UREA: READ LIKE A CHEMIST, COMMIT LIKE A COMPILER

Anonymous authors
Paper under double-blind review

ABSTRACT

The chemical literature is written to be read by chemists: reactions are distributed across schemes, tables, captions, and procedures, and faithful extraction has resisted automation. UREA (UREA Reaction Extracting Agent) is an agentic framework for such multimodal reading: a general-purpose vision-language model cross-references figures with procedures, zooms into regions at will, and maintains a compound ledger, while a deterministic chemistry compiler resolves shared definitions into structured records. The workflow combines structural validation, one-label-one-structure authoring, count reconciliation, and evidence pointers. Reading stays permissive; commitment makes its decisions explicit and checkable. Count reconciliation addresses silent truncation, where whole-document reading returns well-formed output covering only a fraction of a paper's reactions: the agent inventories experiments up front, reconciles its records against that inventory, and revisits shortfalls. We also audit and correct the benchmark's reference annotations; UREA achieves state-of-the-art 89.73 hard-match F1 on the corrected whole-document chemical reaction benchmark. The design transfers across model families and agent harnesses, and the largest observed performance swing in the ablations follows a change in reasoning effort. Adapted through task-specific prompts to standalone molecular recognition, the reader outperforms MinerU.Chem under author review on 500 development images. Together, these results motivate a design pattern that combines flexible reading with explicit, checkable records for domains where structured knowledge must stay faithful to source documents.

1 INTRODUCTION

Three rows in a reaction scheme can describe five experiments. Two rows in a table can report the same yield yet select different substrates and products. An intermediate can appear in both an executed control and a proposed mechanism (Yang et al., 2023; Lindman et al., 2023; Zhang et al., 2023). Recognizing every molecule still leaves the central question unanswered: *what was actually done?* Recovering a reaction dataset requires interpreting these relationships across schemes, tables, captions, and procedures.

We present UREA (UREA Reaction Extracting Agent)[1], an agentic framework organized around one principle: *maintain a shared, compilable reaction workspace as the paper is read.* A vision-language agent decides how the evidence defines each experiment and records its decisions as editable chemical definitions and reaction blocks. A deterministic compiler resolves their references into structured records. Feedback from compilation, chemical checks, visual comparison, and coverage review guides the agent's next inspection or edit. The records are the working objects through which extraction proceeds (Figure 1).

This organization makes related decisions reusable and revisable. Each compound label has one shared definition; a local correction updates every referring record on recompilation. Shared scaffolds support substituent series, while visual checks and source inventories guide focused revision. The model interprets evidence; tools make construction and checking repeatable.

[1]The name recalls Wöhler's 1828 synthesis of urea (Wöhler, 1828).

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.

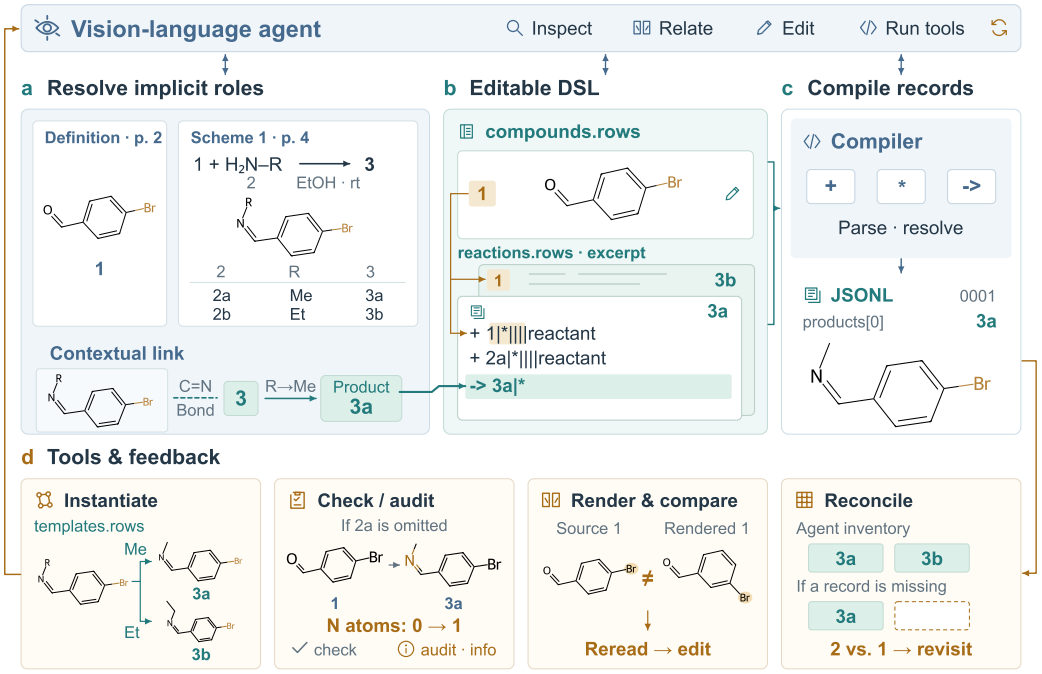


Figure 1: **Read like a chemist, commit like a compiler.** A single agent connects paper evidence to an editable reaction workspace. It resolves implicit roles from context (a), maintains shared compound definitions and reaction blocks (b), and compiles them into records (c). Tools support template instantiation, chemical auditing, visual comparison, and inventory reconciliation (d). The audit highlights an unexplained nitrogen atom if amine **2a** is omitted.

Prior work establishes strong foundations for this approach. Chemical information extraction has advanced from domain-specific text mining to structured extraction with language models (Swain & Cole, 2016; Dagdelen et al., 2024; Schilling-Wilhelmi et al., 2025). Multimodal systems connect figures and procedures (Fan et al., 2024; Leong et al., 2025; Li et al., 2026), and agentic readers use shared dictionaries and memory (Chen et al., 2025a;c). Our focus is the interface between reading and record construction: a shared representation that the agent can edit, compile, inspect, and revise throughout one session.

On a 19-article benchmark with audited references, UREA achieves 89.73 F1 versus 62.53 for direct reading with the same model, harness, and reasoning setting, with comparable precision and far higher recall. Three real cases show how the reader resolves experiment boundaries and branch assignments. Further studies examine which workflow instructions affect extraction and how the approach extends across input scopes and model–harness deployments.

Our contributions are:

- **A compilable workspace for agentic reaction extraction.** We organize multimodal reading around shared chemical definitions and editable reaction blocks, connecting local revision and deterministic record construction through source-guided feedback.

- **An evaluation of extraction quality and reading decisions.** We reconcile benchmark references with source experiments and connect extraction scores to concrete reading decisions through same-model comparisons and case studies.

2 Related Work

Multimodal document extraction. OpenChemIE aligns conditions, resolves molecular coreferences, and reconstructs substrate-scope variants across text, tables, and figures (Fan et al., 2024). MERMES applies multimodal models to electrosynthesis mining (Leong et al., 2024); MERMaid combines footnote-aware record updates with schema-guided graph construction (Leong et al., 2025). ReactionSeek integrates prompted extraction with chemical name resolution and standardization (Li et al., 2026). Shared state and revision also feature prominently in agentic systems: ChemMiner describes a shared coreference dictionary with conflict-driven revisiting (Chen et al., 2025a), and ChemEAGLE combines shared-memory specialist agents with planning and action observers (Chen et al., 2025c). AgentCAT uses evidence collection and review-driven re-extraction for catalytic-reaction data (Yang et al., 2026c). UREA makes shared chemical definitions and reaction blocks directly compilable: the agent edits its reading decisions in place, and reference resolution propagates those edits across the records.

Structured and executable representations. LLM-NERRE extracts scientific entities and relations into user-defined schemas (Dagdelen et al., 2024). Code4Struct and GoLLIE use Python-based schemas and generated instances for information extraction (Wang et al., 2023; Sainz et al., 2024). CMDL represents experimental data in a chemical DSL with compiler checks and JSON export (Park et al., 2023). PICARD and grammar-constrained decoding check admissible continuations during generation (Scholak et al., 2021; Geng et al., 2023); RxnScribe similarly enforces a reaction-sequence grammar (Qian et al., 2023a). Logic-LM revises formalizations using solver feedback (Pan et al., 2023). In chemistry, CLAIRify iteratively repairs generated XDL with verifier feedback (Yoshikawa et al., 2023); ACRA combines XDL validation with source–procedure comparison and execution simulation (Pagel et al., 2026).

Agents, execution feedback, and interfaces. ReAct (Yao et al., 2023) interleaves reasoning and actions, and CodeAct (Wang et al., 2024) uses executable code as a compositional action space. ChemCrow and Coscientist connect tool-using agents to chemical reasoning and experimentation (Bran et al., 2024; Boiko et al., 2023). SWE-agent emphasizes file-editing interfaces (Yang et al., 2024); Self-Debugging and CRITIC use execution or tool feedback to guide revisions (Chen et al., 2024; Gou et al., 2024). Visual feedback also has precedents: Design2Code compares rendered outputs with target screenshots during revision (Si et al., 2025), and VERDICT studies re-rendering and cross-recognizer agreement for chemical structure verification (Guan et al., 2026). In UREA, these feedback channels act on the same reaction workspace throughout reading: a diagnostic or visual discrepancy can lead to a local edit, further source inspection, and recompilation.

Recognition and symbolic reconstruction. DECIMER (Rajan et al., 2023) and MolScribe (Qian et al., 2023b) recover molecular structures from images; ReactionDataExtractor and its successor recover entities and their connections in reaction schemes (Wilary & Cole, 2021; 2023). RxnScribe instead generates reaction diagrams as structured sequences (Qian et al., 2023a), while RxnIM uses a multimodal language model for reaction-image parsing (Chen et al., 2025b). ChemSchematicResolver and RGReco connect scaffold recognition with labels and substituent definitions (Beard & Cole, 2020; Xiang et al., 2026); MarkushGrapher-2 jointly predicts chemical backbones and substituent tables (Strohmeyer et al., 2026). Recent LVLM captioning strategies formulate structure recovery as generating image-grounded molecular descriptions rather than predicting coordinates: RxnCaption reformulates reaction diagram parsing as captioning guided by bbox-and-index visual prompts (Song et al., 2026). UREA's render-back inspection is complementary—it re-renders the agent's own structured claims for visual audit instead of relying on an external detector, keeping the verification loop inside the compilable workspace. MinerU.Chem links recognized molecules and reactions through identifiers and source locations (Yang et al., 2026b).

