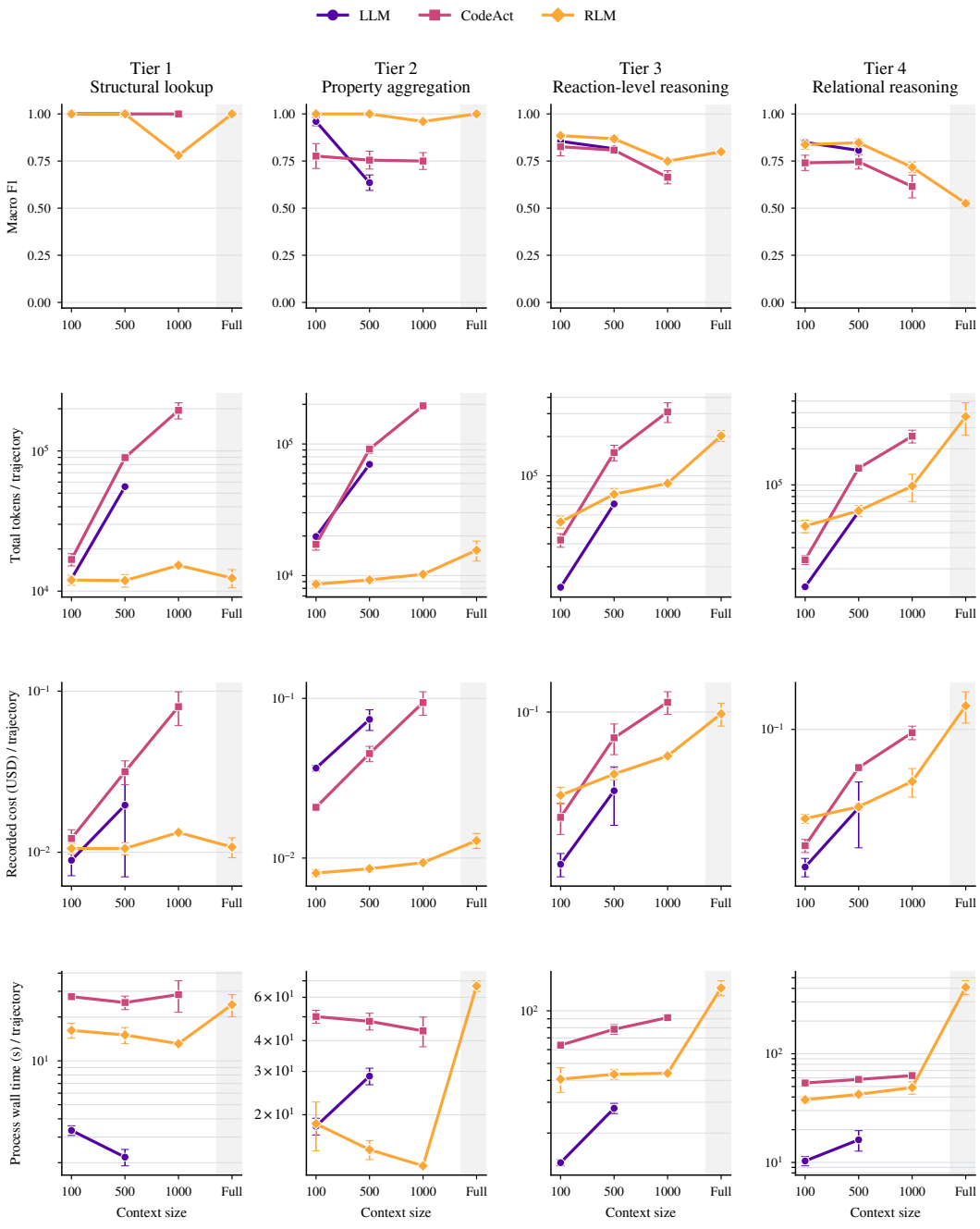

Figure 25: Gemini 3.7 Flash performance and efficiency by tier. Bars show one population standard deviation across five repetitions. API cost is reported in USD; token and process-wall-time means are normalized per successfully scored question trajectory. The $x = 1000$ extension covers CodeAct and RLM, while full-corpus access is RLM-only.

