

ChemReason-Bench: Benchmarking Large Language Models for Procedural Reasoning in Experimental Chemistry

Jinwei Zhang¹, Xucheng Liang², Yu Zhang¹,
Ruijie Yu¹, Xiaokang Yang¹, Yaohui Jin¹, Yanyan Xu^{1*}

¹MoE Key Lab of Artificial Intelligence, AI Institute,
School of Computer Science, Shanghai Jiao Tong University, China

²University of Southern California, USA

{jinwei.zhang, cynthiazhang, el_iauk, xkyang, jinyh, yanyanxu}@sjtu.edu.cn
rickylia@usc.edu

Abstract

Experimental protocols in organic synthesis specify not only the intended transformation but also an executable sequence of operations and conditions. While recent language models show strong chemistry knowledge, widely used evaluations remain less diagnostic of procedure-level decision making. In this setting, correctness requires consistent step ordering, feasibility under stated conditions, faithful entity–role grounding, and schema-parseable outputs that can be automatically validated against operational constraints. We present CHEMREASON-BENCH, a human-validated benchmark for verifiable experimental procedure reasoning built on a structured representation with explicit placeholders and a unified schema, enabling automatic checks of many operational constraints. From 500 reactions, we instantiate 7306 benchmark tasks across six complementary formats: ordering, step validation, condition validation, schema-constrained completion, contrastive choice, and evidence-grounded rationalization. We further release a large-scale instantiation of the same templates for downstream adaptation studies, kept disjoint from the evaluation set. Using a unified evaluation protocol, we benchmark diverse open-source, proprietary, and domain-specific models and observe clear variation across the capability surface. We also report controlled adaptation experiments in the appendix, where supervised fine-tuning improves small models, preference optimization adds limited gains in our setting, and a gap remains to the strongest evaluated systems.

1 Introduction

Experimental procedures form the operational backbone of chemical synthesis (Mehr et al., 2020). They describe not only the intended transformation, but also the ordered operations and conditions (e.g., reagent addition, temperature control,

quenching, extraction, drying, purification) that determine feasibility, safety, and yield (Vaucher et al., 2020; Zhang et al., 2025a). Procedural reasoning remains difficult for language models because correct outputs must respect dependencies that span an entire experiment, follow common laboratory conventions, and satisfy feasibility constraints imposed by the stated conditions (Zeng et al., 2023). Decisions made early in a protocol can limit which work-up steps are appropriate later, and a chemically plausible suggestion can still conflict with the procedure’s operational constraints. As synthesis automation and agentic lab assistants increasingly adopt language interfaces, these requirements become central: generations must remain procedurally consistent and constraint-satisfying, not merely chemically plausible (Burger et al., 2020; Boiko et al., 2023).

Procedure-centric representations have been studied along two closely related lines: extraction (Vaucher et al., 2020) and transcription (Zeng et al., 2023). Extraction approaches convert experimental prose into structured action sequences, enabling machine-executable procedures from patents and free-form laboratory narratives (Zhong et al., 2024). Recent work improves action and event extraction through stronger supervision, including LLM-assisted data generation and refined extraction models (Zhang et al., 2025b). In parallel, transcription-style formulations translate between human-readable descriptions and structured, machine-executable instructions, supporting normalization across heterogeneous sources. At scale, curated collections aggregate procedures from public upstream corpora such as patents and open repositories, providing a foundation for modeling procedure understanding (Liu et al., 2024).

Despite this progress, comparison in Table 3 indicates that many evaluations for chemistry-capable LLMs still emphasize chemistry knowledge and general reasoning, often via QA/MCQ-style bench-

*Corresponding author.

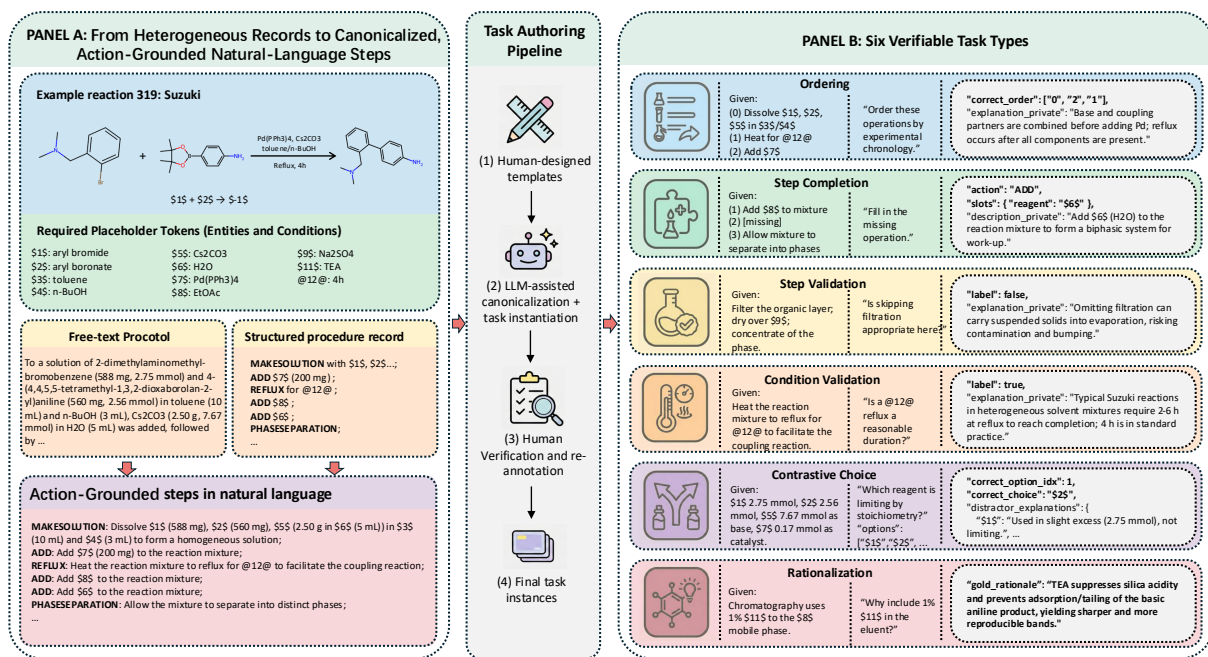


Figure 1: From heterogeneous procedure records to six verifiable procedure reasoning task types.

marks (Mirza et al., 2025; Chen et al., 2025; Li et al., 2026; Xie et al., 2025; Runcie et al., 2025) or broad task suites (Guo et al., 2023; Rein et al., 2023), offering limited diagnostic value for procedure-level decision-making. Consequently, it remains difficult to localize failures in operational reasoning: a model may select the correct option in a multiple-choice setting yet fail when required to enforce step-order constraints, maintain consistent entity–role bindings, or produce a machine-checkable structured decision. A single aggregate score can therefore hide execution-critical failure modes.

We introduce CHEMREASON-BENCH, a human-validated benchmark for verifiable experimental procedure reasoning. It adopts a structured procedure representation with explicit placeholders and a unified schema, enabling automatic checks of many operational constraints and supporting complementary task formats that probe distinct procedural competencies. CHEMREASON-BENCH instantiates six task types covering ordering, step- and condition-level validation, schema-constrained completion, option selection, and evidence-grounded explanation. We also release a large-scale instantiation CHEMREASON-TUNE of the same templates for downstream adaptation studies, with evaluation kept separate from training. Dataset overview is shown in Table 1.

Our main contributions are as follows:

- We introduce CHEMREASON-BENCH, a human-validated benchmark for verifiable experimental procedure reasoning under a unified schema.
- We define six complementary task types and a unified evaluation protocol (task-specific primary/auxiliary metrics and dual gen/lm scoring for discriminative tasks), enabling capability-level diagnosis beyond a single aggregate score.
- We release a large-scale template instantiation CHEMREASON-TUNE for downstream adaptation studies and report systematic results across diverse model families, including controlled adaptation results comparing SFT and GRPO (Appendix).

2 Related Work

Procedure extraction into action sequences. Prior work studies converting experimental procedures into structured action sequences, enabling machine-interpretable and executable procedure representations. Vaucher et al. (2020) introduce large-scale action extraction for chemical synthesis procedures. Recent efforts further improve extraction quality and supervision, including leveraging LLM-generated data for training and robustness (Zhang et al., 2025b). Related work also explores

Item	CR-Bench	CR-Tune
Overall		
Reactions	500	>20000
Task types	6	6
Total instances	7306	123903
Per-task instances		
Ordering	1266	20657
Contrastive Choice	1077	20650
Step Validation	1148	20646
Condition Validation	1117	20655
Step Completion	1483	20657
Rationalization	1215	20638
Fine-tuning splits		
SFT train	-	89210
SFT val	-	9912
SFT total	-	99122
GRPO train	-	19825
GRPO val	-	4956
GRPO total	-	24781

Table 1: Dataset overview of CHEMREASON-BENCH (CR-Bench) and CHEMREASON-TUNE (CR-Tune).

extracting structured information from organic synthesis procedures with fine-tuned language models, highlighting the practicality of schema-based outputs in chemistry text processing (Ai et al., 2024). More broadly, scientific action extraction has been studied in NLP with executable targets, e.g., mapping natural-language actions into programmatic representations (Zhong et al., 2024).

Transcription and procedure prediction. Complementary to extraction, transcription-style formulations translate between human-readable descriptions and machine-executable instructions under explicit operation schemas, supporting normalization across heterogeneous sources (Zeng et al., 2023). A related direction predicts experimental procedures from compact reaction representations, bridging reaction-level descriptions and operational protocols (Vaucher et al., 2021). Reaction-text pretraining and reaction-contextualized modeling further strengthen procedure understanding and generation settings, providing curated resources and modeling approaches that connect reactions to procedural text (Liu et al., 2024).

Automation and tool-augmented chemical assistants. Procedure-level consistency is increasingly important in automation settings where generated outputs may be executed or used for verification. Study on digitizing and automatically executing synthesis literature demonstrates end-to-end pipelines from text to execution (Mehr et al., 2020).

Robotic platforms and autonomous chemistry systems further motivate reliable procedure representations and constraint-aware decision making (Burger et al., 2020; Boiko et al., 2023). Tool-augmented LMs in chemistry emphasize the role of structured interfaces and external tools for grounded, actionable outputs (Bran et al., 2024).

Evaluating chemistry-capable language models.