Chemical text and reaction extraction. ChemicalTagger parses experimental prose into chemical entities and actions (Hawizy et al., 2011), while ChemDataExtractor organizes chemical information from scientific documents (Swain & Cole, 2016). Vaucher et al. (2020) translate experimental prose into compact action sequences that convert losslessly to structured objects. Learned reaction extractors include ChemRxnExtractor (Guo et al., 2022) and ReactIE, which combines question answering with weakly supervised pretraining (Zhong et al., 2023a). Reaction Miner adds PDF

text processing, reaction-aware segmentation, and an expanded set of reaction roles (Zhong et al., 2023b). LLM-based methods recover structured fields from synthesis procedures (Ai et al., 2024) and patents (Vangala et al., 2024); model adaptation and entity self-grounding further refine textual extraction (Kumari & Garg, 2025). At industrial scale, patent-derived reaction databases pair automated extraction with consistency auditing and atom-mapping validation (DianShi-RxnDB; ChEMU; the USPTO-derived corpora) (Wang et al., 2026; He et al., 2020; Lowe, 2017); these pipelines validate records after the fact at corpus scale, whereas UREA performs verification *during* reading at document scale—the two are complementary stages of a curation pipeline, and a compilable workspace of the kind UREA maintains is a natural high-fidelity front end to such databases. Related applications include CatMiner's prompted extraction of heterogeneous-catalysis data (Walls & Linic, 2025) and MaterialBrain's few-shot extraction of material synthesis routes (Yang et al., 2026d). These methods establish the textual foundation for recovering experimental records; our task also requires interpreting the structures and relationships distributed across figures and tables.

3 METHOD

UREA treats extraction as building and revising a compilable chemical workspace. The reader expresses its interpretations as shared compound definitions and experiment-specific reaction blocks. Tools turn these decisions into structured records and actionable feedback (Figure 1).

3.1 TASK AND WORKSPACE OVERVIEW

Given a paper D, we recover the executed reactions in its main-text PDF, including schemes, tables, captions, and experimental descriptions. Optimization trials, failures, and scale-up experiments remain distinct; proposed mechanisms and retrosynthetic plans are excluded. Each record preserves ordered operations and their reported products; an explicitly isolated intermediate establishes a new record boundary.

The workspace $W = (L, T, B)$ separates compound identities in ledger L, reusable scaffolds in template ledger T, and experiment-specific reaction blocks B. A single persistent agent session maintains these files as it reads. Algorithm 1 summarizes its action choices: evidence and feedback determine whether to inspect a source, edit a definition, construct a structure, or check the current records. For example, resolving a scope entry may require joining a substituent label with a scaffold and a caption elsewhere on the page.

3.2 A COMPILABLE REACTION LANGUAGE

We use a line-oriented domain-specific language (DSL) to separate chemical authoring from data exchange (Park et al., 2023; Vaucher et al., 2020). Its row prefixes make an experimental interpretation directly editable: `@` starts a source-linked record, `+` introduces a component, `~` supplies conditions, and `->` names a product. The separator `--` preserves ordered operations within one experiment. The following rows encode the reaction in Figure 1:

Compound definitions L

```
1 | | O=Cc1ccc(Br)cc1
2a | | CN
3a | | CN=Cc1ccc(Br)cc1
```

Reaction block B

```
@ scheme1 page:4
+ 1|*||||reactant
+ 2a|*||||reactant
+ |CCO|EtOH|||solvent
~ temp: rt
-> 3a|*
```

Each component distinguishes its label, SMILES (Weininger, 1988), and name from its experiment-specific amount, concentration, and roles. Conditions, yields, and selectivity retain the source wording; unsupported fields stay empty. Thus, adding a reagent or changing a condition is a local row edit, while the compiler handles serialization.

Algorithm 1 Evidence-guided reaction compilation

Require: Paper D
Initialize workspace $W = (L, T, B)$ and observation history H
repeat
  $a \leftarrow \mathrm{Reader}(D, W, H)$
  Dispatch the selected action a:
    INSPECT: read a selected source region
    EDIT: revise compound definitions, templates, or reaction blocks
    INSTANTIATE: construct structures from a template and substitution map
    COMPILE: resolve (L, B) into records or return diagnostics
    REVIEW: inspect structural, visual, or inventory feedback
  Retain observations and diagnostics in H
until the reader completes final review and inventory reconciliation
return compiled reaction records

3.3 SHARED DEFINITIONS AND CHEMICAL CONSTRUCTION

Resolve once, reuse across records. A structure cell contains either an explicit SMILES s or a $\star$ reference through its label ℓ:

$$\mathrm{resolve}_L(\ell, s) = \begin{cases} L[\ell].\mathrm{smiles}, & s = \star, \\ s, & \text{otherwise.} \end{cases} \tag{1}$$

Compilation $\mathcal{C}(L, B)$ applies this rule, checks grammar and roles, and emits structured records. Duplicate ledger labels and unresolved references return diagnostics identifying the offending definition or row. All references to compound **1** therefore share one structure; revising its definition updates every referring occurrence on recompilation. Shared definitions make cross-record consistency a property of the representation.

Construct paired structures from shared scaffolds. For substrate and product templates T_s, T_p and a row-specific substitution map σ_i, the instantiation tool constructs

$$\big(G_i^s, G_i^p\big) = \big(\mathrm{Inst}(T_s, \sigma_i), \mathrm{Inst}(T_p, \sigma_i)\big). \tag{2}$$

The reader identifies the scaffold, attachment sites, and fragments from the source (Beard & Cole, 2020; Xiang et al., 2026); the tool joins the graphs, removes placeholders, and sanitizes the molecules. Using the same σ_i keeps substitutions aligned across the pair. The reader writes these structures into the workspace, or reads an irregular entry directly when its scaffold differs.

3.4 EVIDENCE-GUIDED REFINEMENT

Feedback identifies what to inspect or edit next (Chen et al., 2024). Three complementary views connect the current records back to their source.

Structural and cross-record feedback. RDKit-based validation (Landrum et al., 2006) checks molecular parseability; an audit flags conflicting identities, suspicious repetitions, and atom- or scaffold-retention anomalies. The reader revisits the implicated evidence and edits the relevant definition or record, retaining legitimate repeats such as optimization trials.

Visual feedback. Render-back sheets expose attachment positions and other structural details for comparison with the source. The protocol directs the reader to compare these labeled structures with the original figure, localize discrepancies, and revise the workspace before recompiling.

Coverage feedback. For each source group g, the reader compares its inventory n_g^{inv} with the emitted count n_g^{out}. A mismatch directs attention back to that figure or table to reconcile the experiments. The protocol ends with a final review and per-source count summary. Appendix A specifies the interfaces.

Table 1: **Reaction extraction across three input scopes.** F1 (%) on final annotations: D1 matches reaction structures; D2 also checks drawn conditions; D3 checks reference-required stereochemistry. All 1,198 / 2,985 / 1,007 references remain in the denominators. †Vision-language backbone replaced by GPT-5.6-Luna. OpenChemIE uses its task-specific PDF/image pipeline; — denotes not evaluated.

System	Configuration	D1: documents	D2: figures	D3: R-groups
OpenChemIE	Released models	18.14	7.13	0.00
ChemEAGLE	GPT-5.6-Luna backbone†	28.34	21.44	33.25
MERMaid	GPT-5.6-Luna backbone†	0.00	4.61	0.00
Direct reading	GPT-5.6-Luna / Codex	62.53	—	—
UREA	GPT-5.6-Luna / Codex	**89.73**	**75.24**	**73.98**

4 EVALUATION

We first evaluate whole-document extraction, examine the reading decisions behind the results, and then test the approach across input scopes and deployment settings.

4.1 EXPERIMENTAL SETUP

Document-level benchmark. Our primary benchmark uses 19 main-text chemistry PDFs from OpenChemIE's document dataset, subsequently used by ChemEAGLE (Fan et al., 2024; Chen et al., 2025c). Extraction covers the executed reactions defined in Section 3.1. The reference unit is an executed experiment: repeated depictions share a reference, whereas separately executed optimization trials remain distinct. We therefore reconcile the supplied database records with the articles, correcting reaction boundaries and structures and recovering omitted experiments. All 19 PDFs are retained, yielding a common inventory of 1,198 executions from the original 2,149 entries (Appendix C).

Metric. ChemEAGLE's released matcher pairs reactions one-to-one using canonicalized reactant and product multisets; we report corpus-level precision, recall, and F1 (%). Matching removes stereochemistry; absent reference conditions make soft and hard scores identical. Missing outputs remain in the reference denominator. Structural matching measures structures and coverage; D2 additionally scores drawn condition molecules. The multimodal benchmarks used here do not jointly score yields, operation order, and free-text condition bindings. We therefore pair structural scores with source-linked case studies of experiment boundaries, reagent selection, and operation order (Section 4.3); Appendix C.1 distinguishes this scope from text-based field benchmarks. Adapter choices are fixed across all systems (Appendix C).

Systems and execution. The default configuration uses GPT-5.6-Luna in Codex with max reasoning effort and a four-hour limit per document. Same-model comparisons retain this configuration unless stated otherwise. *Direct reading* asks the model to traverse the document and write JSON records with a minimal schema. We rerun the released OpenChemIE, ChemEAGLE, and MERMaid implementations (Fan et al., 2024; Chen et al., 2025c; Leong et al., 2025); OpenChemIE retains its models, and ChemEAGLE and MERMaid use GPT-5.6-Luna for vision-language calls. Appendix C specifies output selection and implementation details.

4.2 WHOLE-DOCUMENT EXTRACTION

UREA achieves 89.73 F1, improving on no-tool direct reading by 27.21 points (Table 1). The improvement is recall-dominated: UREA recovers 1,097 of 1,198 references from 1,247 predictions, with 87.97 precision and 91.57 recall. All configurations use the same reference denominator; Appendix D reports complete counts and same-model comparisons.

Variation across articles. UREA wins on 13 of 19 papers; a whole-paper paired bootstrap gives a 95% interval of [11.62, 49.87] points for the gain (Appendix D.1).