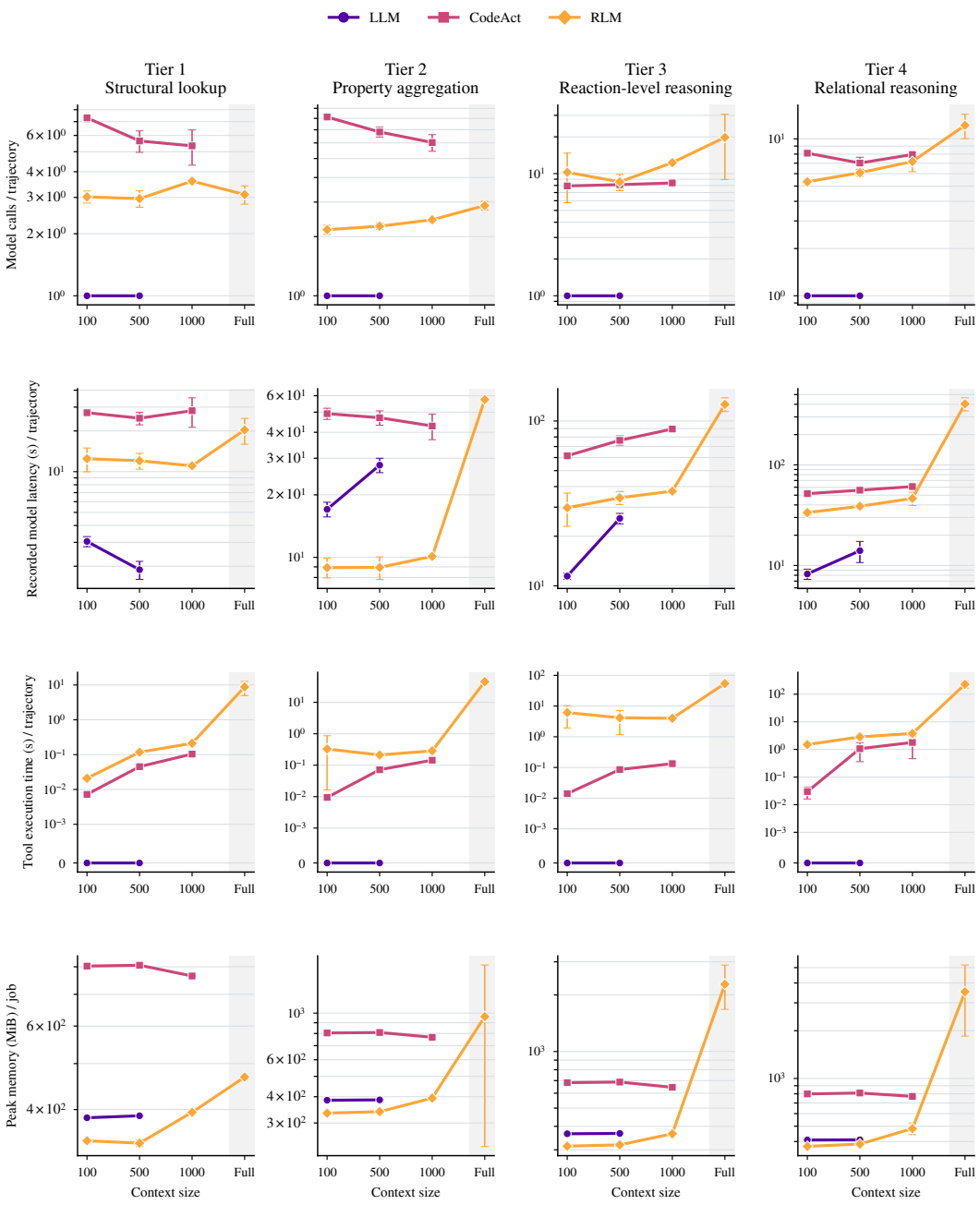

Figure 26: Gemini 3.7 Flash resource diagnostics by tier. Model calls count successful language-model API calls, recorded latency accumulates model-call latency, and tool time records local Python/RDKit execution. These additive measures are normalized per successfully scored question trajectory. Peak memory is the mean per-job maximum of the combined host process tree and Docker processes.