Recent benchmarks for chemistry-capable LMs often emphasize chemistry knowledge and general scientific reasoning via QA/MCQ-style tests or broad task suites (Mirza et al., 2025; Guo et al., 2023; Chen et al., 2025; Li et al., 2026; Xie et al., 2025; Runcie et al., 2025). While valuable for assessing factual and conceptual competence, such evaluations can be less diagnostic of procedure-level decision making that requires enforcing step-order constraints, maintaining consistent entity-role bindings, and producing schema-valid structured decisions.

Positioning. ChemReason-Bench complements prior extraction and transcription work by evaluating verifiable procedure reasoning under an explicit schema with placeholders, and by probing a capability surface via multiple task formats rather than collapsing performance into a single score.

3 Task Formulation

CHEMREASON-BENCH evaluates whether language models can reliably reason over organic synthesis procedures beyond factual chemistry recall. Experimental protocols combine heterogeneous evidence and require procedural consistency, chemical plausibility, and faithful entity-role grounding.

A single aggregate score cannot characterize this capability surface: models may succeed at option selection yet fail to enforce step order, maintain placeholder consistency, or produce schema-valid structured decisions, which matters in automation settings where outputs may be executed or verified.

To make these failure modes measurable, we formulate six verifiable task types that cover complementary axes of procedural reasoning (Fig. 1). We annotate primary and secondary capability alignments in Fig. 2 to highlight the complementary roles of the six tasks and to distinguish task-specific competencies from cross-cutting ones.

3.1 Expert-Designed Task Space

Our task space is expert-defined: we specify procedural abilities relevant to executing and verifying

Task	Input / Output	Metric	Procedural Order	Action Validity	Condition Reasoning	Semantic Grounding	Reaction Rationale	Constrained Output	Work-up & Purif.	Format Faith.
Ordering	In: Text + Candidates + Legend Out: Order List	pairwise_accuracy exact_match kendall_tau_norm	P			S		S	P	S
Step Completion	In: Text + Schema (+Legend) Out: Missing Step (+Slots)	step_completion_score action_em_slot_f1 format_error_rate	P			S		P		S
Step Validation	In: Text (+Legend) Out: Binary	f1_positive accuracy, brier, ece auroc, auprc		P			P		S	S
Cond. Validation	In: Text (+Legend) Out: Binary	f1_positive accuracy, brier, ece auroc, auprc		S	P		P			S
Contrastive Choice	In: Text + Options (+Legend) Out: MCQ Index	top1_accuracy log_loss, ece, mrr				P	P	S	S	S
Rationalization	In: Text (+Legend) Out: Explanation	coverage_f1 rougeL_f1, bleu bert_score_f1		P	S		P			S

■ Primary Capability ■ Secondary Capability

Figure 2: Task–capability coverage matrix. Rows are the six task types and columns are the targeted reasoning capabilities. Dark blue cells indicate *primary* capability coverage and light blue cells indicate *secondary* coverage.

synthesis protocols, then define task forms with verifiable outputs. Each task type is defined by a template that specifies the evidence (context and candidates/options with schema/legend constraints), the required output format, and the evaluation protocol. Representative instances are shown in Fig. 1; complete task instances and format specifications are provided in Appendix D.

We instantiate templates at scale via a controlled authoring pipeline with LLM assistance and human verification (Fig. 3); full details are in Section 4.

3.2 Why Six Tasks

The six tasks jointly define a capability surface rather than a single axis. ORDERING and STEP COMPLETION emphasize procedural structure and constrained generation; the two validation tasks isolate decisions about actions versus conditions; CONTRASTIVE CHOICE stresses grounded selection among distractors; RATIONALIZATION probes causal explanation. Format faithfulness (placeholder consistency and schema compliance) is evaluated across tasks. This multi-task formulation supports diagnostic analysis of where models fail, motivating task-specific reporting rather than collapsing performance into a single scalar.

4 Dataset Construction

Our direct inputs are two curated procedure collections, OPENEXP (Liu et al., 2024) and CHEMTRANS (Zeng et al., 2023), which aggregate experimental protocols from multiple upstream public sources (e.g., patents and open repositories),

with representative upstream sources (e.g., USPTO (Lowe, 2017), ORD (Kearnes et al., 2021), OrgSyn (Organic Syntheses, Inc., 2025)) listed in Fig. 3 as provenance metadata for auditable data lineage.

We canonicalize procedure records into action sequences and render them as action-grounded natural-language steps. Each step is paired with an explicit action label from a fixed inventory of 27 integrated action types (Appendix E.5) and uses explicit placeholders under a unified schema, enabling strict parsing and automatic checking of many operational constraints.

4.1 Benchmark and Training Pools

We construct two reaction pools: a 500-reaction benchmark pool for controlled evaluation and a larger training pool of > 20k reactions for downstream adaptation. From the benchmark pool, we instantiate CHEMREASON-BENCH with 7306 human-validated task instances across six task types. Using the same templates, we additionally instantiate CHEMREASON-TUNE from the training pool to obtain > 120k verifiable training instances; evaluation remains disjoint from training.

4.2 Authoring and Quality Control

Instances are generated by deterministic processing with limited, schema-constrained LLM assistance (e.g., surface realization and candidate generation), followed by automatic validators and targeted human verification. All instances first undergo script-based validation for structural and content consistency, including schema conformity, required-field completeness, abnormal text length, consis-

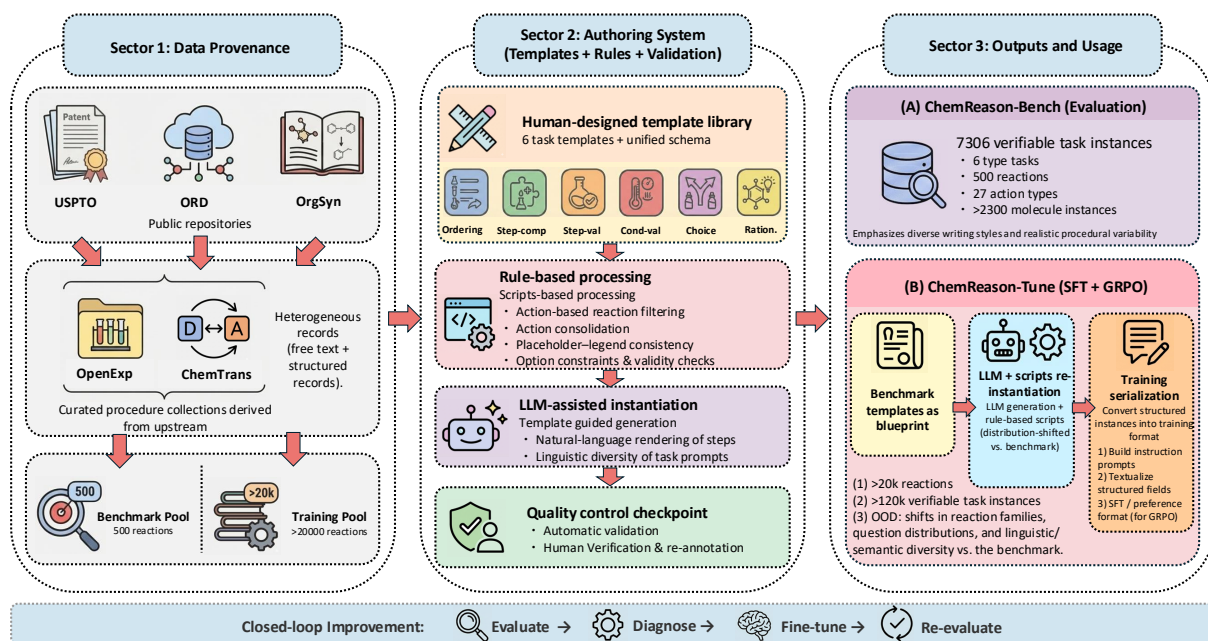


Figure 3: Overview of the ChemReason-Bench and ChemReason-Tune data authoring pipeline.

tency with the underlying experimental record, and duplicate/missing-instance checks; any violation triggers re-annotation. We further apply task-specific quality control with manual review of 50 sampled instances per task type from distinct experiments. For ORDERING and STEP COMPLETION, automatic checks verify consistency with the corresponding experimental record, and sampled instances are required to satisfy 100% correctness. For STEP VALIDATION, CONDITION VALIDATION, CONTRASTIVE CHOICE, and RATIONALIZATION, sampled instances are reviewed for record grounding, question validity, answer correctness, and placeholder consistency, with a 90% acceptance threshold; failures trigger re-annotation. Overall, the revision rate is approximately 9%, and all revised instances are double-checked by a second annotator. On independently double-annotated subsets ($N=50$ each), we obtain Cohen’s $\kappa = 0.72$ for STEP VALIDATION and $\kappa = 0.80$ for CONDITION VALIDATION, indicating substantial agreement. Full authoring details, task-specific checks, and per-task revision statistics are provided in Fig. 3, Appendix E, and Appendix Table 5.

5 Evaluation Protocol

5.1 Metrics

We evaluate models on six task types that capture complementary aspects of experimental procedure reasoning. For each task, we designate a

primary metric for headline comparison and additionally report *auxiliary metrics* that provide a more complete view of model behavior: strict correctness (e.g., exact match), ranking quality for ordering (e.g., Kendall’s τ), probability-sensitive and threshold-free measures for decision tasks (e.g., Brier/ECE, AUROC/AUPRC), and lexical/semantic quality metrics for rationales. The metrics for each task can be summarized in Appendix F.

Two complementary evaluation protocols (gen vs. lm). For the three discriminative tasks (Step Validation, Condition Validation, and Contrastive Choice), we evaluate each model under two complementary protocols. In gen, the model produces a full response that we parse into a discrete decision (YES/NO or an option index), reflecting end-to-end instruction following and generation behavior. In lm, we probe next-token probabilities under a constrained prompt (e.g., Respond with ONLY YES or NO or Respond with ONLY an integer index), yielding calibrated probability estimates over the decision space. We report both gen and lm scores in tables (“gen/lm”), and compute a single task score by averaging the two views. Full protocol details are provided in Appendix F.3.

The overall score macro-averages primary metrics across the six tasks with equal weights. The detailed definitions, computation logic, and formulas for all task-specific metrics are provided in Appendix F.2, F.3 and Table 7.