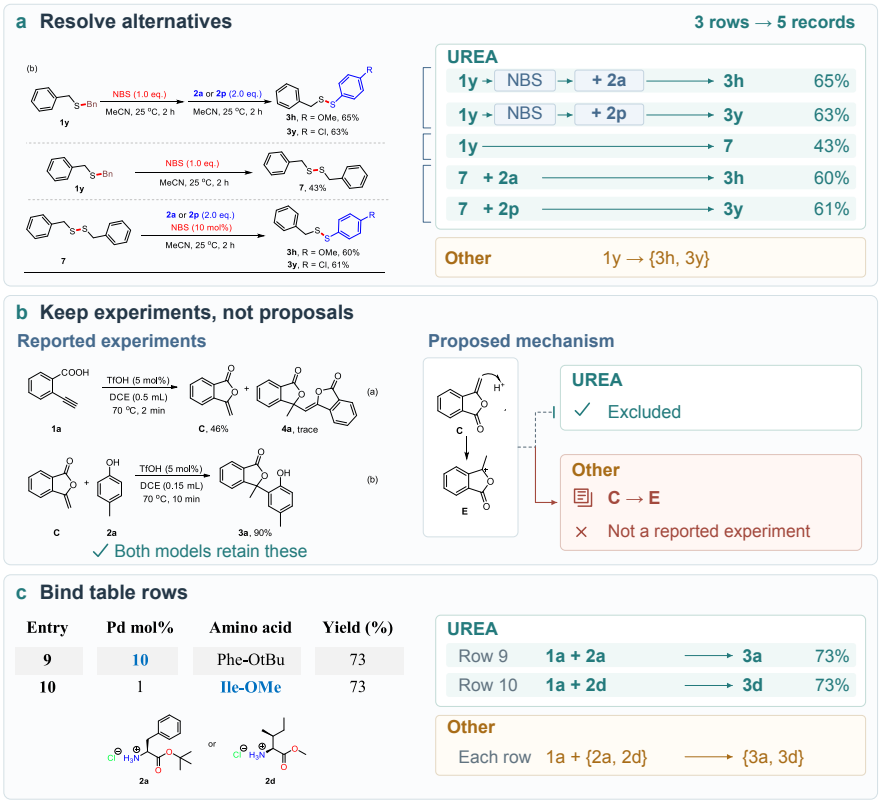


Figure 2: **Reading decisions beyond molecular recognition.** Source excerpts and label-level outputs: (a) three rows specify five experiments; boxes order additions within each record (Yang et al., 2023); (b) both models retain the controls, but ChemEAGLE also exports the proposed **C**→**E** arrow as an extra experiment (Zhang et al., 2023); (c) rows select different partners and products (Lindman et al., 2023). "Other" denotes ChemEAGLE. "Excluded" means omitted from experimental records. Conditions are abbreviated; displayed table fields are rearranged, with both Pd and ligand loadings changing. Braces denote jointly listed species. Appendix F provides source context and OpenChemIE comparisons.

Among the external reruns, ChemEAGLE obtains 28.34 F1 and OpenChemIE obtains 18.14 F1. These reruns evaluate complete-structure extraction under the stated configurations; MERMaid's 271 delivered entries have no reaction structures and receive zero structural matches. Appendix C separates published native-configuration scores from our reruns and documents reference, backbone, adapter, and output differences (Table 8).

4.3 Case studies: deciding what was actually done

Figure 2 compares actual outputs for the three introductory examples.

(a) Separate alternatives while preserving operations. Three scheme rows mix sequential operations, alternative partners, and a separately isolated intermediate. UREA recovers five experiments, preserving the sequential operations. Its first two records retain NBS addition before the source-selected second partner. ChemEAGLE merges alternative products; OpenChemIE leaves the branches unresolved.

(b) Determine which transformations were executed. Compound **C** is isolated experimentally and used in a proposed mechanism; both UREA and ChemEAGLE retain the two executed controls, but ChemEAGLE also exports ten proposed transformations.

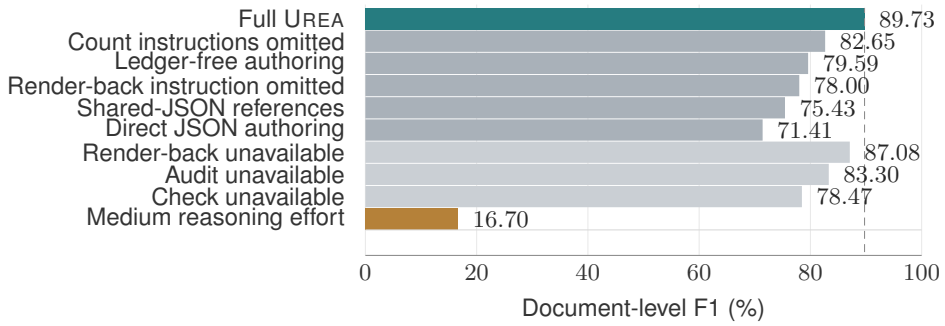


Figure 3: **Extraction is sensitive to workflow instructions and tool availability.** Dark gray bars vary the written instructions with tool access fixed; light gray bars keep the full prompt and block one named tool entrypoint; the gold bar separately changes reasoning effort from max to medium. All use the same 19-document benchmark and full reference denominator.

(c) Bind each row to its selected branch. UREA binds each of the 17 table rows to its selected pair of alternative amino acids and products, whereas ChemEAGLE lists both alternatives in every row and OpenChemIE returns a single header record. Rows 9 and 10 both report 73% yield but differ in partner and Pd loading; UREA represents them as separate, source-linked experiments. The workspace makes these decisions explicit: separate reaction blocks, ordered operations, and references to selected compounds.

4.4 SENSITIVITY TO WORKFLOW INSTRUCTIONS

Instructions, representation, and runtime availability. Figure 3 varies the written instructions (dark bars) and blocks named tool entrypoints at runtime with the full prompt unchanged (light bars). Plain nested JSON reaches 71.41 F1 and shared-reference JSON reaches 75.43, compared with 89.73 for the full row/ledger workflow. Ledger-free authoring reaches 79.59, omitting render-back instructions 78.00, and omitting count reconciliation 82.65 (Table 9). These comparisons characterize the representation together with its authoring instructions.

Runtime removal is not instruction omission. Blocking the check entrypoint yields 78.47 F1; blocking audit yields 83.30. The render-back entrypoint intervention reaches 87.08, compared with 78.00 when its instruction is omitted. The distinct outcomes motivate examining the agent's executed workflow alongside the requested intervention; Appendix D.1 reports the observed tool returns.

Reasoning effort is a separate execution choice. Keeping the full protocol but changing Codex's reasoning setting from max to medium lowers F1 to 16.70. This exceeds every instruction-level difference: the execution setting matters alongside the written protocol.

4.5 FIGURE EXTRACTION AND R-GROUP RESOLUTION

Tasks and scoring. D2 contains 324 figures from ChemEAGLE's research, OpenChemIE, and review subsets, with 2,985 audited reaction references (Chen et al., 2025c). Hard matching requires reactants, products, and condition molecules designated as drawn in the reference. D3 contains 78 R-group figures and 1,007 instantiated reactions from OpenChemIE (Fan et al., 2024). Both use maximum one-to-one matching and full denominators; the datasets overlap and are reported separately (Appendix D.2).

Recovering complete reaction series. With GPT-5.6-Luna/Codex max, UREA reaches 75.24 hard F1 on D2 and 73.98 instance F1 on D3 (Table 1). UREA leads each D2 source subset among the evaluated runs (Figure 4a). Its D2 structure-only score is 78.47 (Table 12); D3 additionally tests expansion of shared R-group drawings into complete structures (Table 13).

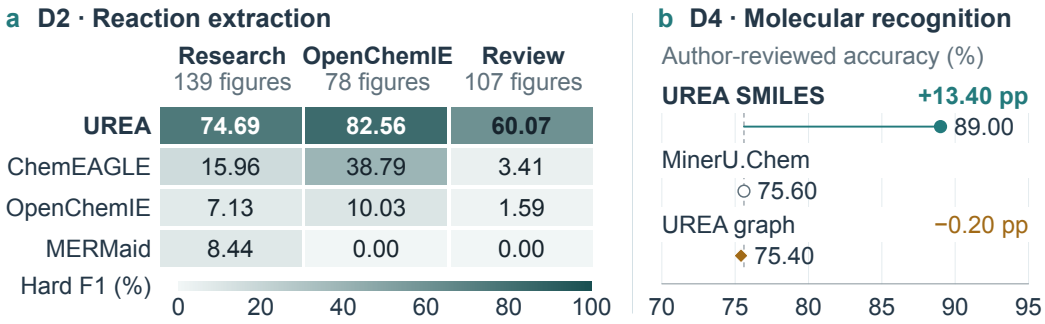


Figure 4: **Reading chemistry across tasks.** (a) D2 hard F1; 1,403 / 1,085 / 497 reference reactions. (b) Author-reviewed accuracy on 500 D4 development images; missing outputs count as errors. Dashed line: MinerU.Chem; gaps: percentage points.

Table 2: **Six high-reasoning deployments on D1.** Docs counts usable outputs out of 19; every row retains the 1,198-reference denominator. Each deployment uses its harness-native reasoning controls.

Model	Harness	Effort setting	Docs	P	R	F1
GPT-5.6-Luna	Codex	max	19	87.97	**91.57**	**89.73**
DeepSeek V4.1 Flash	Codex	max	18	**91.64**	75.96	83.07
DeepSeek V4.1 Flash	Claude Code	31k budget	19	80.34	70.62	75.17
DeepSeek V4.1 Flash	pi	high	17	91.49	71.79	80.45
Qwen3.8-Flash	Claude Code	31k budget	19	77.75	72.04	74.78
Qwen3.8-Flash	pi	high	19	88.77	79.22	83.72

4.6 Beyond reaction extraction: molecular recognition

For D4, a task-specific SMILES reader retains image-grounded interpretation and optional local cross-checks. On 500 MolRecBench-Wild development images (Yang et al., 2026a), author-reviewed accuracy is 89.00% for the SMILES reader versus 75.60% for MinerU.Chem (Yang et al., 2026b) (Figure 4b; the graph-output variant appears in Appendix E). The SMILES reader recovers 445 versus 378 structures (13.40 points, missing outputs counted as errors), extending the reader's utility from experiment extraction to molecular digitization; Appendix E details prompts, sampling, and review.

4.7 Deployment across models and harnesses

Across three models and three harnesses (Table 2), DeepSeek V4.1 Flash reaches 83.07 F1 with Codex and Qwen3.8-Flash 83.72 with pi.

5 Conclusion

Urea turns chemical literature reading into the construction of an editable, compilable workspace. Shared definitions, ordered reaction blocks, and evidence-guided checks connect multimodal interpretation to reusable records. Whole-document results, source-linked cases, and cross-model deployments demonstrate the utility of this organization; molecular-recognition studies extend its image-grounded reading strategy.

AI USE STATEMENT

We used generative AI assistants for conceptual framing, hypothesis development and refinement, methodology and experimental-design feedback, toolchain and evaluation-adapter implementation, reference-data auditing and reformatting, error analysis, and result interpretation. They also assisted with literature search and summarization, Chinese–English translation, drafting and editing, reference organization, and preparation of figures and tables. Checks included unit tests, executed evaluation pipelines, comparison of cited claims with primary sources, and regeneration of quantitative artifacts from frozen snapshots. The authors retain responsibility for the text, claims, data, and artifacts.

ETHICS STATEMENT

The workspace retains source pointers to support inspection and correction of extracted chemical knowledge. Experimental reuse requires verification against the original articles and expert assessment of chemical safety, feasibility, and dual-use risks. Reuse and redistribution of source articles, figures, and benchmark materials should respect their licenses and access terms.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A specify record semantics, compilation, and tool interfaces. Appendices B–E document prompts, execution settings, reference construction, scoring, output selection, and analyses. Quantitative figures and tables are generated from frozen snapshots with reference and prediction fingerprints. The primary backbone (GPT-5.6-Luna) is served through a public subscription endpoint, and the workflow runs end-to-end on other public models as well (DeepSeek V4.1 Flash and Qwen3.8-Flash via their public APIs), where UREA retains 83–84 F1 (Table 2); Table 6 maps every display name to its recorded API identifier. The deterministic toolchain (compilation, chemistry, audit, render-back) is model-independent.