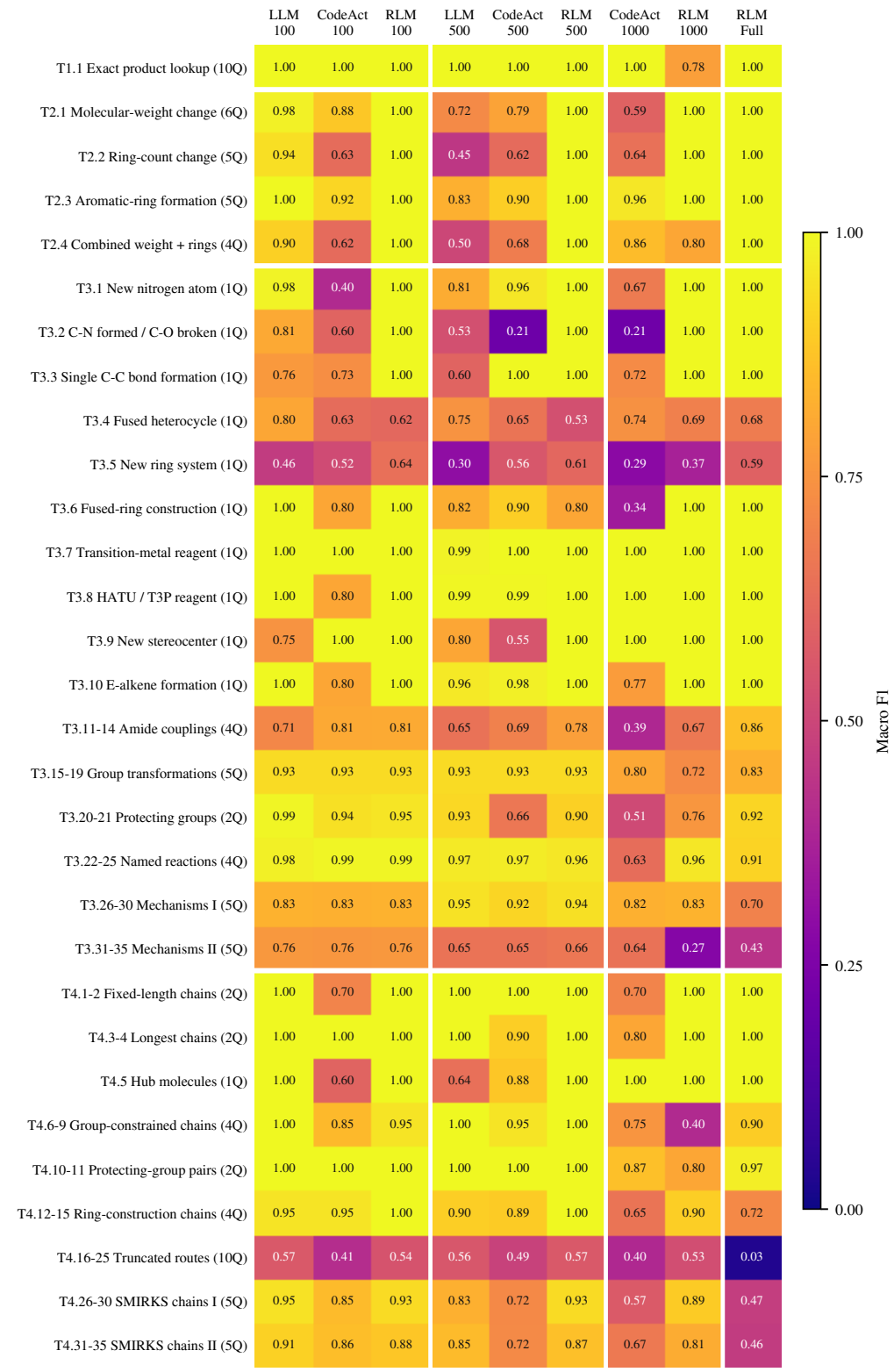

Figure 27: Gemini 3.7 Flash task-level macro-F1, averaged across five repetitions. Terminal failures contribute zero. The heatmap preserves all 30 task configurations and includes the $x = 1000$ CodeAct and RLM extensions; full-corpus access is RLM-only.