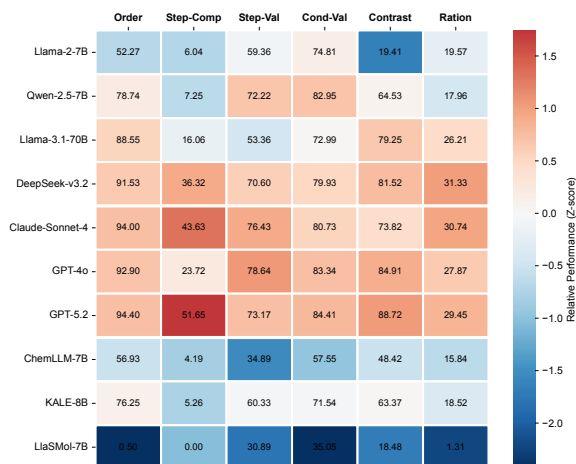


Figure 4: Per-task performance profile (selected models). Heatmap of task-level scores for a representative subset of models spanning open-source, proprietary, and domain-specific systems. Cells show the task scores in numbers, while the color encodes relative performance after per-task standardization (Z-score) to highlight strengths and weaknesses across tasks.

5.2 Overall Reporting

We report Primary-Overall as a macro-average over the six task families, assigning equal weight to each task type regardless of its instance count. Concretely, for the three discriminative tasks (STEP VALIDATION, CONDITION VALIDATION, and CONTRASTIVE CHOICE), we first compute a single task-level *primary metric* by averaging the gen and lm primary metrics (reported as “gen/lm” in the main tables). For the remaining tasks (ORDERING, STEP COMPLETION, and RATIONALIZATION), the primary metric is unaffected by the gen/lm protocol, so using either view (or their average) yields the same value. The final Primary-Overall is then obtained by averaging the six task-level primary metrics; the computation procedure is provided in Appendix F.3.

Appendix G and Appendix Tables 8–13 provides six per-task tables with all reported metrics. In each table, the Primary metric column corresponds to the task’s primary metric for that model: it is either the direct primary value (for tasks not affected by gen/lm) or the mean of gen and lm (for the three discriminative tasks). Averaging these six appendix Primary metric values per model exactly recovers the Primary-Overall reported in the main table.

6 Evaluation Results and Analysis

6.1 Experiment Introduction

We evaluate a diverse set of models spanning open-source general LLMs (Llama 2 (Touvron et al., 2023) /3.1 (Grattafiori et al., 2024), Mixtral (Jiang et al., 2024), Phi-3 (Abdin et al., 2024), Gemma 2 (Team et al., 2024), Qwen 2.5 (Qwen Team et al., 2025), and DeepSeek-V3.2 (DeepSeek-AI et al., 2025)), proprietary systems (Claude Sonnet 4 (Anthropic, 2025), Gemini 2.5 Flash (Comanici et al., 2025), GPT-4o (Hurst et al., 2024), GPT-5.2 (OpenAI, 2025), Grok-4 (xAI, 2025) and Qwen-3-max (Team et al., 2025)), and chemistry-focused models (ChemLLM (Zhang et al., 2024), LiaSMol (Yu et al., 2024), KALE (Dai et al., 2026), and ChemDFM (Zhao et al., 2025b)).

Across models, we use the same prompts and deterministic decoding (temperature = 0) with a fixed maximum output budget per task, ensuring comparable generation constraints across providers.

We report main results on CHEMREASON-BENCH in Table 2, with a compact Primary-Overall summary (macro-average over the six task families). Figure 5 visualizes the same ranking, and Figure 4 provides a task-wise view for a representative subset of models.

6.2 Primary-Overall leaderboard

Across model categories, proprietary models achieve the highest average Primary-Overall (66.8), followed by open-source models (52.1), while domain-specific chemistry models lag behind on average (30.5). The best proprietary model in Table 2 is GPT-5.2 (70.30), and the strongest open-source model is DeepSeek-v3.2 (65.21), leaving a gap of ≈ 5.1 points in Primary-Overall. Within open-source systems, scaling and training recipe both matter: Qwen-2.5-72B (64.65) approaches the best open-source result, while smaller open models are substantially lower, indicating that the benchmark remains challenging under realistic deployment budgets.

6.3 Per-task strengths and bottlenecks

Task-level metrics reveal uneven progress across procedural skills. ORDERING appears comparatively mature for strong general LLMs (e.g., PairAcc in the low-to-mid 90s for top proprietary models), suggesting that many models can recover coarse procedural dependencies when candidate steps are provided. In contrast, STEP COMPLE-

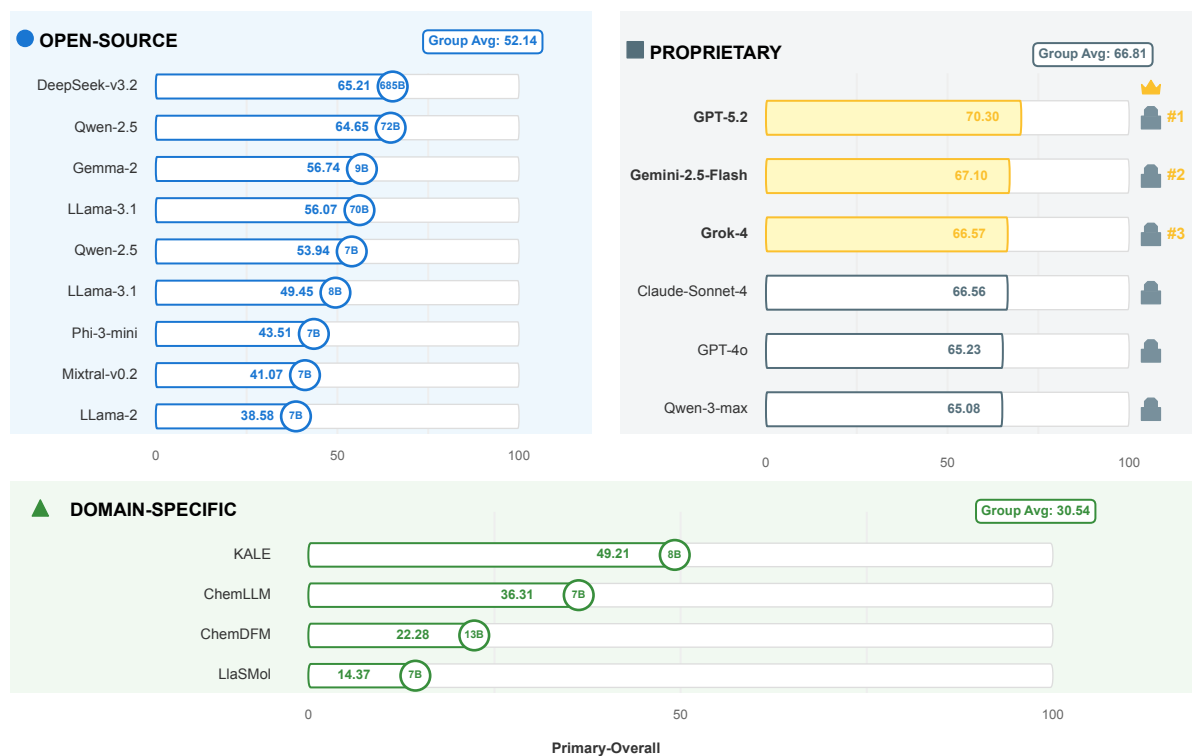


Figure 5: CHEMREASON-BENCH leaderboard. Primary-Overall comparison across three model groups: open-source general-purpose LLMs, proprietary LLMs, and domain-specific chemistry models. Bars report each model’s Primary-Overall score (macro-averaged over the six task families), with group-average scores shown for reference.

TION is a major bottleneck: the best proprietary result (GPT-5.2, SC-Score 51.65) is substantially higher than the best open-source result (DeepSeek-v3.2, 36.32), reflecting the difficulty of reconstructing structured missing actions and slots under strict constraints. RATIONALIZATION scores remain modest across the board (CovF1 roughly in the 20–31 range for the strongest models), indicating that producing explanations that reliably cover required evidence remains challenging even when task answers are correct.

6.4 Two-protocol reporting for discriminative tasks (gen vs. lm)

For the three discriminative tasks (STEP VALIDATION, CONDITION VALIDATION, CONTRASTIVE CHOICE), Table 2 reports “gen/lm” scores. The gen view measures end-to-end instruction following and answer extraction from generated text, while the lm view probes next-token preferences under a constrained decision prompt and supports probability-based analyses. Several models exhibit large gen–lm gaps (e.g., Mixtral-v0.2 and Phi-3-mini on validation tasks), suggesting that a model can be competitive in free-form generation yet brittle when decisions are evaluated through token-

level probability mass. Other models show closer agreement between gen and lm (e.g., Qwen-2.5), indicating more stable decision behavior across the two complementary evaluation views. For models where lm results are unavailable (shown as “–”), only gen is reported due to API/logprob constraints.

6.5 Domain-specific chemistry models

Despite domain specialization, the evaluated chemistry-focused models underperform strong general-purpose LLMs on Primary-Overall. One plausible explanation is that CHEMREASON-BENCH stresses procedural consistency across heterogeneous evidence and output constraints (schemas, legends/placeholders), which are not guaranteed to improve from chemistry pretraining alone. Notably, KALE-8B shows competitive performance on some discriminative subtasks (e.g., Contrast under lm), but still falls short on the structured reconstruction and explanation-style tasks that require combining procedural control with faithful, constraint-aware generation.