REFERENCES

Qianxiang Ai, Fanwang Meng, Jiale Shi, Brenden Pelkie, and Connor W. Coley. Extracting structured data from organic synthesis procedures using a fine-tuned large language model. *Digital Discovery*, 3(9):1822–1831, 2024. doi: 10.1039/D4DD00091A.

Alibaba Cloud. Model Studio: Model lifecycle and updates, 2026. URL https://www.alibabacloud.com/help/en/model-studio/newly-released-models. August 26 model-service launch entry; accessed September 23, 2026.

Edward J. Beard and Jacqueline M. Cole. ChemSchematicResolver: A toolkit to decode 2d chemical diagrams with labels and R-groups into annotated chemical named entities. *Journal of Chemical Information and Modeling*, 60(4):2059–2072, 2020. doi: 10.1021/acs.jcim.0c00042. URL https://pubs.acs.org/doi/10.1021/acs.jcim.0c00042.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8. arXiv:2304.05376.

Kexin Chen, Yuyang Du, Junyou Li, Hanqun Cao, Menghao Guo, Xilin Dang, Lanqing Li, Jiezhong Qiu, Guangyong Chen, and Pheng Ann Heng. ChemMiner: A large language model agent system for chemical literature data mining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 7654–7662, 2025a. URL https://openaccess.thecvf.com/content/ICCV2025W/VisionDocs/html/Chen_ChemMiner_A_Large_Language_Model_Agent_System_for_Chemical_Literature_ICCVW_2025_paper.html.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2304.05128.

Yufan Chen, Ching Ting Leung, Jianwei Sun, Yong Huang, Linyan Li, Hao Chen, and Hanyu Gao. Towards large-scale chemical reaction image parsing via a multimodal large language model. *Chemical Science*, 16:21464–21474, 2025b. doi: 10.1039/D5SC04173B.

Yufan Chen, Ching Ting Leung, Bowen Yu, Jianwei Sun, Yong Huang, Linyan Li, Hao Chen, and Hanyu Gao. A multi-agent system enables versatile information extraction from the chemical literature. *arXiv preprint arXiv:2507.20230*, 2025c.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15:1418, 2024. doi: 10.1038/s41467-024-45563-x.

DeepSeek. DeepSeek API change log, 2026. URL `https://api-docs.deepseek.com/updates/`. September 10 release entry; accessed September 23, 2026.

Vincent Fan, Yujie Qian, Alex Wang, Amber Wang, Connor W. Coley, and Regina Barzilay. OpenChemIE: an information extraction toolkit for chemistry literature. *Journal of Chemical Information and Modeling*, 64(14):5521–5534, 2024. doi: 10.1021/acs.jcim.4c00572. arXiv:2404.01462.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10932–10952. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.674. URL `https://aclanthology.org/2023.emnlp-main.674/`. arXiv:2305.13971.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.11738.

Yani Guan, Dengpan Dong, Shuang Luo, Zi Wei, Joah Han, Dan Hannah, Yumin Zhang, Qichao Hu, and Kang Xu. VERDICT: Agreement beats pixel-space verification in real-document OCSR. *arXiv preprint arXiv:2608.22183*, 2026. URL `https://arxiv.org/abs/2608.22183`.

Jiang Guo, A. Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W. Coley, Klavs F. Jensen, and Regina Barzilay. Automated chemical reaction extraction from scientific literature. *Journal of Chemical Information and Modeling*, 62(9):2035–2045, 2022. doi: 10.1021/acs.jcim.1c00284.

Lezan Hawizy, David M. Jessop, Nico Adams, and Peter Murray-Rust. ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, 3:17, 2011. doi: 10.1186/1758-2946-3-17.

Jiayuan He, Dat Quoc Nguyen, Saber A. Akhondi, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Zubair Afzal, Zenan Zhai, Biaoyan Fang, Hiyori Yoshikawa, Ameer Albahem, Lawrence Cavedon, Trevor Cohn, Timothy Baldwin, and Karin Verspoor. Overview of ChEMU 2020: named entity recognition and event extraction of chemical reactions from patents. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction (CLEF 2020), LNCS*, pp. 237–254, 2020. doi: 10.1007/978-3-030-58219-7_18.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, et al. PubChem 2025 update. *Nucleic Acids Research*, 53(D1):D1516–D1525, 2025. doi: 10.1093/nar/gkae1059.

Madhavi Kumari and Prabha Garg. Enhanced chemical reaction mining through combined model adaptation and prompt engineering. *Journal of Chemical Information and Modeling*, 65(19):10348–10363, 2025. doi: 10.1021/acs.jcim.5c02011.

Greg Landrum et al. RDKit: Open-source cheminformatics. `https://www.rdkit.org/`, 2006.

Shi Xuan Leong, Sergio Pablo-García, Zijian Zhang, and Alán Aspuru-Guzik. Automated electrosynthesis reaction mining with multimodal large language models (MLLMs). *Chemical Science*, 15(43):17881–17891, 2024. doi: 10.1039/D4SC04630G.

Shi Xuan Leong, Sergio Pablo-García, Brandon Wong, and Alán Aspuru-Guzik. MERMaid: universal multimodal mining of chemical reactions from PDFs using vision-language models. *Matter*, 8(12):102331, 2025. doi: 10.1016/j.matt.2025.102331.

Jiawei Li, Minzhou Li, Qi Yang, and Sanzhong Luo. ReactionSeek: LLM-powered literature data mining and knowledge discovery in organic synthesis. *Nature Communications*, 17:3356, 2026. doi: 10.1038/s41467-026-70180-1.

Jens Lindman, Anubha Yadav, Johan Gising, and Mats Larhed. Two-chamber aminocarbonylation of aryl bromides and triflates using amino acids as nucleophiles. *The Journal of Organic Chemistry*, 88:12978–12985, 2023. doi: 10.1021/acs.joc.3c00972.

Daniel M. Lowe. Chemical reactions from US patents (1976-Sep2016). figshare dataset, 2017.

OpenAI. API changelog, 2026. URL `https://developers.openai.com/api/docs/changelog`. July 9 release entry; accessed September 23, 2026.

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

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.248.

Nathaniel H. Park, Matteo Manica, Jannis Born, James L. Hedrick, Tim Erdmann, Dmitry Yu. Zubarev, Nil Adell-Mill, and Pedro L. Arrechea. Artificial intelligence driven design of catalysts and materials for ring opening polymerization using a domain-specific language. *Nature Communications*, 14:3686, 2023. doi: 10.1038/s41467-023-39396-3.

Yujie Qian, Jiang Guo, Zhengkai Tu, Connor W. Coley, and Regina Barzilay. RxnScribe: a sequence generation model for reaction diagram parsing. *Journal of Chemical Information and Modeling*, 63(13):4030–4041, 2023a. doi: 10.1021/acs.jcim.3c00439. arXiv:2305.11845.

Yujie Qian, Jiang Guo, Zhengkai Tu, Zhening Li, Connor W. Coley, and Regina Barzilay. MolScribe: robust molecular structure recognition with image-to-graph generation. *Journal of Chemical Information and Modeling*, 63(7):1925–1934, 2023b. doi: 10.1021/acs.jcim.2c01480. arXiv:2205.14311.

Kohulan Rajan, Henning Otto Brinkhaus, M. Isabel Agea, Achim Zielesny, and Christoph Steinbeck. DECIMER.ai: an open platform for automated optical chemical structure identification, segmentation and recognition in scientific publications. *Nature Communications*, 14:5045, 2023. doi: 10.1038/s41467-023-40782-0.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Y3wpuxd7u9`. arXiv:2310.03668v5.

Mara Schilling-Wilhelmi, Martiño Ríos-García, Sherjeel Shabih, María Victoria Gil, Santiago Miret, Christoph T. Koch, José A. Márquez, and Kevin Maik Jablonka. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews*, 54(3):1125–1150, 2025. doi: 10.1039/D4CS00913D. arXiv:2407.16867.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9895–9901. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.779. arXiv:2109.05093.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2Code: Benchmarking multimodal code generation for automated front-end engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3956–3974. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.199. URL https://aclanthology.org/2025.naacl-long.199/.

Jiahe Song, Chuang Wang, Bowen Jiang, Yinfan Wang, Hao Zheng, Fan Li, et al. Rxncaption: Reformulating reaction diagram parsing as visual prompt guided captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2511.02384; code and dataset: github.com/opendatalab/RxnCaption.

Tim Strohmeyer, Lucas Morin, Gerhard Ingmar Meijer, Valéry Weber, Ahmed Nassar, and Peter Staar. MarkushGrapher-2: End-to-end multimodal recognition of chemical structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2603.28550.

Matthew C. Swain and Jacqueline M. Cole. ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016. doi: 10.1021/acs.jcim.6b00207.

Sarveswara Rao Vangala, Sowmya Ramaswamy Krishnan, Navneet Bung, Dhandapani Nandagopal, Gomathi Ramasamy, Satyam Kumar, Sridharan Sankaran, Rajgopal Srinivasan, and Arijit Roy. Suitability of large language models for extraction of high-quality chemical reaction dataset from patent literature. *Journal of Cheminformatics*, 16:131, 2024. doi: 10.1186/s13321-024-00928-8.

Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11:3601, 2020. doi: 10.1038/s41467-020-17266-6.

Benjamin W. Walls and Suljo Linic. Use of large language models for extracting and analyzing data from heterogeneous catalysis literature. *ACS Catalysis*, 15(17):14751–14763, 2025. doi: 10.1021/acscatal.5c03844.

Xingyao Wang, Sha Li, and Heng Ji. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3640–3663. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.202. URL https://aclanthology.org/2023.acl-long.202/.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50208–50232. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24h.html. arXiv:2402.01030.

Yubin Wang, Xingjian Wei, Jiang Wu, others, and Conghui He. Dianshi-rxndb: A large-scale, fine-grained organic reaction data platform built via a fully automated pipeline for researchers and ai agents. *arXiv preprint arXiv:2609.06703*, 2026. Fully automated extraction and normalization of patent-scale reactions; ORD-schema based.

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.

Damian M. Wilary and Jacqueline M. Cole. ReactionDataExtractor: A tool for automated extraction of information from chemical reaction schemes. *Journal of Chemical Information and Modeling*, 61(10):4962–4974, 2021. doi: 10.1021/acs.jcim.1c01017.

Damian M. Wilary and Jacqueline M. Cole. ReactionDataExtractor 2.0: A deep learning approach for data extraction from chemical reaction schemes. *Journal of Chemical Information and Modeling*, 63(19):6053–6067, 2023. doi: 10.1021/acs.jcim.3c00422.

Friedrich Wöhler. Ueber künstliche bildung des harnstoffs. *Annalen der Physik und Chemie*, 88(2): 253–256, 1828. doi: 10.1002/andp.18280880206.