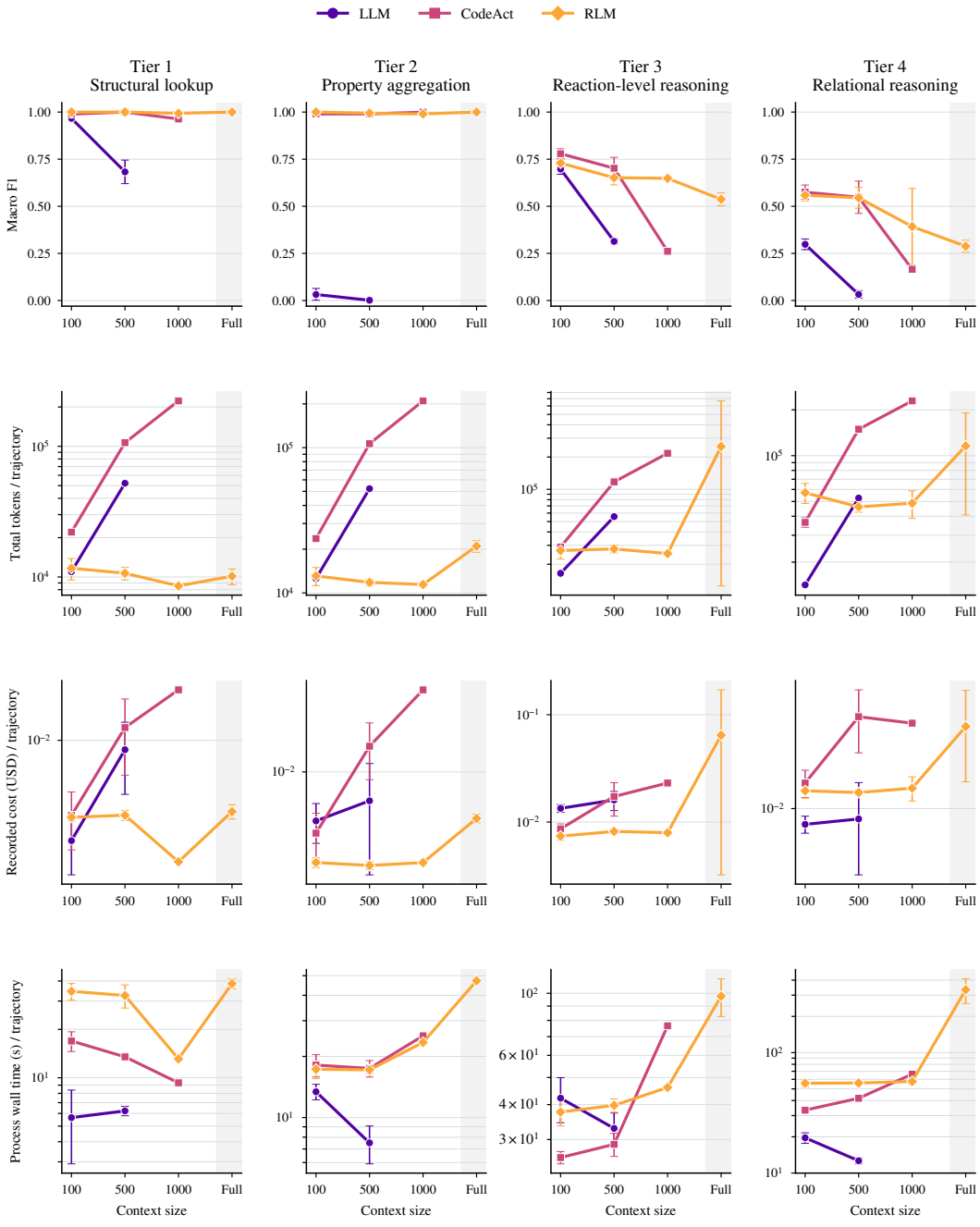

Figure 28: GPT-5 mini performance and efficiency by tier. Bars show one population standard deviation across five repetitions. API cost is reported in USD; token and process-wall-time means are normalized per successfully scored question trajectory. The $x = 1000$ extensions cover CodeAct and RLM, while full-corpus access is RLM-only.