Summary. Overall, CHEMREASON-BENCH differentiates models not only by chemistry knowl-

Model	Size	Overall	Order	Contrast	Step-Val	Cond-Val	Step-Comp	Ration.
		↑	PairAcc.↑	Top1Acc.↑ (gen/lm)	F1+↑ (gen/lm)	F1+↑ (gen/lm)	SC-Score↑	CovF1↑
Open-source Models								
LLama-2	7B	38.58	52.27	24.05 / 14.76	66.84 / 51.88	74.93 / 74.70	6.04	19.57
Mixtral-v0.2	7B	41.07	64.58	55.71 / 52.65	69.43 / 31.50	80.55 / 23.88	4.68	20.28
Phi-3-mini	7B	43.51	79.83	59.24 / 51.07	72.46 / 16.98	83.33 / 20.39	5.30	24.19
Qwen-2.5	7B	53.94	78.74	63.42 / 65.65	75.10 / 69.35	80.85 / 85.06	7.25	17.96
LLama-3.1	8B	49.45	73.14	62.58 / 61.37	70.90 / 45.45	80.80 / 61.79	9.23	22.87
Gemma-2	9B	56.74	83.92	67.50 / 70.75	69.51 / 75.34	82.09 / 82.49	11.54	21.15
LLama-3.1	70B	56.07	88.55	77.25 / 81.24	62.62 / 44.09	82.71 / 63.27	16.06	26.21
Qwen-2.5	72B	64.65	91.32	77.25 / 80.50	80.47 / 78.36	88.60 / 85.88	25.73	25.31
DeepSeek-v3.2	685B	65.21	91.53	81.52 / -	70.60 / -	79.93 / -	36.32	31.33
Proprietary Models								
Claude-Sonnet-4	-	66.56	94.00	73.82 / -	76.43 / -	80.73 / -	43.63	30.74
Gemini-2.5-Flash	-	67.10	92.46	82.27 / -	76.45 / -	86.89 / -	35.74	28.83
GPT-4o	-	65.23	92.90	83.94 / 85.89	82.54 / 74.73	87.21 / 79.48	23.72	27.87
GPT-5.2	-	70.30	94.40	88.67 / 88.77	80.16 / 66.18	83.74 / 85.07	51.65	29.45
Grok-4	-	66.57	90.02	78.09 / 78.46	81.76 / 77.87	87.01 / 84.56	35.48	30.05
Qwen-3-max	-	65.08	93.78	79.20 / 82.36	73.31 / 76.24	78.03 / 82.26	30.54	30.46
Domain-specific Models								
ChemLLM	7B	36.31	56.93	44.38 / 52.46	28.13 / 41.66	45.09 / 70.02	4.19	15.84
LlaSMol	7B	14.37	0.50	1.58 / 35.38	61.78 / 0.00	70.09 / 0.00	0.00	1.31
KALE	8B	49.21	76.25	60.17 / 66.57	57.78 / 62.87	65.11 / 77.97	5.26	18.52
ChemDFM	13B	22.28	14.86	25.72 / 39.55	62.76 / 13.31	73.15 / 6.22	0.60	7.88

Table 2: Overall performance on CHEMREASON-BENCH. Each cell reports *composite score / primary metric* for each task (see Section 5 for detailed definitions and normalization).

edge, but also by their ability to maintain procedurally consistent, constraint-aware reasoning across task formats, which is critical for synthesis assistants and automation-oriented settings.

We note that this experiment is conducted in a placeholder-normalized setting and therefore evaluates procedure-level reasoning under controlled entity and condition binding assumptions, rather than end-to-end understanding of free-form experimental prose.

7 From Evaluation to Fine-Tuning

Although CHEMREASON-BENCH is designed for diagnostic evaluation, the same verifiable task formulations provide a practical interface for adapting LLMs into synthesis-oriented assistants. We use the templates, unified schema, and validators to instantiate large-scale training supervision, enabling both instruction tuning and reward-based optimization with automatic checking.

7.1 Training Data: CHEMREASON-TUNE

We instantiate a $> 20k$ -reaction training pool into $> 120k$ verifiable task instances (CHEMREASON-TUNE). To prevent leakage, we split at the reaction-record level so all instances derived from the same reaction remain in the same split. Construction and

split details are provided in Appendix H.

7.2 Objectives: SFT and GRPO

We fine-tune in two stages. SFT trains models to follow task instructions and produce schema-valid structured outputs. We then apply GRPO-style optimization using verifiable rewards computed by deterministic parsers/validators aligned with each task (Appendix Table 14), reinforcing constraint-satisfying decisions, structured reconstruction, and placeholder consistency. Training dynamics, including reward and KL trajectories, are shown in Appendix Figs. 11. Objective definitions and implementation details are deferred to Appendix H.

7.3 Controlled Adaptation Results

Table 15 reports controlled adaptation results on three open-source backbones. SFT yields substantial gains in primary overall across all three models. The fine-tuned Gemma-2-9B reaches performance comparable to proprietary systems on this benchmark, while a gap remains to the strongest model. Adapted models achieve the best results on COND-VAL and STEP-COMP, indicating that schema-constrained decision making and structured reconstruction benefit strongly from automatically checkable supervision. GRPO yields limited improve-

ments beyond SFT in our setting. Appendix reward and KL curves suggest that GRPO training remains reasonably stable, but does not translate into consistent gains over SFT on the benchmark.

Overall, these results suggest a practical deployment trade-off: compact fine-tuned models can be competitive for procedure reasoning at substantially lower cost and with greater accessibility for laboratory and automation workflows.

8 Conclusions

We introduced CHEMREASON-BENCH, a human-validated benchmark for verifiable experimental procedure reasoning in organic synthesis, together with CHEMREASON-TUNE for controlled adaptation studies. Across open-source, proprietary, and chemistry-focused models, results show substantial task-wise variation, indicating that aggregate scores can mask failures in procedure-level consistency, constraint satisfaction, and format faithfulness. The benchmark’s multi-task design provides a diagnostic view of complementary procedural competencies under an explicit schema with placeholders and automatic validators. Using the same verifiable interface, we conducted controlled adaptation experiments. Supervised fine-tuning substantially improves compact open models and can bring them to performance comparable to proprietary systems on parts of the benchmark, while additional preference optimization yields limited gains in our setting. Overall, CHEMREASON-BENCH provides a verifiable framework for evaluating and improving language models intended to generate or verify executable synthesis procedures.

Limitations

Data quality and representation. Although each benchmark instance is human-validated, our inputs are drawn from multiple upstream sources with varying reporting conventions and occasional underspecification. Our canonical representation (action-tagged steps with explicit placeholders) is designed to improve structural consistency, support machine-checkable outputs, and enable fine-grained diagnosis of procedural reasoning errors. However, this representation also narrows the scope of what is evaluated. In particular, experiments conducted on placeholder-normalized texts do not measure end-to-end understanding of authentic free-form experimental prose, such as entity recognition, coreference resolution, implicit condition recovery,

or free-text-to-placeholder binding. As a result, these settings should be interpreted as evaluating procedure-level decision making under controlled normalization assumptions, rather than full end-to-end comprehension of raw laboratory narratives. Moreover, the representation may not capture all nuances in real experimental writing, such as implicit assumptions, partially specified quantities, or context that is routine for practitioners but not explicit in the record.

Scope of procedures and task coverage.

CHEMREASON-BENCH targets organic synthesis procedures and a fixed set of six task families chosen to yield verifiable outputs under an explicit schema. While these tasks cover complementary procedural competencies that are central to many synthesis protocols, they are not intended to span the full breadth of synthesis practice. For example, multi-step campaigns, process optimization, analytical verification, safety-critical exception handling, and domains beyond small-molecule organic chemistry (e.g., biochemistry, materials, electrochemistry) are not explicitly modeled in the current benchmark. Extending coverage along these dimensions is a natural direction for future work.

Training findings may not generalize. Our controlled adaptation results use a particular training recipe (LoRA-based SFT followed by GRPO-style optimization) and one reward specification aligned with our task validators. In this setting, GRPO provides smaller marginal gains beyond SFT; however, this outcome may vary with reward design, sampling strategy, compute budget, base model choice, and hyperparameter tuning, and should be interpreted as an empirical observation within our current configuration rather than a general claim about preference optimization for chemistry.

Potential risks. Models evaluated on CHEMREASON-BENCH may be over-trusted in automation-oriented settings despite remaining failure modes in constraint satisfaction and schema faithfulness. The benchmark does not certify execution safety, and outputs should be used with appropriate human oversight and laboratory safeguards. Dataset artifacts and validator-defined schemas may also bias models toward superficially valid but practically incomplete procedures.

Acknowledgments

This work was jointly supported by the Shanghai Municipal Science and Technology Explorers Program (24TS1403300), the National Natural Science Foundation of China (72432007), and the Fundamental Research Funds for the Central Universities. The computations in this paper were run on the AI for Science Platform, supported by the Artificial Intelligence Institute at Shanghai Jiao Tong University.

References

- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Qianxiang Ai, Fanwang Meng, Jiale Shi, Brenden Pelkie, and Connor W. Coley. 2024. [Extracting structured data from organic synthesis procedures using a fine-tuned large language model](#). *Digital Discovery*, 3(9):1822–1831.
- Anthropic. 2025. [System card: Claude Opus 4 & Claude Sonnet 4](#). Accessed: 2025-12-19.
- Daniil A. Boiko, Robert MacKnight, Benjamin Kline, and Gabriel Gomes. 2023. [Autonomous chemical research with large language models](#). *Nature*, 624(7992):570–578.
- Marcus Bran, Samuel Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. [Augmenting large language models with chemistry tools](#). *Nature Machine Intelligence*, 6(5):525–535.
- Glenn W. Brier. 1950. [Verification of forecasts expressed in terms of probability](#). *Monthly Weather Review*, 78(1):1–3.
- Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M. Alston, Buyi Li, Rob Clowes, Nicola Rankin, Brandon Harris, Reiner Sebastian Sprick, and Andrew I. Cooper. 2020. [A mobile robotic chemist](#). *Nature*, 583(7815):237–241.
- Xiuying Chen, Tairan Wang, Taicheng Guo, Kehan Guo, Juexiao Zhou, Haoyang Li, Zirui Song, Xin Gao, and Xiangliang Zhang. 2025. [Unveiling the power of language models in chemical research question answering](#). *Communications Chemistry*, 8(1):4.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Weichen Dai, Yezeng Chen, Zijie Dai, Yubo Liu, Zhijie Huang, Yixuan Pan, Baiyang Song, Chengli Zhong, Xinhe Li, Zeyu Wang, Zhuoying Feng, and Yi Zhou. 2026. [Kale-lm-chem: Vision and practice toward an ai brain for chemistry](#). *Preprint*, arXiv:2409.18695.
- Jesse Davis and Mark H. Goadrich. 2006. [The relationship between precision-recall and ROC curves](#). In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. [DeepSeek-v3.2: Pushing the frontier of open large language models](#). *Preprint*, arXiv:2512.02556.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. [What can large language models do in chemistry? A comprehensive benchmark on eight tasks](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 59662–59688. Curran Associates, Inc.
- James A. Hanley and Barbara J. McNeil. 1982. [The meaning and use of the area under a receiver operating characteristic \(ROC\) curve](#). *Radiology*, 143(1):29–36.
- Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel,

- Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 399 others. 2024. [GPT-4o system card](#). *Preprint*, arXiv:2410.21276.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of experts](#). *Preprint*, arXiv:2401.04088.
- Steven M. Kearnes, Michael R. Maser, Michael Wlekinski, Adam Kast, Abigail G. Doyle, Spencer D. Dreher, John M. Hawkins, Klavs F. Jensen, and Connor W. Coley. 2021. [The open reaction database](#). *Journal of the American Chemical Society*, 143(45):18820–18826.
- Hanzheng Li, Xi Fang, Yixuan Li, Chaozheng Huang, Junjie Wang, Xi Wang, Hongzhe Bai, Bojun Hao, Shenyu Lin, Huiqi Liang, Linfeng Zhang, and Guolin Ke. 2026. [Rxnbench: A multimodal benchmark for evaluating large language models on chemical reaction understanding from scientific literature](#). *Preprint*, arXiv:2512.23565.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Zhiyuan Liu, Yaorui Shi, An Zhang, Sihang Li, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. 2024. [ReactXT: Understanding molecular “reaction-ship” via reaction-contextualized molecule-text pretraining](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5353–5377.
- Daniel M. Lowe. 2017. [Chemical reactions from US patents \(1976–Sep 2016\)](#). Figshare dataset. Accessed: 2025-12-19.
- Sam H. M. Mehr, Mark Craven, Andrei I. Leonov, Gregory Keenan, and Leroy Cronin. 2020. [A universal system for digitization and automatic execution of the chemical synthesis literature](#). *Science*, 370(6512):101–108.
- Anas Mirza, Nithin Alampara, Sandeep Kunchapu, Syed Moin, Jianyang Guo, Seyed Ali Chithrananda, and Connor W. Coley. 2025. [A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists](#). *Nature Chemistry*, pages 1–8.
- OpenAI. 2025. [Update to GPT-5 system card: GPT-5.2](#). Accessed: 2025-12-19.
- Organic Syntheses, Inc. 2025. [Organic syntheses](#). Accessed: 2025-12-19.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. [BLEU: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. [GPQA: A graduate-level google-proof q&a benchmark](#). *Preprint*, arXiv:2311.12022.
- Nicholas T. Runcie, Charlotte M. Deane, and Fergus Imrie. 2025. [Assessing the chemical intelligence of large language models](#). *Preprint*, arXiv:2505.07735.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram e, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Qwen Team, An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chu-jie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, and 42 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Alain C. Vaucher, Philippe Schwaller, Joppe Geluykens, Vishnu H. Nair, Anna Iuliano, and Teodoro Laino. 2021. [Inferring experimental procedures from text-based representations of chemical reactions](#). *Nature Communications*, 12(1):2573.
- Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. 2020. [Automated extraction of chemical synthesis actions from experimental procedures](#). *Nature Communications*, 11(1):3601.

- xAI. 2025. [Grok 4 fast model card](#). Accessed: 2025-12-19.
- Jiaqing Xie, Weida Wang, Ben Gao, Zhuo Yang, Haiyuan Wan, Shufei Zhang, Tianfan Fu, and Yuqiang Li. 2025. [Qcbench: Evaluating large language models on domain-specific quantitative chemistry](#). *Journal of Chemical Information and Modeling*, 65(22):12268–12278.
- Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. [LlaSMol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset](#). *Preprint*, arXiv:2402.09391.
- Zheni Zeng, Yi-Chen Nie, Ning Ding, Qian-Jun Ding, Wei-Ting Ye, Cheng Yang, Maosong Sun, Weinan E, Rong Zhu, and Zhiyuan Liu. 2023. [Transcription between human-readable synthetic descriptions and machine-executable instructions: an application of the latest pre-training technology](#). *Chemical Science*, 14(35):9360–9373.
- Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. 2024. [ChemLLM: A chemical large language model](#). *Preprint*, arXiv:2402.06852.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.
- Yu Zhang, Yang Han, Shuai Chen, Ruijie Yu, Xin Zhao, Xianbin Liu, Kaipeng Zeng, Mengdi Yu, Jidong Tian, Feng Zhu, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. 2025a. [Large language models to accelerate organic chemistry synthesis](#). *Nature Machine Intelligence*, 7:1010–1022.
- Yu Zhang, Ruijie Yu, Jidong Tian, Feng Zhu, Jiapeng Liu, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. 2025b. [ChemActor: Enhancing automated extraction of chemical synthesis actions with LLM-generated data](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24291–24314, Vienna, Austria. Association for Computational Linguistics.
- Zehua Zhao, Zhixian Huang, Junren Li, Siyu Lin, Junting Zhou, Fengqi Cao, Kun Zhou, Rui Ge, Tingting Long, Yuexiang Zhu, Yan Liu, Jie Zheng, Junnian Wei, Rong Zhu, Peng Zou, Wenyu Li, Zekai Cheng, Tian Ding, Yaxuan Wang, and 12 others. 2025a. [SU-PERChem: A multimodal reasoning benchmark in chemistry](#). *Preprint*, arXiv:2512.01274.
- Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, Kai Yu, and Xin Chen. 2025b. [Developing ChemDFM as a large language foundation model for chemistry](#). *Cell Reports Physical Science*, 6(4):102523.
- Xianrui Zhong, Yufeng Du, Siru Ouyang, Ming Zhong, Tingfeng Luo, Qirong Ho, Hao Peng, Heng Ji, and Jiawei Han. 2024. [ActionIE: Action extraction from scientific literature with programming languages](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12656–12671, Bangkok, Thailand. Association for Computational Linguistics.

A Code and Data Availability

The code, data, and related project resources are available at <https://github.com/Khadaz/ChemReason-Bench>.

B Comparison with chemistry LLM benchmarks

Existing chemistry LLM benchmarks have played an important role in evaluating chemical knowledge, problem solving, and multimodal understanding. However, most of them primarily assess answer correctness in QA-style settings, with limited emphasis on structured procedural decision making. As shown in Table 3, CHEMREASON-BENCH is designed to fill this gap by targeting reasoning over experimental procedures with schema-constrained, machine-verifiable outputs, enabling evaluation not only of whether an answer is correct, but also of whether it is operationally consistent and automatically checkable.

C Introduction of Six Task Types

CHEMREASON-BENCH evaluates whether language models can reliably reason over organic synthesis procedures beyond factual chemistry recall. Experimental protocols combine heterogeneous evidence, such as free-text narratives, structured reaction records, and action-tagged step sequences, and require models to maintain procedural consistency, chemical plausibility, and faithful entity–role grounding. We now define the six task types used in CHEMREASON-BENCH. Each task corresponds to a target capability profile (Fig. 2) and has a verifiable output structure.

Ordering (Procedural Order). **Goal:** Given a local procedural context, determine the correct chronological order of several candidate operations. **Input:** Text context plus a small set of candidate steps (with `step_ids`) and the legend when placeholders appear. **Output:** An ordered list of `step_ids`. **Rationale:** Probes procedural dependency reasoning (e.g., *wash* → *brine* → *dry*, or complete reaction → workup → purification). **Evaluation:** Exact match and rank-based measures (e.g., pairwise accuracy and normalized Kendall- τ) quantify both strict correctness and near-misses.

Step Completion (Missing Step). **Goal:** Fill in one missing atomic operation that must be con-

sistent with the surrounding steps. **Input:** A partial action-grounded sequence in natural language with one position masked; optionally the schema/legend and slot constraints. **Output:** A single action (and slots when applicable) describing the missing step. **Rationale:** Tests constrained generation: the model must output exactly one atomic step that respects the procedure state and remains faithful to placeholders. **Evaluation:** Action exact match, slot F1, and format-error rate measure correctness and constraint compliance.

Step Validation (Action Validity). **Goal:** Decide whether a proposed procedural action is appropriate and chemically plausible in context. **Input:** A context window and a candidate action statement (with the legend as needed). **Output:** Binary label (True/False). **Rationale:** Targets local procedural common sense and feasibility (e.g., skipping filtration before concentration, drying an aqueous layer, incompatible additions). **Evaluation:** F1/Accuracy and calibration metrics (e.g., Brier, ECE), plus AUROC/AUPRC for imbalance settings.

Condition Validation (Condition Reasoning). **Goal:** Judge whether specified reaction conditions (e.g., temperature, duration, atmosphere, dryness) are reasonable for the described step. **Input:** Text context including condition tokens and legend bindings. **Output:** Binary label (True/False). **Rationale:** Conditions determine feasibility and selectivity; validating them requires linking procedural intent to correct operating windows. **Evaluation:** As in Step Validation, with emphasis on calibrated decision quality.

Contrastive Choice (Semantic Grounding under Options). **Goal:** Select the correct option among multiple candidates (e.g., reagent role, limiting reagent, correct workup component), with plausible distractors. **Input:** Context + options (+ legend). **Output:** An option index (and optionally the chosen token). **Rationale:** Tests semantic grounding and role recognition: mapping placeholders to functional roles and selecting the chemically appropriate choice under controlled alternatives. **Evaluation:** Top-1 accuracy and probabilistic metrics such as log loss/ECE (and ranking metrics when applicable).

Rationalization (Reaction Rationale / Explanation). **Goal:** Produce a short explanation of *why*

Benchmark	Primary focus	Task format	Scale	Verifiable outputs
CHEMREASON-BENCH (ours)	Procedure-level reasoning under explicit schema/placeholder constraints	6 tasks (ordering/validation/completion/choice/rationalization), structured I/O	500 rxns \rightarrow 7,306 tasks	Schema-parseable outputs + automatic validators
ChemBench (Mirza et al., 2025)	Broad chemistry knowledge & reasoning (vs. chemists)	QA/MCQ + open-ended	>2.7k QA pairs	Answer-key based (not procedure schema constraints)
RxnBench (Li et al., 2026)	Multimodal reaction understanding from literature PDFs	Single-figure QA + full-document QA	1,525 Q (SF-QA) + 108 papers (FD-QA)	Answer-key based; focuses on multimodal perception/reasoning
SUPERChem (Zhao et al., 2025a)	Reasoning-intensive chemistry problems (incl. multimodal)	Problem solving / reasoning-path style evaluation	500 problems	Typically answer-key based (not schema-valid structured decisions)
QCBench (Xie et al., 2025)	Quantum chemistry subfields	QA/problem solving	350 problems	Answer-key based
ChemIQ (Runcie et al., 2025)	Organic chemistry QA/reasoning	QA	796 questions	Answer-key based
ChemLLMBench (Guo et al., 2023)	What LLMs can do in chemistry (multi-task)	Benchmark suite across multiple task types	Eight tasks (suite-style evaluation)	Mostly answer-key based

Table 3: Comparison to representative chemistry LLM benchmarks. Most existing benchmarks emphasize QA-style evaluation with answer keys, whereas CHEMREASON-BENCH targets procedure-level decision making with schema-parseable outputs and automatic validation of many operational constraints.

an action, condition, or choice is correct in the procedure. **Input:** Context and the decision target (e.g., why include an additive, why low temperature, why this wash). **Output:** A natural-language rationale constrained to the provided context and placeholders. **Rationale:** Probes causal explanation that connects procedural choices to operational or feasibility reasons (e.g., drying agents remove water; base enables coupling; brine assists phase separation). Different from full mechanism derivation, we treat this as reaction feasibility / condition-cause explanation rather than complete mechanistic proof. **Evaluation:** Coverage-oriented and semantic similarity measures.