Yuanjie Xiang, Yanghong Luo, Renshuang Liu, Jiajun Tao, Wei Hu, Mei Ouyang, and Wei Liu. RGReco: a unified framework for automated R-group recognition in chemical publications. *Journal of Cheminformatics*, 18:45, 2026. doi: 10.1186/s13321-026-01179-5.

Haote Yang, Hui Wang, Chen Zhu, Jingchao Wang, Linye Li, Hongbin Lai, Huijie Ao, Yongxuan Lyu, Jiang Wu, Jiaxing Sun, Lua Chen, Yuanyuan Cao, Ruijie Zhang, Shengxin Lu, Lijun Wu, Bin Wang, and Conghui He. Molrecbench-wild: A real-world benchmark for optical chemical structure recognition. *arXiv preprint arXiv:2605.05832*, 2026a.

Haote Yang, Jiang Wu, Jingchao Wang, Xingjian Wei, Lixin Ma, et al. Mineru.chem: A high-precision system for optical chemical structure and reaction recognition. *arXiv preprint arXiv:2608.03525*, 2026b.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2405.15793.

Ke Yang, Yanqi Luo, Qingyue Hu, Mengjie Song, Junxiang Liu, Zhengyi Li, Bijin Li, and Xiaoqiang Sun. Selective C(sp3)–S bond cleavage of thioethers to build up unsymmetrical disulfides. *The Journal of Organic Chemistry*, 88:13699–13711, 2023. doi: 10.1021/acs.joc.3c01355.

Wei Yang, Zihao Liu, Tao Tan, Xiao Hu, Hong Xie, Lulu Li, Xin Li, Jianyu Han, Defu Lian, and Mao Ye. AgentCAT: An LLM agent for extracting and analyzing catalytic reaction data from chemical engineering literature, 2026c. URL https://arxiv.org/abs/2602.18479.

Yi Yang, Zhimeng Liu, Weize Wu, Yuyang Zhang, Hongbo Zhang, Jing Lin, Siyu Wu, Zihan Chen, Ye Sun, Ruiming Li, Nan Wang, Rui Wang, Zipeng Liu, Huobin Tan, Lei Shi, Yue Zhang, Hongyi Gao, and Ge Wang. MaterialBrain: High-performance material synthesis extraction via human–AI-curated few-shot large language models. *Journal of Chemical Information and Modeling*, 66 (1):228–245, 2026d. doi: 10.1021/acs.jcim.5c02299.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.03629.

Naruki Yoshikawa, Marta Skreta, Kourosh Darvish, Sebastian Arellano-Rubach, Zhi Ji, Lasse Bjørn Kristensen, Andrew Zou Li, Yuchi Zhao, Haoping Xu, Artur Kuramshin, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Large language models for chemistry robotics. *Autonomous Robots*, 47:1057–1086, 2023. doi: 10.1007/s10514-023-10136-2.

Yanqiu Zhang, Zhiliang Zhang, Yunxin Xia, Jiayi Wang, Yanqing Peng, and Gonghua Song. TfOH-catalyzed cascade reaction: Metal-free access to 3,3-disubstituted phthalides from o-alkynylbenzoic acids. *The Journal of Organic Chemistry*, 88:12924–12934, 2023. doi: 10.1021/acs.joc.3c00760.

Ming Zhong, Siru Ouyang, Minhao Jiang, Vivian Hu, Yizhu Jiao, Xuan Wang, and Jiawei Han. ReactIE: Enhancing chemical reaction extraction with weak supervision. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12120–12130. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.findings-acl.767. URL https://aclanthology.org/2023.findings-acl.767/.

Ming Zhong, Siru Ouyang, Yizhu Jiao, Priyanka Kargupta, Leo Luo, Yanzhen Shen, Bobby Zhou, Xianrui Zhong, Xuan Liu, Hongxiang Li, Jinfeng Xiao, Minhao Jiang, Vivian Hu, Xuan Wang, Heng Ji, Martin Burke, Huimin Zhao, and Jiawei Han. Reaction Miner: An integrated system for chemical reaction extraction from textual data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 389–402. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-demo.36. URL https://aclanthology.org/2023.emnlp-demo.36/.

Appendix guide. Appendix A gives executable representations and tool interfaces; Appendix B summarizes task prompts and execution settings; Appendix C defines datasets, reference construction, and metrics; Appendix D reports additional results and execution analyses; Appendix E describes molecular recognition; and Appendix F supplies enlarged source evidence for the case study.

A Representation and tool interfaces

Execution boundaries. Each record represents one executed experiment and its ordered operations (Figure 5b). Scope cells, optimization trials, failures, and scale-up repeats remain separate executions; repeated depictions of the same experiment are reconciled. An isolated intermediate marks a boundary between records.

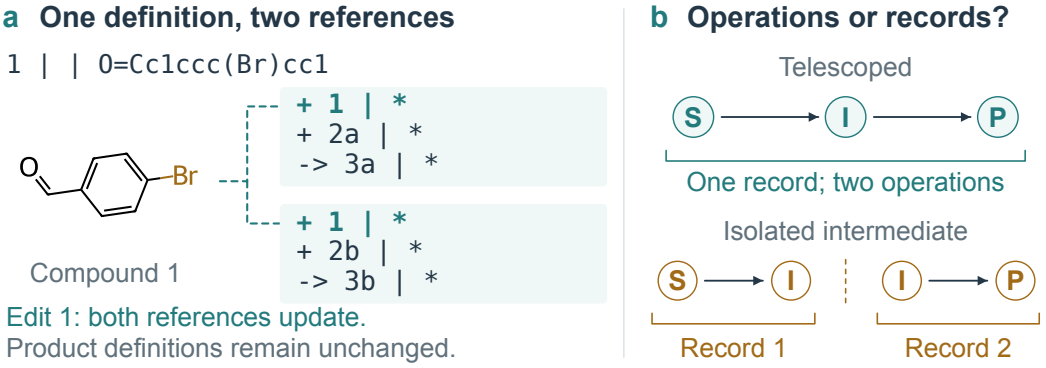


Figure 5: **Two kinds of sharing, two explicit boundaries.** (a) Dashed links bind row references to one compound definition. Editing that definition updates both references; product definitions remain separate. Snippets show label/structure fields. (b) Substrate/intermediate/product states distinguish telescoped operations from a separately isolated intermediate.

Executable rows and shared definitions. Figure 5a uses a five-entry ledger and two reaction blocks sharing aldehyde `1`. An `@` starts a record, whereas `--` starts another operation. Component cells specify label, structure, name, amount, concentration, and roles; product cells specify label, structure, yield, and stereochemical annotation. A `*` resolves a structure or name from the ledger; missing targets produce line-numbered errors. Condition values and yields remain strings, and components may carry several roles.

Paired construction and output views. `chem instantiate` substitutes one fragment map into both source-selected scaffolds. Each numbered scaffold dummy and fragment dummy must have one neighbor; the tool joins these neighbors using the scaffold attachment bond, removes the dummies, and sanitizes the graph. For example, substituting `[*]C` into `[*:1]N` and `[*:1]N=Cc1ccc(Br)cc1` gives `CN` and `CN=Cc1ccc(Br)cc1`. Irregular rows may be authored directly. The canonical view retains ordered operations and multi-role components. The flat compatibility view prioritizes reactant, catalyst/ligand, solvent, then other roles; it retains the first operation's temperature/time and summarizes later conditions. The evaluation adapter instead uses the first role and pools reactants across operations.

Instantiation edge cases. A dummy with zero or multiple neighbors fails compilation with a line-numbered error rather than guessing an attachment; after the join, valence violations reject the row while unspecified stereocenters remain unspecified—stereochemistry is carried by the printed evidence string, not inferred. For under-specified drawings, all feasible attachment sites are expanded only when the source marks a group as non-located; definite numeric repeats are expanded, indefinite (n/m) repeats kept once, Markush placeholders transcribed as printed.

A.1 COMPLEMENTARY FEEDBACK CHANNELS

The interfaces separate executable checks from evidence inspection. PubChem (Kim et al., 2025) and optional DECIMER (Rajan et al., 2023) provide suggestions, with the paper authoritative. Printed stereochemical evidence determines absolute configuration.

`compile` consumes rows and ledger definitions to produce canonical/flat records or line-numbered errors; `chem instantiate` consumes scaffolds and a fragment map to produce structures. On compiled records, `check` reports errors/warnings (errors return failure), `chem audit` reports conflicts/inspection items, and `chem renderback` produces labeled molecular sheets. `submit` exports flat records for a document; validation is performed through the preceding check commands. These are subcommands of `tools/rs.py`; image inspection compares their visual outputs with the source.

Render-back inspection. `chem renderback` renders every compiled structure with RDKit, grouped by source figure into labeled sheets (record identifier, label, structure, page pointer) that the agent inspects side by side with the original. Discrepancy identification is deliberately *not* automated: no detector scores the pair, so recognition uncertainty is not laundered into a confidence number; a visual mismatch maps directly to a workspace edit whose recompilation propagates to every record sharing the reference. Structures that fail to render or sanitize are flagged as warnings with figure and entry identifiers.

Count-before-extract reconciliation. The reader tallies expected experiments per source group (scheme, table, or narrative block) before writing and reconciles compiled records against this inventory group by group at the end; a shortfall triggers a targeted re-read and missing entries are fetched back. Omitting an experiment from *both* output and tally is the known blind spot of pure counting, so the tally is taken before reading (the source is the counting unit) and cross-checked against captions and running text during audit.

Audit rules. A fixed checklist over the compiled dataset; each rule emits a *conflict* (almost surely an error) or an *inspection item* (legal but worth re-checking): (i) the same catalyst/reagent name resolving to multiple structures (conflict); (ii) one label pointing to different structures across records (conflict); (iii) duplicate reactions with identical reactant and product multisets; (iv) per-figure minority stereo-specification; (v) records lacking yields where the paper likely prints one; (vi) heavy-atom deficits aggregated by signature (a shared missing co-reactant is often drawn once above a table); (vii) an ee/dr/er annotation whose structure lacks the corresponding stereocenters (conflict; 1:1 mixtures exempt).

Table 3: **Complementary checks on twelve synthetic inputs.** Check reports errors; Audit reports conflicts/informational items (C/I). Inventory compares output with a reader-maintained count. "–" denotes a downstream check skipped after compilation failed.

Synthetic condition	Compile	Check	Audit C/I	Inventory
Source-consistent control	Pass	0	0/1	Agree
Repeated-execution control	Pass	0	0/2	Agree
Unrecognized role	Reject	–	–	–
Unresolved ledger reference	Reject	–	–	–
Duplicate ledger label	Reject	–	–	–
Invalid SMILES	Pass	2	0/1	Agree
Conflicting inline structure	Pass	0	1/1	Agree
Consistent positional error	Pass	0	0/1	Agree
Output-only omission	Pass	0	0/0	Mismatch
Output and inventory omission	Pass	0	0/0	Agree
Incorrect page pointer	Pass	0	0/1	Agree
Incorrect yield string	Pass	0	0/1	Agree

Table 3 uses the implemented tools. Repeated executions receive an informational notice. A consistently encoded positional isomer, wrong yield, or wrong page pointer requires source inspection;

jointly omitting an execution from output and inventory also leaves counts equal. Together, these checks guide further source inspection and revision.