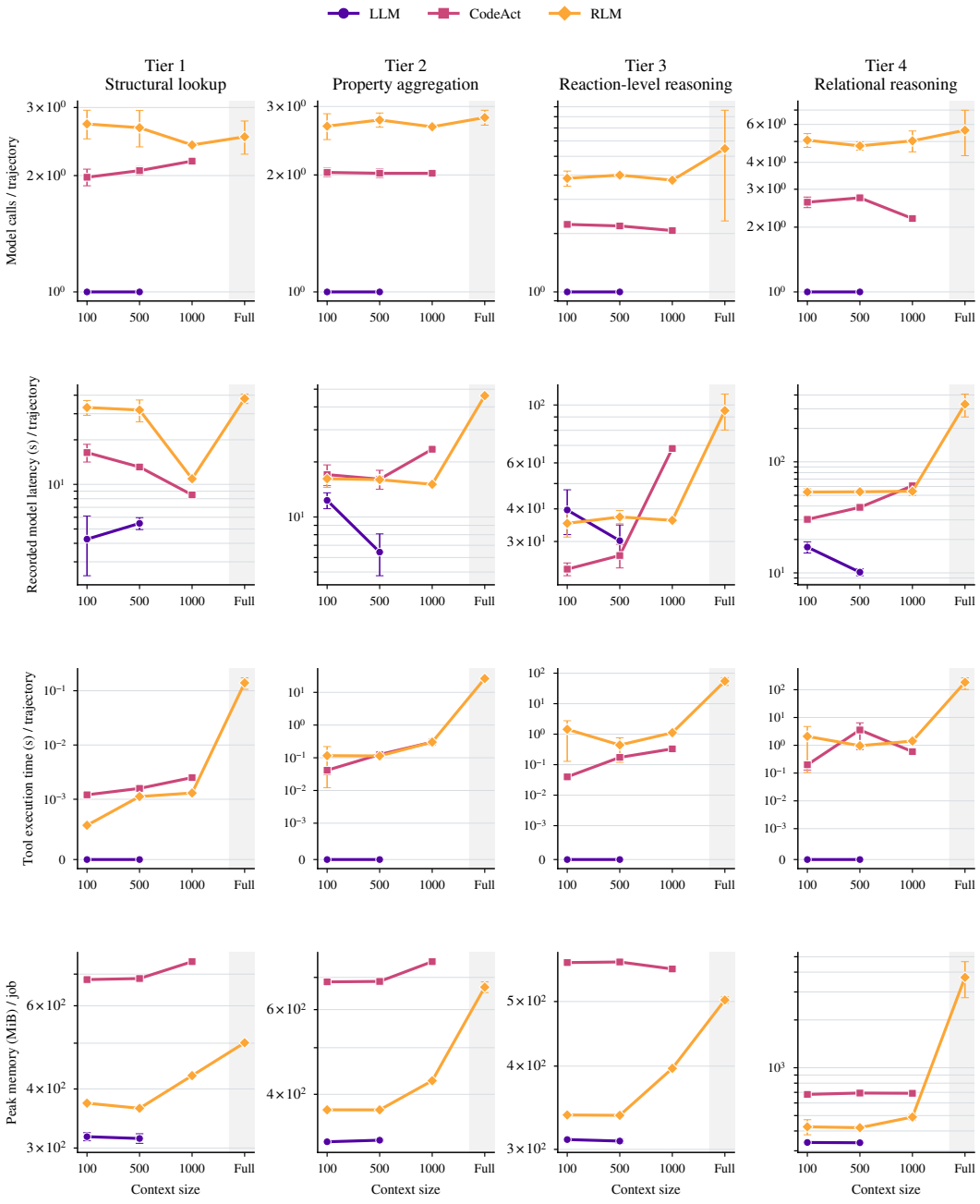

Figure 29: GPT-5 mini resource diagnostics by tier. Model calls count successful language-model API calls, recorded latency accumulates model-call latency, and tool time records local Python/RDKit execution. These additive measures are normalized per successfully scored question trajectory. Peak memory is the mean per-job maximum of the combined host process tree and Docker processes.