D Example Prompt–Answer Pairs (Benchmark id=319)

Table 4: One example per task type. Each row shows the prompt (input) and the corresponding answer (ground truth).

Task	Prompt / Answer
Ordering	<p>Prompt:</p> <pre>{ "version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "ordering_319_1", "task_type": "ordering", "context": "Initial reagent setup before starting the coupling reaction.", "question": "Arrange the following operations (identified by step_id) in the correct chronological order.", "steps_to_order": [{ "step_id": "0", "description": "Stir a mixture of \$1\$, \$2\$, and \$5\$ in \$3\$/\$4\$ to obtain a homogeneous suspension." }, { "step_id": "1", "description": "Add the palladium catalyst \$7\$ to the reaction mixture." }, { "step_id": "2", "description": "Heat the reaction mixture to reflux to initiate the Suzuki coupling." }], "options": null, "legend": { "\$1\$": "2-dimethylaminomethyl-bromobenzene", "\$2\$": "4-(4,4,5,5-tetramethyl-1,3,2-dioxaborolan-2-yl)aniline", "\$3\$": "toluene", "\$4\$": "n-BuOH", "\$5\$": "Cs2CO3", "\$7\$": "Pd(PPh3)4" } }</pre> <p>Answer:</p> <pre>{ "task_id": "ordering_319_1", "task_type": "ordering", "ground_truth": { "correct_order": ["0", "1", "2"], "gold_explanation": "Reagents and base must be present before the catalyst is introduced; heating to reflux can only commence after all coupling components are in solution." } }</pre>
Step Completion	<p>Prompt:</p> <pre>{ "version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "step_completion_319_1", "task_type": "step_completion", "context": "Identify the missing reagent addition at this point.", "question": {"before_action": "ADD", "before": "Add \$8\$ to the reaction mixture.", "after_action": "PHASESEPARATION", "after": "Allow the mixture to separate into distinct phases."}, "steps_to_order": null, "options": null, "legend": { "\$6\$": "H2O", "\$8\$": "EtOAc" } }</pre> <p>Answer:</p> <pre>{ "task_id": "step_completion_319_1", "task_type": "step_completion", "ground_truth": { "action": "ADD", "slots": { "reagent": "\$6\$", "description_private": "Add \$6\$ to the reaction mixture." } } }</pre>
Step Validation	<p>Prompt:</p> <pre>{ "version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "step_validation_319_2", "task_type": "step_validation", "context": "Immediately after phase separation.", "question": "Is it acceptable to skip the filtration step and proceed directly to concentration of the organic phase?", "steps_to_order": null, "options": null, "legend": {} }</pre> <p>Answer:</p> <pre>{ "task_id": "step_validation_319_2", "task_type": "step_validation", "ground_truth": { "label": false, "explanation_private": "Particulate palladium black and Cs2CO3 fines can remain suspended; omitting filtration would carry solids into the evaporator, risking contamination and bumping." } }</pre>
Condition Validation	<p>Prompt:</p> <pre>{ "version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "condition_validation_319_1", "task_type": "condition_validation", "context": "Reflux in \$3\$/\$4\$ for 4 h.", "question": "Is a 4-hour reflux a reasonable duration for this Pd-catalysed Suzuki coupling?", "steps_to_order": null, "options": null, "legend": { "\$3\$": "toluene", "\$4\$": "n-BuOH" } }</pre> <p>Answer:</p> <pre>{ "task_id": "condition_validation_319_1", "task_type": "condition_validation", "ground_truth": { "label": true, "explanation_private": "Typical Suzuki reactions in heterogeneous solvent mixtures often require 2-6 h at reflux to reach completion; 4 h is within standard practice." } }</pre>
Contrastive Choice	<p>Prompt:</p> <pre>{ "version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "contrastive_choice_319_1", "task_type": "contrastive_choice", "context": "A Suzuki-Miyaura coupling is performed using \$1\$ (2.75 mmol) and \$2\$ (2.56 mmol) with \$5\$ (7.67 mmol) as base and \$7\$ (0.17 mmol) as catalyst in toluene/n-BuOH.", "question": "Which reagent is the limiting reagent in this coupling based on the stoichiometry given?", "steps_to_order": null, "options": ["\$1\$", "\$2\$", "\$5\$", "\$7\$"], "legend": { "\$1\$": "2-dimethylaminomethyl-bromobenzene", "\$2\$": "4-(4,4,5,5-tetramethyl-1,3,2-dioxaborolan-2-yl)aniline", "\$5\$": "Cs2CO3", "\$7\$": "Pd(PPh3)4" } }</pre> <p>Answer:</p> <pre>{ "task_id": "contrastive_choice_319_1", "task_type": "contrastive_choice", "ground_truth": { "correct_option_idx": 1, "correct_choice": "\$2\$", "distractor_explanations": { "\$1\$": "Present in slight excess (2.75 mmol), not limiting.", "\$5\$": "Base is used in large excess (7.67 mmol).", "\$7\$": "Pd catalyst at 0.17 mmol is catalytic, not consumed stoichiometrically." } } }</pre>

Task	Prompt / Answer
Rationalization	<p>Prompt:</p> <pre data-bbox="453 282 1445 367">{"version": "v1.0", "split": "test", "benchmark_id": 319, "task_id": "rationalization_319_2", "task_type": "rationalization", "context": "Chromatography uses 1% \$11\$ in the eluent.", "question": "What is the purpose of adding 1% \$11\$ to the \$8\$/hexanes mobile phase?", "steps_to_order": null, "options": null, "legend": {"\$11\$": "TEA", "\$8\$": "EtOAc"}}</pre> <p>Answer:</p> <pre data-bbox="453 405 1445 468">{"task_id": "rationalization_319_2", "task_type": "rationalization", "ground_truth": {"gold_rationale": "TEA suppresses silica surface acidity, preventing tailing and adsorption of the basic aniline product, which would otherwise give broad or irreproducible bands."}}</pre>

E Dataset Construction

E.1 From structured procedure records to natural-language step renderings

We render the canonicalized action sequences into action-grounded natural-language steps, pairing each step with an explicit action label from a fixed inventory (27 action types after integration and canonicalization in our datasets). This natural-language rendering provides a common, human-readable interface that aligns with instruction-tuned LLMs, enriches semantic cues (roles, intent, and procedural context), and supports both evaluation and training in realistic assistant settings. At the same time, the action tags and placeholder bindings keep the representation structured enough to allow strict parsing and machine-checkable constraints when instantiating tasks.

PII and offensive content screening. We screen source text for potential personally identifying information (PII) and offensive content using simple pattern-based filters (e.g., email/URL/phone-number patterns) and spot checks. Any detected PII is removed during canonicalization; we release only processed procedure text and structured representations.

E.2 Authoring system: rules, LLM-assisted instantiation, and quality control

The authoring system combines deterministic processing with limited, schema-constrained LLM assistance (Fig. 3, Sector 2). Rule-based modules perform action consolidation, enforce placeholder–legend consistency, and apply task-specific validity constraints, ensuring that each instance admits a machine-checkable target. On top of this normalized backbone, LLM-assisted instantiation is restricted to surface realization and candidate generation. It renders steps and prompts in more natural language and diversifies prompt phrasing while preserving the underlying procedural constraints and output space under a unified schema.

All instances first undergo automatic validation for structural and content consistency. These checks include schema conformity for each task type, completeness of required fields, abnormal text length, consistency with the underlying experimental record, and duplicate or missing instance detection. Any violation triggers re-annotation.

Beyond structural validation, we apply task-specific quality control through manual review.

For each task type, we randomly sample 50 instances from distinct experiments. Annotators verify: (i) consistency between the context and the corresponding original experimental record, (ii) whether the question is grounded in and logically derived from the record, (iii) correctness of the answer with respect to both the experimental record and established chemical knowledge, and (iv) correct and semantically aligned usage of task-specific placeholders, including binding consistency. For ORDERING and STEP COMPLETION, scripts additionally verify that the constructed ground-truth ordering and completion target are consistent with the corresponding experimental record, and sampled instances are required to satisfy 100% correctness. For STEP VALIDATION, CONDITION VALIDATION, CONTRASTIVE CHOICE, and RATIONALIZATION, sampled instances are required to satisfy a 90% acceptance threshold. In these latter cases, LLM-assisted checks are used only as auxiliary filtering to assist human review and do not replace manual inspection.

Re-annotation is triggered whenever sampled instances fail the corresponding acceptance threshold or automatic validation identifies structural inconsistencies. All revised instances are double-checked by a second annotator, and disagreements are resolved through discussion. Overall, the revision rate is approximately 9%. Task-specific acceptance thresholds and approximate revision rates are summarized in Table 5.

To quantify subjectivity in validity-focused judgments, we independently double-annotated randomly sampled subsets of STEP VALIDATION and CONDITION VALIDATION instances (N=50 each). We report Cohen’s $\kappa = 0.72$ for STEP VALIDATION and $\kappa = 0.80$ for CONDITION VALIDATION, indicating substantial agreement. Most disagreements arose from borderline cases involving procedural plausibility or the reasonableness of stated conditions, rather than schema-level inconsistencies, and were resolved through discussion in the final adjudicated version.

E.3 Benchmark and training pools

We construct two reaction pools (Fig. 3, Sector 1): a 500-reaction benchmark pool for controlled evaluation, and a larger training pool of > 20k reactions for downstream model adaptation.