B Prompts and reproducible execution

B.1 Task prompts and controlled interventions

All document configurations read the main-text PDF and retain the same 19-paper denominator. Table 4 summarizes instruction changes and three runtime entrypoint interventions; other tool access remains available.

Table 4: **What each intervention changes.** Runtime arms block a named entrypoint; medium effort reuses the full prompt.

Configuration	Requested change
Full workflow	Persistent ledgers; compilation, count reconciliation, audit, and render-back.
Direct reading	Single-pass minimal JSON; no prescribed workspace or feedback loop.
Shared-JSON references	Nested JSON with a shared compound table and per-record references.
Direct JSON	Author nested JSON; add initial inventory and retain feedback requests.
Ledger-free	Remove both ledgers; author inline structures, retaining instantiation.
Count omitted	Remove per-figure counts and final reconciliation.
Render-back omitted	Remove render-back requests and its tool-list mention.
Audit unavailable	Full prompt unchanged; block the audit command entrypoint.
Render-back unavailable	Full prompt unchanged; block the render-back command entrypoint.
Check unavailable	Full prompt unchanged; block the check command entrypoint.
Medium effort	Full prompt; change only the reasoning setting.

The runtime labels denote entrypoint restrictions; Table 10 records their observed returns.

B.2 Execution settings and artifact map

Table 5: **Document-level execution settings.** Each selected run has a four-hour timeout and uses its harness-native reasoning controls.

Model	Harness	Requested reasoning	Outputs / 19
GPT-5.6-Luna	Codex	max	19
DeepSeek V4.1 Flash	Codex	max	18
DeepSeek V4.1 Flash	Claude Code	31,000 tokens	19
Qwen3.8-Flash	Claude Code	31,000 tokens	19
DeepSeek V4.1 Flash	pi	high	17
Qwen3.8-Flash	pi	high	19

The document manifests record Docker execution and Codex CLI 0.154.0. Claude Code's thinking budget is requested through the adapter; pi's manifests record a 272,000-token context limit and its launch scripts specify `high` reasoning. All prompt interventions use GPT-5.6-Luna/Codex, with only the effort intervention changing `max` to `medium`. Extraction inputs comprise the source papers and task instructions; reference annotations are isolated from the reader, and network controls are recorded per run. Appendix C defines retry selection; Appendices D.2 and E specify figure- and molecule-specific execution.

Model versions and access dates. Table 6 pairs the recorded API identifiers with the providers' public release dates (OpenAI, 2026; DeepSeek, 2026; Alibaba Cloud, 2026). The selected document-level runs started on September 17–19, 2026 for the deployment configurations and September 24–25, 2026 for the full-protocol family and its interventions, all on GPT-5.6-Luna except the DeepSeek V4.1 Flash and Qwen3.8-Flash deployments (September 18, 2026). Release

dates describe public availability; the recorded identifiers resolve to provider-managed API endpoints whose underlying checkpoints can change.

Table 6: **Model identifiers and public release dates.** For Qwen3.8-Flash, the date is the model-service launch recorded by the provider. Official records were checked on September 23, 2026.

Model	Recorded API identifier	Release date
GPT-5.6-Luna	`gpt-5.6-luna`	2026-07-09
DeepSeek V4.1 Flash	`deepseek-flash`	2026-09-10
Qwen3.8-Flash	`qwen3.8-flash`	2026-08-26

C Datasets, reference construction, and metrics

C.1 Dataset inventory and metric scope

Table 7 specifies the input units, targets, and matching criteria for each separately reported task.

Table 7: **Evaluation units and reference versions.** D2 and D3 have overlapping inputs; D4 evaluates molecular transcription on a development subset.

Set	Inputs	Targets	Final reference / criterion
D1	19 main-text PDFs	1,198	Source-grounded execution inventory; non-stereochemical reaction matching.
D2	324 figures	2,985	Research v3, OpenChemIE v4, review v2; soft and hard reaction matching.
D3	78 scope figures	1,007	Aligned v4; complete structures.
D4	500 molecule images	500	Fixed development subset; author-reviewed molecular transcription.

What a structural match includes. D1 and D2 soft matching compare complete reactant and product structures after the released scorer's normalization, without stereochemistry. D2 hard matching additionally requires reference-designated drawn condition molecules. These scores evaluate reaction structures; yields, operation order, and free-text conditions are retained as additional output fields. Text-based benchmarks such as ChEMU evaluate condition and yield entities and their event links (He et al., 2020); our whole-document task additionally requires binding information across schemes, tables, captions, and procedures. The benchmarks used here supply no joint score for this complete experiment record, so structural matching is complemented by source-linked inspection of the extracted fields in Section 4.3.

Micro-averaged reaction matching. For evaluation unit u, let P_u be its predicted reactions, G_u its references, and M_u a maximum-cardinality one-to-one matching under the dataset's rule. The corpus counts and micro scores are

$$T = \sum_u |M_u|, \qquad N_P = \sum_u |P_u|, \qquad N_G = \sum_u |G_u|, \tag{3}$$

$$\text{Precision} = T/N_P, \qquad \text{Recall} = T/N_G, \qquad F_1 = 2T/(N_P + N_G). \tag{4}$$

Each prediction and reference contributes to at most one match. Precision is set to zero when there are no predictions. Empty or missing outputs contribute no matches and retain their references. Matching is document-local for D1 and figure-local for D2/D3; corpus scores pool the resulting counts. The exact normalization and condition reconciliation follow the protocols below.

C.2 Source-grounded reference reconstruction

The 19-paper D1 benchmark originally supplies 2,149 Reaxys-aligned references. Our unit is a paper-supported execution. For example, an upstream precursor and a final product can be linked

even when the reported experiment starts from an isolated intermediate; counting both would reward an unsupported execution boundary. We reconstruct the inventory from source evidence, reconcile repeated depictions, and correct molecular identities and label assignments while retaining distinct optimization trials, failures, and scale-ups.

All 19 documents remain. The resulting 1,198 entries comprise 516 retained, 574 reconstructed or revised, and 108 explicitly added entries. These categories describe the entries in the final inventory. Each entry has a document-local source identifier and page-level evidence anchor; 20 unsuccessful experiments have an empty product side where no product structure is supported. The original references are retained separately, and every system uses the same corrected inventory and matcher. The score snapshot records full prediction/reference fingerprints.

Audit trail of the reference inventory. The inventory went through audited correction rounds—a structural pass (eleven fixes), a major-graph pass (three papers' backbone and chain-length errors), an as-charged pass (complete salts, counterions, recovered substrates), and a final adjudication of salt roles and unreacted substrates—each keeping the inventory fixed in size and order, retaining the original references separately, and re-scoring all frozen cohorts uniformly. Two aggregate checks characterize the reference revisions: the *original* supplied references scored as predictions against the corrected inventory give 37.94 F1 on D1 and 99.70 on D2. These measure agreement between reference versions under the matcher, while source-linked adjudications document the underlying corrections. Typical adjudications differing from the database include single-"main-reactant" entries for multi-component executions (59% of the original corpus), duplicate product accounting across precursor variants, and executed-boundary corrections. The final inventory was verified by an independent blind review: two non-author reviewers with chemistry training audited the entries against the source articles, and the inventory was released after their audit.

D1 matching. The released ChemEAGLE scorer (commit `995cb65`; RDKit 2025.09.5) removes stereochemistry, canonicalizes structures, normalizes eligible metal/charged structures through InChI, and finds a one-to-one reaction assignment. We retain its reactant–condition reconciliation; the adapter uses primary roles and pools reactants across operations. Complete-structure matching tests whether participants specify the transformation, while the full reference denominator tests coverage.

Output selection. Each configuration contributes its latest successful output per document by manifest start time. This selection includes operational retries; the reported scores characterize the selected outputs. Recorded contamination exclusions apply, with one scanner-source quotation retained after adjudication as a false positive. Missing outputs retain their references and yield zero matches.

External implementations and comparison scope. None of the three external systems ships predictions for our document-level inputs (released artifacts are benchmark materials only), so comparison requires execution; we rerun the released implementations unchanged in all task-specific components, with one disclosed substitution (vision-language calls routed to GPT-5.6-Luna, marked †). OpenChemIE runs locally at zero API cost; the ChemEAGLE/MERMaid reruns consume 7.14M input and 2.62M output tokens across 1,418 calls (API-equivalent $4.58), the same order as one UREA corpus pass. Every system receives the same references, matcher, denominators, and output-selection policy; failures and empty deliveries remain in the denominators. Its published Reaxys evaluation accepts participant subsets and equivalent tautomers, and its scope benchmark evaluates a separate input setting (Fan et al., 2024); our common criterion requires complete structural recovery and coverage of the execution inventory under automatic PDF processing.

ChemEAGLE retains its code and specialized checkpoints, with vision-language calls routed to GPT-5.6-Luna through an adapter. MERMaid also uses GPT-5.6-Luna; VisualHeist runs in the compatible ChemEAGLE environment and supplies its downstream images. MERMaid's published accuracy concerns a 100-article, three-domain evaluation (Leong et al., 2025). Here, SMILES attachment encounters dictionary-cleaning failures and missing optimization-key warnings; the 271 delivered entries contain no reaction structures and receive zero structural matches. Scoring all delivered outputs measures each pipeline's end-to-end conversion under the shared task.

Reconciling published and rerun scores. Table 8 places the available ChemEAGLE numbers side by side and distinguishes four aspects of their evaluation protocols.

(i) Reference version. The published D1 score of 62.39 uses the original database references; the other two columns use the corrected execution inventory. The 37.94 reference-to-reference F1 quantifies cross-version agreement under the structural matcher.

(ii) Backbone and harness. On D2 the reference versions nearly coincide (99.70 reference-to-reference F1), while the adapted rerun scores 21.44 against the published native-configuration score of 76.27. UREA scores 75.24 on the corrected references. The native and adapted results provide configuration-specific reference points, with the backbone, harness, and execution protocol specified for each. Published subset scores are 70.88/81.16/80.73 for research/OpenChemIE/review; UREA scores 74.69/82.56/60.07 on the corrected versions.

(iii) Output form. On D1, 32% of the rerun's predictions (246 of 771) are unsubstituted R-group templates, which cannot match concrete-structure references; this is the table-form output the released pipeline produces for whole-document inputs, not a scoring artifact.

(iv) Failure retention. Reruns keep failed and empty deliveries in the denominators: ChemEAGLE delivers outputs for 273 of 324 D2 figures (Table 13), and the references of undelivered figures count against it under our coverage policy.

Reporting these protocol dimensions preserves both the common-reference comparison and the published native-configuration context.