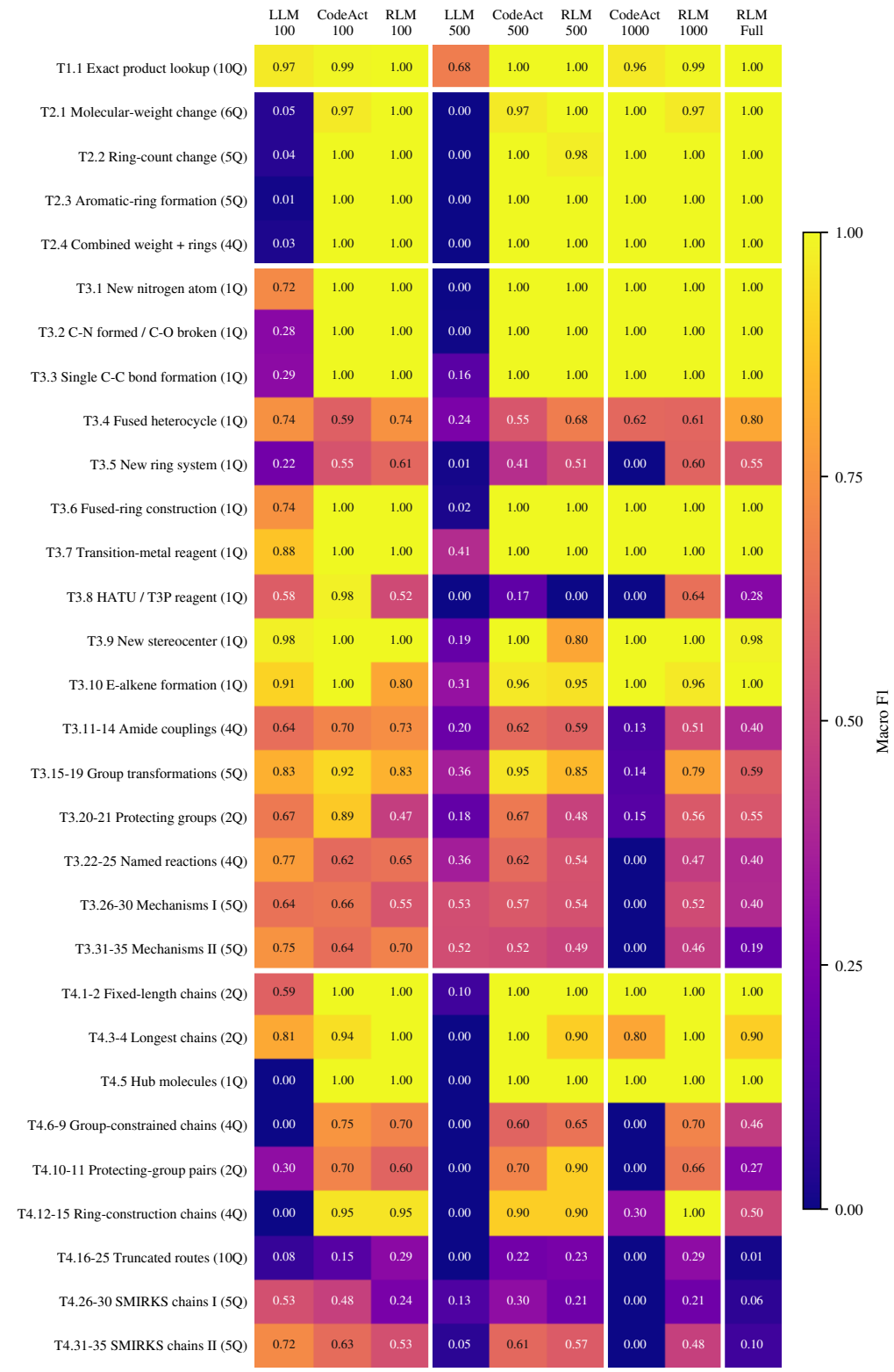

Figure 30: GPT-5 mini task-level macro-F1, averaged across five repetitions. Terminal failures contribute zero. The heatmap preserves all 30 task configurations and includes the $x = 1000$ CodeAct and RLM extensions; full-corpus access is RLM-only.

C TRACE-BASED FAILURE DIAGNOSTICS

We interpret failures by following the attempted computation through generated code, tool observations, intermediate strategy changes, and the scored answer. The review is stratified across the five paper models, all three interfaces, and the reaction-level and relational task families. A scientific cause is recorded only when the trace supplies direct evidence; runs containing provider or transport failures remain execution failures. Because the review is designed to cover distinct mechanisms, the examples establish where and how a capability can fail, while the benchmark scores quantify performance.

The quantitative trace cohort contains 6,179 jobs and 20,771 expected question trajectories with reproducible final scores. Figure 32 separates exact, partial, zero-score, and execution-failure outcomes across this population. Its second panel uses 2,229 corrected-task question outputs whose retained predictions permit exact reconstruction of precision and recall. We label only observable set differences: empty answers, omissions without false positives, extra selections without false negatives, or mixtures of both. Another 1,441 recovered outputs lack sufficient retained detail and are not imputed. These signatures identify where to inspect traces; they do not by themselves establish a chemical cause.

The benchmark construction provides the necessary controls. Every scored positive is included in the sampled contexts, so a literal reagent-name search or brittle SMARTS occurs after evidence access has been resolved. Full-corpus RLM supplies access to all 122,456 reactions, allowing incomplete enumeration and overbroad chain rules to be attributed to chemistry reasoning or operation failures. The chemistry-rule-supplied control removes rule induction, while deterministic execution removes model operation. Mechanical graph tasks and the truncated-route control provide corresponding controls for relational and route-level failures.

The examples support three conclusions. First, executable or recursive access does not guarantee systematic execution: tool syntax, corpus parsing, and state preservation can fail before chemical correctness is evaluated. Second, representation matters. Literal names, raw SMILES fragments, and narrow SMARTS are unreliable substitutes for atom correspondence and structural reaction rules. Third, the remaining full-corpus errors are transformation dependent. The same interface may enumerate one chain family exactly and fail through omission or overgeneration on another, locating the bottleneck in rule induction and exhaustive application instead of corpus visibility alone.

D HUMAN-BASELINE STUDY

Design. Three chemistry-trained volunteers completed the study. Several tasks require substantial corpus-wide computation even with tools. Each volunteer therefore received a stratified subset spanning all tiers and task families, with selected overlap for agreement measurement. The volunteers submitted 52 responses covering 43 unique questions. Tools and AI assistance were allowed, while the annotator remained responsible for checking the method and final answer. The application recorded active browser and self-reported offline time separately and autosaved work. The baseline analysis uses the 51 responses containing an answer. One "cannot answer" response is retained in completion and timing summaries but excluded from accuracy.

Scoring. We score each submitted answer against the benchmark ground truth. Exact-set match requires set equality, while precision, recall, and F1 retain partial credit for large set-valued answers. Active and offline times are not summed because they may overlap.

The two non-exact Tier-2 responses concerned a molecular-weight filter and a combined molecular-weight/ring filter; both were scored against the stated deterministic definitions, which remained unchanged. The study recorded 8.46 active browser hours, 10.71 browser wall hours, and 5.76 self-reported offline/tool hours in total; the latter can overlap with an open browser session. Median active time per submitted response increased from 3.32 minutes in Tier 1 to 7.44 minutes in Tier 4, while median offline time increased from 2 to 10 minutes (Figure 33). Mean confidence was 4.07 on a five-point scale. Tool-use records show RDKit on 51 responses, local scripts on 48, an LLM or coding agent on 45, offline processing on 13, literature or web search on 2, and consultation on 2; categories are non-exclusive.