CHEMREASON-BENCH (evaluation instances). Using a human-designed template library with a

Table 5: Task-specific quality control and approximate revision rates.

Task	Sample	Threshold	Rev. rate
ORDER	50	100%	~5%
STEP-VAL	50	$\geq 90\%$	~9%
COND-VAL	50	$\geq 90\%$	~11%
STEP-COMP	50	100%	~6%
CONTRAST	50	$\geq 90\%$	~8%
RATIONA.	50	$\geq 90\%$	~16%
Overall	—	—	~9%

unified schema (Fig. 3, Sector 2), we instantiate CHEMREASON-BENCH from the benchmark pool, yielding 7306 verifiable task instances spanning six task types (Fig. 6).

CHEMREASON-TUNE (training instances). Using the same template library as a blueprint, we instantiate a substantially larger training set from the > 20k-reaction pool (Fig. 3, Sector 3B), resulting in > 120k verifiable task instances. To improve robustness under distribution shifts, we allow controlled re-instantiation by combining LLM generation with rule-based scripts (e.g., varying surface realizations while preserving underlying constraints), and serialize the resulting instances into standard formats for instruction tuning (SFT) and preference/format training (for GRPO).

Usage in this paper. Unless otherwise specified, all evaluation results in the main paper are reported on CHEMREASON-BENCH, while the training pool is used only for adaptation and auxiliary analyses.

Template-based instantiation. Each task template specifies the evidence presented to the model (context, candidates/options, and constraints such as a schema and legend), the required output type, and the evaluation protocol. Brief examples are illustrated in Fig. 1, with additional instances and formatting details deferred to Appendix D.

Intended use. CHEMREASON-BENCH is intended for research on procedure-level reasoning and evaluation; it is not a certification of laboratory execution safety.

E.4 Chemical composition and coverage analysis

Beyond reporting scale, we characterize the chemical and procedural coverage of the benchmark using heatmaps that cross-tabulate reaction types

(columns) against key chemical facets (rows); the reaction-type taxonomy accounts for the vast majority of benchmark records, with a catch-all Other category for rare or ambiguous cases.

Concretely, we report reaction-type-conditioned frequencies over: (i) fine-grained product functional groups (Fig. 8), (ii) coarse-grained product functional-groups (Fig. 9a), (iii) key reagents (canonical SMILES; Fig. 10b), (iv) solvents (canonical SMILES; Fig. 10a), and (v) work-up reagents (Fig. 9b). Each cell denotes the percentage of reactions within a reaction type that contain the corresponding facet, highlighting both broad coverage (diverse non-zero bands) and specialization (concentrated hotspots).

We summarize record-level coverage using an any-hit criterion: a facet is counted as present if at least one canonical item is recognized in the corresponding field for a record. Under this definition, solvents and product functional groups achieve high coverage in the benchmark pool (e.g., 97.6% and 94.2%), whereas key reagents and work-up reagents are lower (e.g., 28.0% and 43.2%). This gap is expected because (i) “key reagent” is often context-dependent and may not be uniquely identifiable in multi-reagent transformations under a strict operational definition, and (ii) our work-up extraction is conservative due to canonicalization constraints (e.g., a whitelist of reliably normalized reagents). Accordingly, we use these plots as coverage diagnostics rather than completeness claims, making explicit which chemical facets can be extracted and compared consistently across heterogeneous sources.

Conditioning the heatmaps on reaction type (rows) improves interpretability by revealing which reagents/solvents/functional-group patterns concentrate in particular families and how broadly the benchmark spans distinct chemistries.

E.5 Action Types

This appendix first enumerates the action inventory used throughout CHEMREASON-BENCH (Table 6), i.e., the full set of action tags available in our canonical procedure representation. We then report the frequency distribution of *missing* action types in the STEP COMPLETION task: we count each occurrence of an action tag that is masked as the prediction target across all STEP COMPLETION instances, and summarize actions by absolute counts and proportions. This statistic contextualizes the structured reconstruction setting by showing which

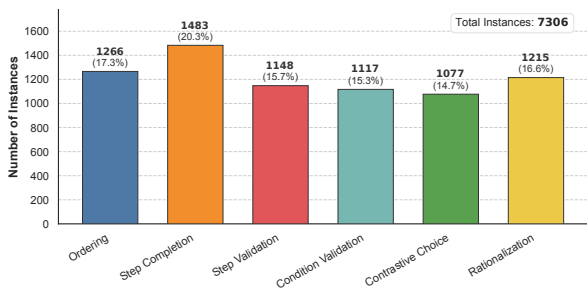


Figure 6: Distribution of CHEMREASON-BENCH task instances across six task types. Numbers above bars indicate counts and the corresponding percentages; the total number of instances is 7306.

ADD	COLLECTLAYER	COLUMN
CONCENTRATE	DEGAS	DISTILL
DRYSOLID	DRYSOLUTION	EVAPORATE
EXTRACT	FILTER	MAKESOLUTION
MICROWAVE	PARTITION	PH
PHASESEPARATION	QUENCH	RECRYSTALLIZE
REFLUX	SETTEMPERATURE	SONICATE
STIR	TRANSFER	TRITURATE
WAIT	WASH	YIELD

Table 6: Action inventory (action space) used in the canonical procedure representation throughout CHEMREASON-BENCH (27 action types).

operations are commonly required versus rare as completion targets.

F Metric Details

F.1 Metrics Overview

Ordering. We use `pairwise_accuracy` as the primary metric, measuring the fraction of correctly ordered step pairs implied by the prediction. We additionally report `exact_match` and `kendall_tau_norm` to capture strict sequence correctness and rank correlation.

Step Completion. We evaluate structured step reconstruction using `step_completion_score`, which averages `action_em` and `slot_f1`. We further report `format_error_rate` to quantify invalid structured outputs (e.g., unparseable or missing required fields).

Step & Condition Validation. Both validation tasks are binary decisions. We report positive-class F1 (`f1_positive`) as the primary metric to balance precision and recall under potential class imbalance. Beyond thresholded accuracy, we include probability-sensitive metrics (`brier` (Brier, 1950), `ece` (Guo et al., 2017)) and threshold-free ranking

metrics (`auroc` (Hanley and McNeil, 1982), `auprc` (Davis and Goadrich, 2006)).

Contrastive Choice. For multiple-choice selection, `top1_accuracy` is the primary metric. We also report `log_loss` and `mrr` to reflect probabilistic ranking quality, together with `top-label_ece` to assess calibration.

Rationalization. Rationales are evaluated primarily by `coverage_f1`. We additionally report overlap-based and embedding-based generation metrics (`rougeL_f1` (Lin, 2004), `bleu` (Papineni et al., 2002), `bert_score_f1` (Zhang et al., 2020)).

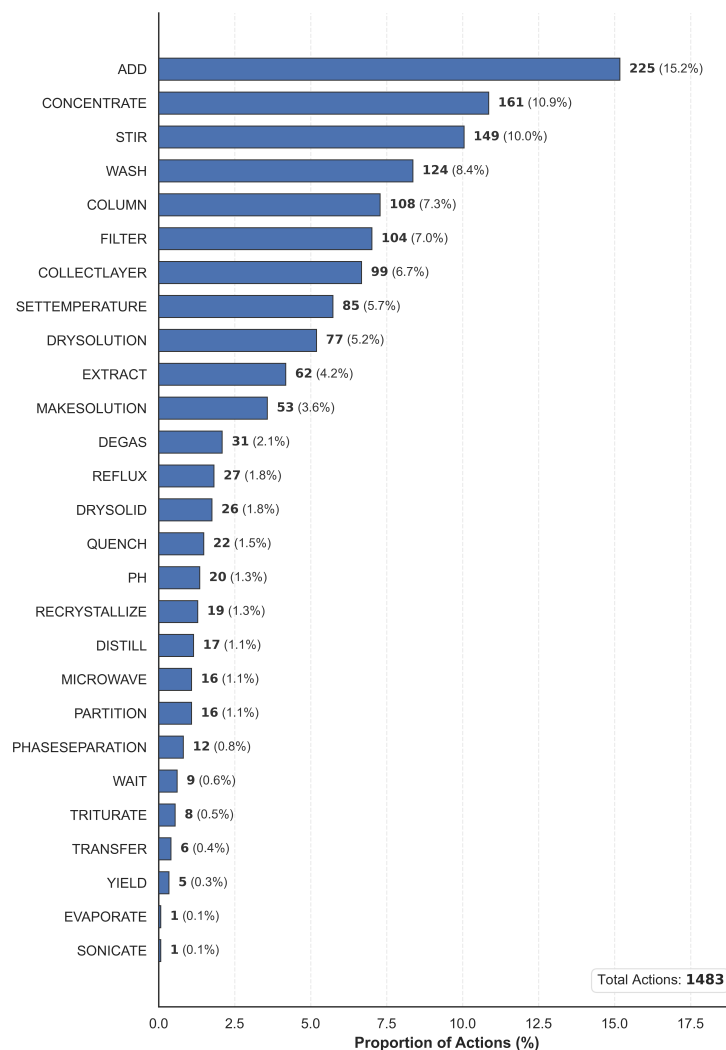


Figure 7: Distribution of missing action types in STEP COMPLETION. Each bar shows the proportion of a given action tag among all *masked* (to-be-predicted) actions (total missing actions: 1483), with the absolute count annotated. Common targets include ADD, CONCENTRATE, and STIR, while long-tail targets (e.g., EVAPORATE, SONICATE) occur rarely.