Table 8: **ChemEAGLE under three protocols.** Published = original references under ChemEAGLE's own backbone and harness (subset values in text); rerun = corrected references with vision-language calls routed to GPT-5.6-Luna; UREA = corrected references. The published column and the other two use different reference versions, disclosed rather than aligned.

	Published	Rerun†	UREA (corrected refs)
D2 overall (hard)	**76.27**	21.44	75.24
D1 document-level	62.39	28.34	89.73

Baseline coverage. We rerun three document-level systems end-to-end against a shared execution inventory and matcher, retaining published-configuration context in Table 8. RxnCaption-VL adds a complementary diagram-parsing comparison on D2/D3 (Tables 12, 13). Its adapter converts caption-format predictions to structural records using OCSR over the predicted bounding boxes (Appendix D.2); this configuration scores 0.12 hard F1 on D2 and 0.00 on D3 instances. Together, the comparisons cover whole-document reading, reaction-diagram parsing, and molecular transcription under explicit input and output contracts.

D ADDITIONAL RESULTS AND EXECUTION ANALYSIS

Interventions and deployments. Table 4 specifies the instruction changes; the three runtime arms keep the full prompt byte-identical and block a named tool entrypoint (audit, render-back, or check). Table 10 records the feedback actually returned under these interventions. Direct JSON replaces the row/ledger authoring workflow; ledger-free authoring removes both ledger files while retaining symbolic instantiation. Table 2 reports six deployments, with model identifiers in Table 6. Each result characterizes its model, endpoint, and harness under the native reasoning controls in Table 5.

D.1 PAPER-LEVEL ROBUSTNESS AND OBSERVED EXECUTION

Paired paper-level analysis. We recover per-paper matched, predicted, and reference counts from the selected outputs using the frozen D1 annotations and scorer; their sums reproduce the reported scores. Each of 10,000 bootstrap replicates (seed 20260921) resamples 19 whole papers with replacement, using identical indices for both configurations, then pools counts before computing the micro-F1 difference. Figure 6 shows pointwise 2.5th–97.5th percentile intervals with linear interpolation. The resampling measures variation across papers in the frozen evaluated outputs. The

Table 9: **Complete document-level comparisons.** All rows use 1,198 references across 19 papers. The lower block uses GPT-5.6-Luna/Codex at max effort, except the final medium-effort row. Docs counts selected usable outputs for that block; "—" is not reported on this basis. †GPT-5.6-Luna replaces the original vision-language backbone.

Configuration	Docs	Predictions	P	R	F1
UREA (full)	19	1,247	87.97	**91.57**	**89.73**
Direct reading	10	638	**89.97**	47.91	62.53
Count instructions omitted	19	1,234	81.44	83.89	82.65
OpenChemIE	—	897	21.18	15.86	18.14
ChemEAGLE†	—	771	36.19	23.29	28.34
MERMaid†	—	271	0.00	0.00	0.00
Count instructions omitted	19	1,234	81.44	83.89	82.65
Ledger-free authoring	19	1,310	76.18	83.31	79.59
Render-back instruction omitted	19	1,261	76.05	80.05	78.00
Shared-JSON references	18	1,175	76.17	74.71	75.43
Direct JSON authoring	18	1,177	72.05	70.78	71.41
Render-back unavailable	19	1,209	86.68	87.48	87.08
Audit unavailable	19	1,263	81.16	85.56	83.30
Check unavailable	19	1,185	78.90	78.05	78.47
Medium reasoning effort	19	1,029	18.08	15.53	16.70

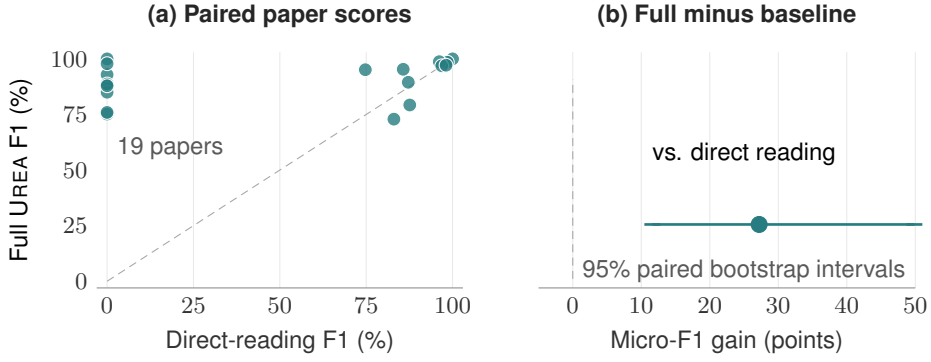


Figure 6: **The whole-document advantage extends across papers.** Left: each point pairs the selected outputs for one article; the diagonal marks equal F1. Right: gains pool reaction counts, with 95% percentile intervals from 10,000 paired whole-paper resamples.

full-versus-direct gain ranges from 23.61 to 32.03 points when any one paper is omitted; paper-level outcomes are 13 wins and six losses. The representation comparisons vary the authoring format together with its instructions.

Observed tool use. Table 10 pairs explicit tool requests with their returns across 76 selected runs, connecting each instruction or entrypoint intervention to its observed execution. All 19 full-protocol runs request both audit and render-back, and 18 receive render-back summaries. The counts describe returned artifacts; visual comparison uses image inspection.

Table 10: **Observed requests and returns under instruction and entrypoint interventions.** Entries count runs out of 19 with at least one explicit request or paired return containing the indicated summary. The final column counts runtime dependency errors in render-back returns, including errors preceding successful retries.

	Audit		Render-back		Dependency error
Configuration	Request	Summary	Request	Summary	in RB return
UREA (full)	19/19	19/19	19/19	18/19	16/19
Render-back instruction omitted	19/19	18/19	14/19	1/19	14/19
Render-back unavailable	19/19	19/19	16/19	0/19	1/19

Selected-run resource use. Table 11 sums selected successful runs, separating cached from uncached input; prose ratios use unrounded totals. The full protocol completes all 19 papers for $11.97 API-equivalent, with 87.6% of its input tokens served from cache.

Table 11: **Descriptive telemetry for selected successful outputs.** Token columns are in millions; Run-hours sums per-run wall time. Prose ratios use unrounded totals. Counts exclude discarded or failed attempts.

Configuration	Runs	Output	Uncached input	Cached input	Run-hours
UREA (full)	19	1.47	29.94	211.08	11.80
Direct reading	10	0.82	11.18	94.66	14.14
Count instructions omitted	19	1.48	26.97	224.80	16.23
Ledger-free authoring	19	1.41	25.77	201.51	17.49
Render-back instruction omitted	19	1.32	21.20	162.64	14.49
Shared-JSON references	18	1.25	22.42	198.19	34.10
Direct JSON authoring	18	1.37	28.64	213.17	22.59
Render-back unavailable	19	1.37	20.39	184.55	9.48
Audit unavailable	19	1.52	28.04	225.49	34.69
Check unavailable	19	1.49	28.91	231.49	11.86
Medium reasoning effort	19	0.21	4.92	12.05	2.55

Ledger sharing statistics. In the selected full-protocol runs, authoring goes entirely through the ledger: all 5,803 structure cells across the 1,247 records are $\star$ references (zero inline SMILES) resolving to 1,633 distinct definitions. Reuse is the dominant pattern: 488 definitions (30%) serve at least two records and receive 80.0% of the references; a shared definition serves 9.5 records on average (up to 144 for one solvent), and 148 definitions serve five or more. A single correction updates every citing record on recompilation, concentrating review on reusable definitions. Verification concentrates the same way: reviewing each of the 1,633 definitions once covers all 5,803 citing cells.

Cost, wall time, and failure accounting. One document costs $0.42–0.65 API-equivalent across the max-effort configurations (full protocol $11.97 over 19 papers, $0.63 per document; all-synthesis set $0.80, i.e., $0.030 per reaction; medium effort $0.08); 86.7–90.0% of input tokens are cache hits (87.6% for the full protocol, about 12.7M input tokens per document at $0.20/M uncached, $0.02/M cached, $1.20/M output). Wall time averages 37 minutes per document for the full protocol under max effort with a four-hour cap (30–114 minutes across configurations); medium effort finishes in minutes but collapses to 16.70 F1. Every input with no selected usable output stays in the reference

denominator; exhausted retries and native empty or filtered outputs receive zero matches. Cheaper public models reach 83.07 and 83.72, bounding dependence on the primary backbone.

Source-grounded verification. Compilation establishes record consistency; source inspection establishes the reading's fidelity. Execution boundaries, page anchors, and audit checks direct that inspection to the relevant evidence.

D.2 FIGURE-LEVEL PROTOCOLS AND COMPLETE RESULTS

Inputs and D2 matching. D2 has 139 research, 78 OpenChemIE, and 107 review figures (1,403, 1,085, and 497 references); versions are research v3, OpenChemIE v4, and review v2. D3 uses aligned v4 references (1,007 instances). D2 and D3 share some inputs; identical image files reuse third-party inference, and byte-distinct images are processed separately. Corrections address connectivity, labels, and evidence for stereochemistry; unavailable source articles/supporting information and unresolved semantic questions remain recorded for author review. D2 soft matching ignores stereo; hard matching additionally requires reference-designated drawn condition molecules, while allowing extra predicted conditions. Both retain the released ChemEAGLE role-reconciliation behavior (Table 12).

Table 12: **D2 subset results and condition sensitivity.** Hard matching includes reference-designated drawn conditions; soft matching compares reaction structures. Overall scores pool matched, predicted, and reference counts.

	D2 subset hard F1			D2 overall F1	
System	Research	OpenChemIE	Review	Hard	Soft
OpenChemIE	7.13	10.03	1.59	7.13	11.57
ChemEAGLE	15.96	38.79	3.41	21.44	23.52
MERMaid	8.44	0.00	0.00	4.61	9.73
RxnCaption-VL	0.26	0.00	0.00	0.12	0.12
UREA	**74.69**	**82.56**	**60.07**	**75.24**	**78.47**

Execution and adapters. UREA uses GPT-5.6-Luna/Codex at max effort with a one-hour figure timeout and its ledgers, instantiation, compilation, and checks. External pipelines allow 30 minutes per attempt and two attempts, selecting the first format-valid completion; native empty/filtered outputs count as complete. UREA follows the document output-selection policy above. External runs allow only model-inference network traffic; UREA's controls are recorded per run. ChemEAGLE retains specialized checkpoints with GPT-5.6-Luna vision-language calls. OpenChemIE uses native image recognition, coreference, and R-group substitution without PDF text/tables. MERMaid retains filtering and SMILES postprocessing; external PubChem is disabled, with local dictionary/name fallbacks retained. RxnCaption-VL runs its released detection and captioning weights unchanged; its adapter crops the source image at each predicted bounding box and reads the structure with the same OCSR used above. Adapters preserve the delivered structures, including invalid strings and placeholders.