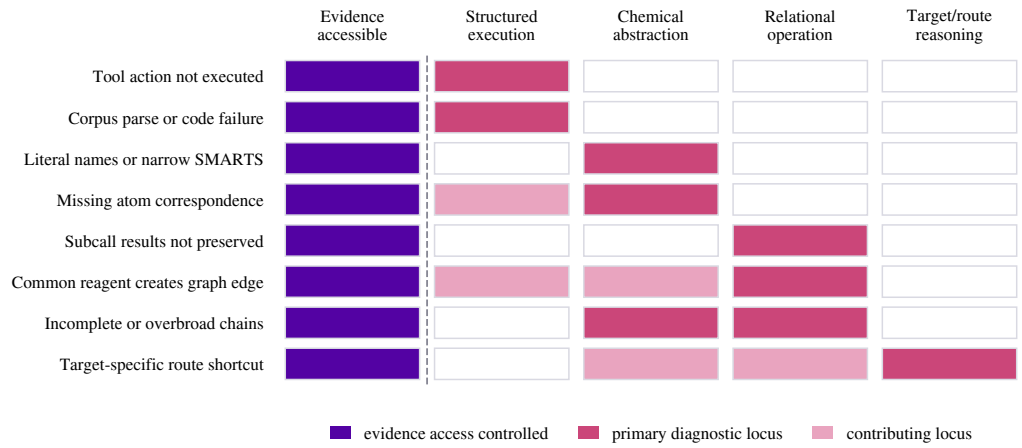

Figure 31: Trace-supported failure mechanisms mapped to the benchmark's diagnostic capabilities. The map distinguishes primary and contributing mechanisms after accounting for evidence access; it documents locations supported by the traces, not their population frequencies.

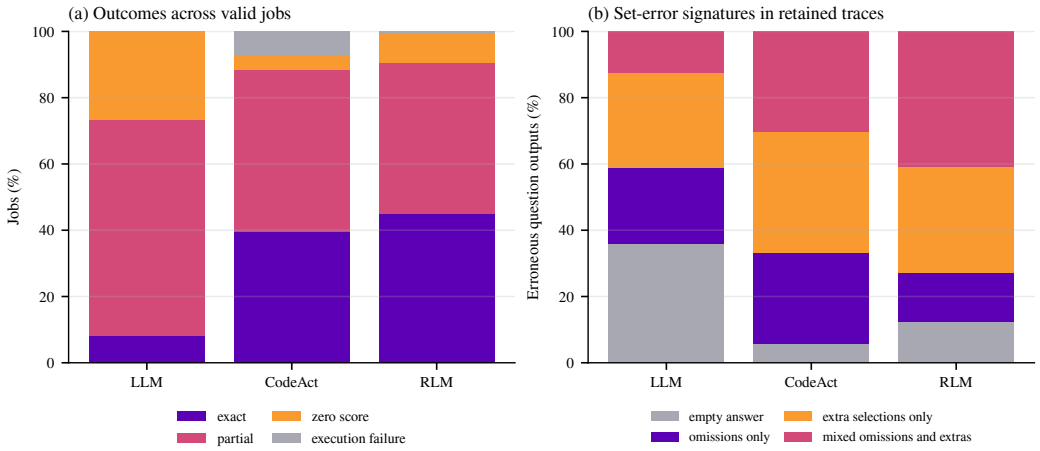

Figure 32: Failure accounting across the five paper models. (a) Outcomes for 6,179 trace-eligible jobs. (b) Set-error signatures among erroneous outputs in the 2,229-question corrected-task trace cohort. Scientific errors and execution failures are counted separately.

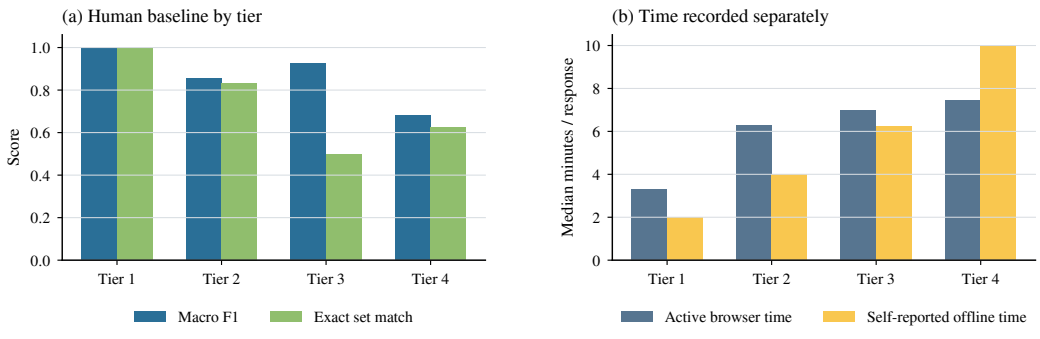

Figure 33: Human validation by tier. (a) Macro-F1 and exact-set-match accuracy. (b) Median active browser time and self-reported offline/tool time per submitted response; the two timing measures are recorded separately and are not additive.

Table 7: Representative trace-supported failure mechanisms. "Diagnostic locus" names the capability implicated after accounting for evidence access; the evidence column states the observation supporting that assignment.

Model	Interface	Task	Diagnostic locus	Direct trace evidence
Claude Haiku 4.5	CodeAct	T3.7	Structured execution	Python appears inside an unsupported action wrapper; no tool observation follows, yet the response declares the code successful and returns 55 indices with no overlap with the 102 references.
Claude Haiku 4.5	CodeAct	T3.8	Chemistry reasoning	Executed code searches SMILES for literal `HATU`, `PF6`, and `T3P`; the trace displays the structural HATU cation and hexafluorophosphate but still returns no matches.
DeepSeek V4 Flash	CodeAct	T4.5	Structured execution	An invalid RDKit import is followed by a parser that repeatedly reads zero reaction records, preventing graph traversal.
DeepSeek V4 Flash	RLM	T3.31–35	Structured execution	After RDKit errors, local yes/no validation calls are made but their decisions are not preserved in the final reaction set.
GPT-5 mini	CodeAct	T3.4	Chemistry reasoning	Protonation- and tautomer-specific SMARTS return no fused heterocycles although all 21 references are present in the context.
Gemini 3.7 Flash	CodeAct	T4.16–25	Relational operation	Trifluoroacetic acid passes a generic heavy-atom threshold and links otherwise unrelated reactions; the trace detects the spurious edge and adds an ad hoc exclusion.
Gemini 3.7 Flash	RLM	T4.31–35	Rule formulation and enumeration	Within one full-corpus run, two chain families are exact, one recovers only 1 of 37 chains, and another returns 80 candidates with zero overlap with 9 references.
Qwen3.5-397B-A17B	RLM	T4.31–35	Rule formulation and enumeration	The run returns 223 candidates for a 9-chain Suzuki family, 18 for a 4-chain Negishi family, and no valid Boc–Buchwald or three-step chain.
Qwen3.5-397B-A17B	CodeAct	T4.16–25	Target/route reasoning	A common graph procedure is exact on four targets but fails on others after target-specific chemistry conditions are introduced.

Table 8: Human-baseline performance. Overall values macro-average the 51 scored responses.

Scope	Scored responses	Exact match	Precision	Recall	F1
Tier 1	7	1.000	1.000	1.000	1.000
Tier 2	12	0.833	0.854	0.854	0.854
Tier 3	16	0.500	0.929	0.951	0.925
Tier 4	16	0.625	0.695	0.741	0.680
Overall	51	0.686	0.848	0.869	0.842

Inter-annotator agreement. The overlap yielded 11 pairwise reviewer–question comparisons across the three reviewer pairs. Raw correctness agreement ranged from 0.80 to 1.00 and pairwise Cohen's κ from 0.545 to 1.000. Across all answered overlap, nominal Krippendorff's α was 0.777 for submitted-answer identity and 0.744 for correctness. These agreement figures summarize the deliberately limited overlap and are interpreted alongside the item-level disagreement list; they are not precise population estimates. Machine-readable item metrics, timing records, tool-use counts, pairwise disagreements, and the immutable input checksums accompany the artifact.

E Benchmark Release and Contamination Policy

The anonymous supplement includes all 100 questions and reference answers, stable task metadata and prompts, answer-generation code, verification tests that regenerate the answer sets and check exact agreement with the released references, context builders and fixed sampling seeds, parsers and scorers, experiment definitions, and a version-pinned software environment. It also includes the chemistry rules and deterministic executor used in the controlled study. This documentation records provenance, composition, construction, and intended use following established dataset-documentation practice (Gebru et al., 2021). The code and benchmark will be released publicly as *RxnHaystack-Public-v1* upon acceptance.

The reaction corpus is reconstructed from the CC BY 4.0 *Reaction SMILES USPTO year 2023* dataset (van der Lingen, 2023). The artifact records the upstream DOI and attribution, cleaning procedure, pinned RDKit version, row counts, and SHA-256 file fingerprints for the raw and cleaned corpora. Matching fingerprints verify byte-identical reconstruction. Dataset files remain outside the software repository and can be reconstructed with the supplied script.

The human-evaluation component comprises the annotation application, study definitions, instructions, assignment logic, schemas, and analysis code.

Public benchmark exposure can complicate later model comparisons, although the effect depends on the training regime and degree of exposure (Sainz et al., 2023; Bordt et al., 2025). For contamination-sensitive evaluation of future models, we will create a distinct *RxnHaystack-Heldout* stream from new targets, thresholds, reaction classes, task parameters, and sampling seeds selected after the public release. Its versioned instances and answers will remain private and be evaluated through a hosted service. Papers should identify the exact public version or held-out stream used, and the two scores will be reported separately.