Coverage and reaction output. Table 13 separates delivery from matching; missing figures remain in all denominators. Three UREA D2 outputs are empty. MERMaid returns 268 empty D2 outputs and filters out all D3 outputs; 98 of OpenChemIE's 101 D3 candidates retain placeholders.

Error decomposition. Condition recovery dominates D2's hard-matching losses: dropping the drawn-condition requirement raises overall F1 from 75.24 to 78.47 (Table 12); 93 soft matches fail hard matching on drawn conditions (49 in the research subset). On D3, stereochemistry is a small residual: of 784 connectivity-level matches, 614 agree exactly, 135 add stereochemistry where the reference is silent, and 35 fail (12 omitting reference-specified centers, 23 conflicting); 126 of the 170 stereo-differing pairs differ on double-bond geometry alone.

Native OpenChemIE benchmark. We also attempted UREA on OpenChemIE's native R-group benchmark as released: 78 single figures under figure-only prompts, scored by the paper's exact-

match metric, reimplemented and validated by regenerating the released pipeline (precision reproduces within 1.4 points). We do not report scores from this benchmark. The released references specify 385 double-bond configurations, of which our conservative text-evidence audit of the same figures retains 38 (347 lack verifiable E/Z support), and the reported 69.5 F1 is not reproducible with the currently released pipeline, which returns no reactions for 17 of 78 figures (62.6 on regeneration): absolute scores there mix reference errors with reading errors and cannot be read as task performance. All system comparisons therefore use the corrected references (Table 13).

Table 13: **Complete figure-level counts and final scores.** Outputs counts delivered figure outputs, including empty outputs. Scores measure reaction extraction on D2 and instantiated reactions on D3. All references remain in the denominator.

System	Outputs	Predictions	Matched	P	R	F1
D2: hard matching (324 figures; 2,985 references)						
OpenChemIE	324	438	122	27.85	4.09	7.13
ChemEAGLE	273	2,024	537	26.53	17.99	21.44
MERMaid	324	530	81	15.28	2.71	4.61
RxnCaption-VL	323	367	2	0.54	0.07	0.12
UREA	324	2,778	2,168	**78.04**	**72.63**	**75.24**
D3: instances (78 figures; 1,007 references)						
OpenChemIE	78	101	0	0.00	0.00	0.00
ChemEAGLE	57	593	266	44.86	26.42	33.25
MERMaid	78	0	0	0.00	0.00	0.00
RxnCaption-VL	78	78	0	0.00	0.00	0.00
UREA	78	1,018	749	**73.58**	**74.38**	**73.98**

E MOLECULAR-RECOGNITION DEVELOPMENT STUDY

Section 4.6 and Figure 4b compare graph-output and direct-SMILES readers with MinerU.Chem (Yang et al., 2026b) on 500 MolRecBench-Wild development images (Yang et al., 2026a). The variants differ in both output representation and task-specific prompts.

Table 14: **Molecular-recognition counts underlying Figure 4b.** The common denominator is 500 development images under author review; †missing/unusable outputs count as errors.

System	Correct / 500	Missing†	Accuracy (%)
MinerU.Chem	378	0	75.60
UREA (graph variant)	377	7	75.40
UREA (SMILES variant)	445	8	**89.00**

Sampling from the 5,024-image local snapshot is stratified by benchmark subset and MinerU.Chem's released graph/SMILES outcome categories; seed selection targets those full-snapshot proportions. The final SMILES prompt requests one image-grounded transcription, permits at most one zoom or DECIMER check, and writes one JSON record directly. Generic labels and indefinite repeats use the task's normalization rules. The adaptation transfers image-grounded reading to single-molecule transcription.

Authors inspect source images with predictions or visualizations and record one current judgment per item. Table 14 retains all 500 images, including seven graph and eight SMILES failures. Prompt development used some of these images; reported accuracy follows the authors' image-based judgments.

F CASE-STUDY EVIDENCE

The following unmodified excerpts support Figure 2. We verified external systems' raw reactant/product lists against their adapted D1 predictions.

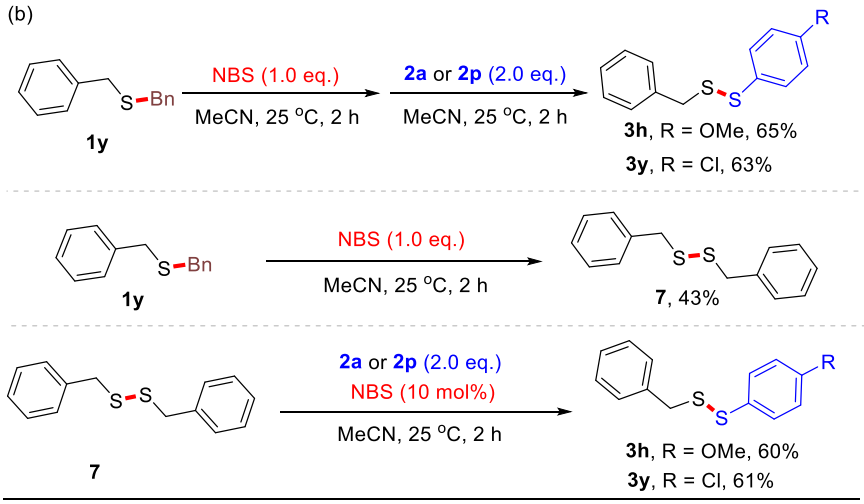
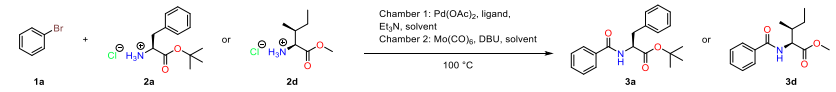


Figure 7: **Alternative partners and sequential operations.** Scheme 2(b) of Yang et al. (2023) (PDF p. 7): three rows encode five executions; UREA records 94–98 retain the branches and isolated intermediate **7** (Figure 2a). ChemEAGLE and OpenChemIE each export three records; both retain the separate preparation of **7**.

Table 1. Optimization of Reaction Conditions for the Palladium(0)-Catalyzed Aminocarbonylation Reaction[a]



Entry	Time (h)	Solvent	Pd mol%	Ligand	Amino acid	AA equiv	Yield (%)
1	2	DMF	5	XantPhos	Phe-OtBu	2	75
2	2	DMF	5	dppf	Phe-OtBu	2	NI
3	2	DMF	5	PPh_3	Phe-OtBu	2	26
4	2	DMF	5	dpePhos	Phe-OtBu	2	45
5	2	DMF	5	dppe	Phe-OtBu	2	NI
6	2	DMF	5	XantPhos	Phe-OtBu	3	70
7	2	DMF	5	XantPhos	Phe-OtBu	1.2	78
8	2	DMF	1	XantPhos	Phe-OtBu	1.2	82
9	2	DMF	10	XantPhos	Phe-OtBu	1.2	73
10	2	DMF	1	XantPhos	Ile-OMe	1.2	73
11	2	DMF	0.1	XantPhos	Ile-OMe	1.2	11
12	0.5	DMF	1	XantPhos	Ile-OMe	1.2	33
13	4	DMF	1	XantPhos	Ile-OMe	1.2	88
14	4	2-MeTHF	1	XantPhos	Ile-OMe	1.2	80[b]
15	4	MeCN	1	XantPhos	Ile-OMe	1.2	39
16	4	1,4-dioxane	1	XantPhos	Ile-OMe	1.2	NI[b, c]
17[d]	4	DMF	1	XantPhos	Phe-O*t*Bu	1.2	NI[c]

Figure 8: **Row-specific alternatives.** Table 1 of Lindman et al. (2023) (PDF p. 2). Rows 1–9 and 17 select **2a/3a**; rows 10–16 select **2d/3d**. Rows 9/10 share 73% yield but differ in partner and Pd loading (Figure 2c).

F.1 EXECUTED CONTROLS VERSUS PROPOSED TRANSFORMATIONS

In Zhang et al. (2023), PDF p. 4 reports isolation and characterization of **C**; Scheme 6 (p. 6) shows its formation (46%, with trace **4a**) and conversion with **2a** to **3a** (90%). Scheme 7 below instead proposes paths through **A**–**G**. UREA retains the two controls (records 65–66) and excludes the proposed transformations in Scheme 7. ChemEAGLE retains both controls plus ten mechanism records; OpenChemIE exports 19 mechanism records from overlapping detected regions. The experimental context establishes the role of each transformation.

Scheme 7. Proposed Mechanism

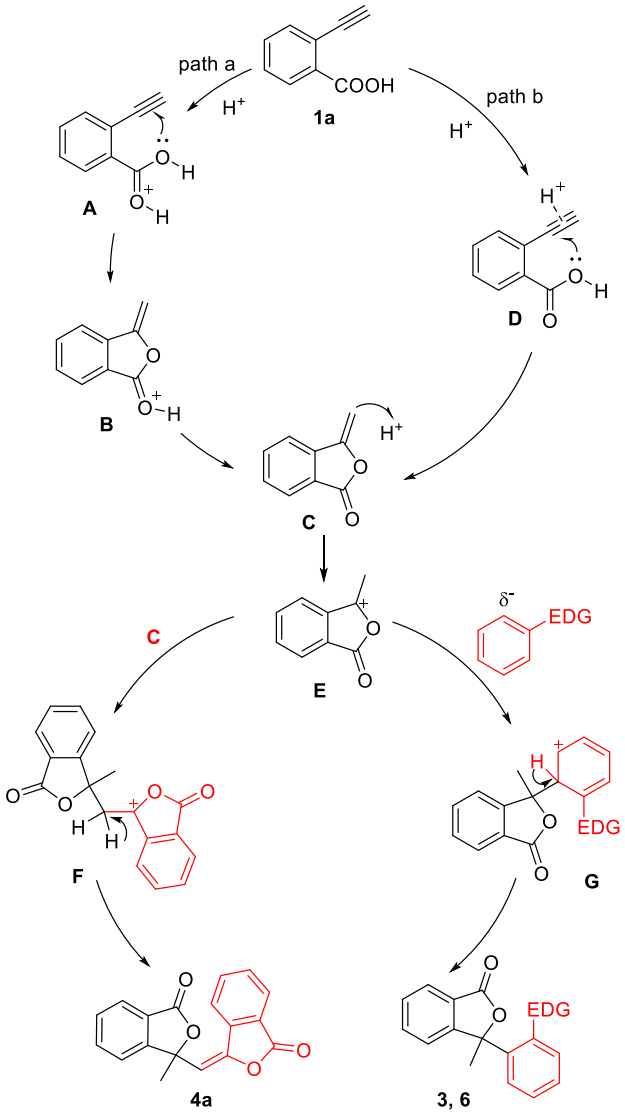


Figure 9: **Experimental context distinguishes controls from mechanistic proposals.** Unmodified Scheme 7 of Zhang et al. (2023) (PDF p. 7). Executed controls appear separately in Scheme 6 and Figure 2b.