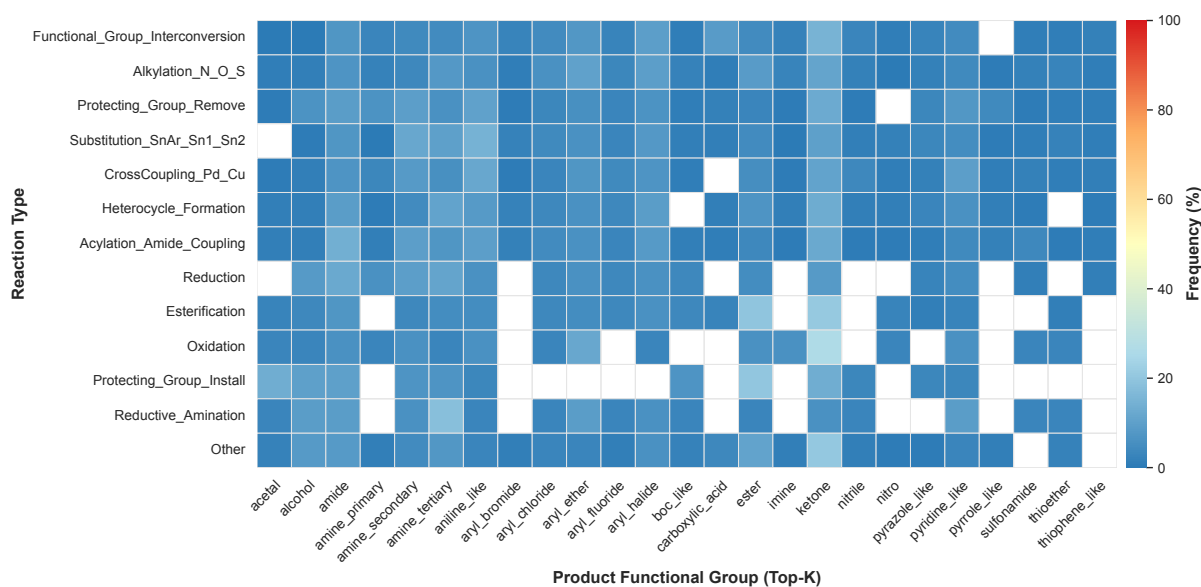
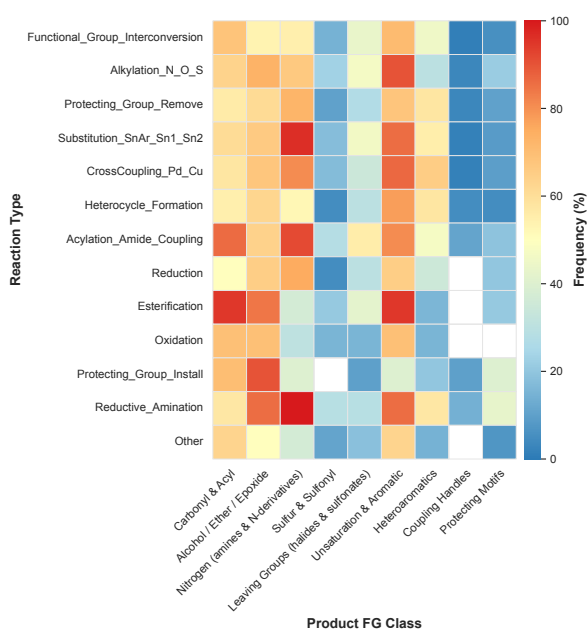
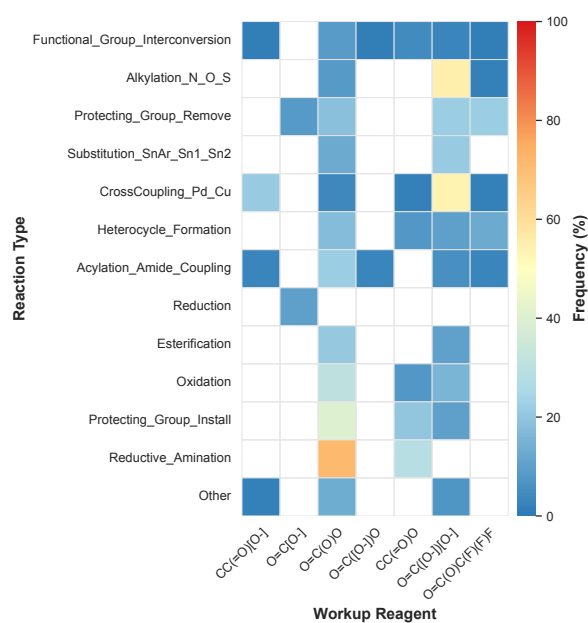


Figure 8: Reaction type \times product functional groups (fine-grained; top-K).



(a) Reaction type \times product functional-group classes (coarse-grained).



(b) Reaction type \times work-up reagents.

Figure 9: Supplementary heatmaps characterizing dataset composition across reaction types (I): functional groups and work-up reagents.

F.2 Metric Formulas Per Task Types

Table 7: Metric formulas used by the evaluator.

Task	Reported metrics	Formulas / definitions
		<p>Notation: gold order $T = [t_1, \dots, t_M]$, prediction $P = [p_1, \dots]$. Filter invalid ids: $P' = \langle p \in P : p \in T \rangle$, $n = P'$, and $\text{pos}_T(x)$ is the index of x in T.</p> <p>Pairwise accuracy:</p>
	pairwise_accuracy (P)	$\text{PA} = \begin{cases} 0, & n < 2 \\ \frac{1}{\binom{n}{2}} \sum_{1 \leq a < b \leq n} \mathbf{1}[\text{pos}_T(P'_a) < \text{pos}_T(P'_b)], & \text{otherwise} \end{cases}$
Ordering	exact_match	Exact match: $\text{EM} = \mathbf{1}[P \equiv T]$.
	kendall_tau_norm	<p>Normalized Kendall-τ:</p> $C = \sum_{a < b} \mathbf{1}[\text{pos}_T(P'_a) < \text{pos}_T(P'_b)], \quad D = \sum_{a < b} \mathbf{1}[\text{pos}_T(P'_a) > \text{pos}_T(P'_b)]$ $\tau = \frac{C - D}{C + D} \quad (C + D > 0), \quad \tau_{\text{norm}} = \frac{\tau + 1}{2} \quad (\text{else } 0).$
	step_completion_score (P)	<p>Let predicted action \hat{a}_i and gold action a_i.</p> <p>Action exact match: $\text{ActionEM} = \frac{1}{N} \sum_i \mathbf{1}[\hat{a}_i = a_i]$.</p> <p>Slot F1: treat filled slots as a set/multiset of fields; compute per instance</p>
	action_em	$F1_i = \frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i} \quad (\text{else } 0), \quad \text{SlotF1} = \frac{1}{N} \sum_i F1_i,$
Step Completion	slot_f1	where TP/FP/FN are counted by matching structured slots under the script's normalization rules.
	format_error_rate	<p>Format error rate: $\text{FER} = \frac{1}{N} \sum_i \mathbf{1}[\text{format/unit invalid}]$.</p> <p>Step-completion score: (as implemented)</p>
		$\text{Raw} = 0.8 \cdot \text{ActionEM} + 0.2 \cdot \text{SlotF1}, \quad \text{SCS} = \text{Raw} \cdot (1 - \text{FER}).$
	f1_positive (P)	<p>Let $y_i \in \{0, 1\}$ be gold, $\hat{y}_i \in \{0, 1\}$ predicted label, $s_i \in [0, 1]$ predicted score, N instances.</p> <p>Accuracy: $\text{Acc} = \frac{1}{N} \sum_i \mathbf{1}[\hat{y}_i = y_i]$.</p> <p>Positive F1: $\text{TP} = \sum_i \mathbf{1}[y_i=1 \wedge \hat{y}_i=1]$, $\text{FP} = \sum_i \mathbf{1}[y_i=0 \wedge \hat{y}_i=1]$, $\text{FN} = \sum_i \mathbf{1}[y_i=1 \wedge \hat{y}_i=0]$,</p>
	accuracy	$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1}^+ = \frac{2PR}{P + R} \quad (\text{else } 0).$
	brier	Brier: $\text{Brier} = \frac{1}{N} \sum_i (s_i - y_i)^2$.
	ece	ECE (10 bins): $b_i = \min(B - 1, \lfloor s_i B \rfloor)$, $B=10$. For bin b : $n_b = \sum_i \mathbf{1}[b_i = b]$, $\text{acc}_b = \frac{1}{n_b} \sum_{i: b_i=b} y_i$, $\text{conf}_b = \frac{1}{n_b} \sum_{i: b_i=b} s_i$,
Step Validation	auroc	$\text{ECE} = \sum_{b=0}^{B-1} \frac{n_b}{N} \text{acc}_b - \text{conf}_b .$
	auprc	<p>AUROC (rank form): with tie-averaged ranks r_i of s_i (ascending), $n_+ = \sum_i y_i$, $n_- = N - n_+$, $S_+ = \sum_{i: y_i=1} r_i$,</p>
		$\text{AUROC} = \frac{S_+ - \frac{n_+(n_++1)}{2}}{n_+ n_-} \quad (\text{undefined if } n_+ = 0 \text{ or } n_- = 0).$
		AUPRC (PR-curve area): sort by s descending; define precision/recall at each prefix and compute trapezoid area over (Rec, Prec) points.
Condition Validation	f1_positive (P) accuracy, brier, ece auroc, auprc	Same as Step Validation (binary classification with score s_i and label y_i): accuracy, f1_positive, brier, ece (10 bins), auroc (rank form), auprc (PR area).

Continued on next page.

Task	Reported metrics	Formulas / definitions
Contrastive Choice		Let gold index $y_i \in \{1, \dots, K\}$, predicted probs $\mathbf{p}_i = (p_{i1}, \dots, p_{iK})$, and predicted top-1 $\hat{y}_i = \arg \max_j p_{ij}$. Top-1 accuracy: $\text{Top1} = \frac{1}{N} \sum_i \mathbf{1}[\hat{y}_i = y_i]$. Log loss: with $\varepsilon = 10^{-15}$, $\tilde{p}_i = \min(\max(p_{i, y_i}, \varepsilon), 1 - \varepsilon)$, $\text{LogLoss} = \frac{1}{N} \sum_i -\log(\tilde{p}_i)$
	top1_accuracy (P)	
	log_loss	MRR: rank options by descending p_{ij} ; let rank_i be the 1-based rank of y_i ,
	mrr	$\text{MRR} = \frac{1}{N} \sum_i \frac{1}{\text{rank}_i}$
	ece	Top-label ECE (10 bins): confidence $c_i = \max_j p_{ij}$, correctness $h_i = \mathbf{1}[\hat{y}_i = y_i]$, bin by $b_i = \min(B - 1, \lfloor c_i B \rfloor)$, $\text{ECE} = \sum_b \frac{n_b}{N} \text{acc}_b - \text{conf}_b , \quad \text{acc}_b = \frac{1}{n_b} \sum_{i: b_i=b} h_i, \quad \text{conf}_b = \frac{1}{n_b} \sum_{i: b_i=b} c_i$
Rationalization	coverage_f1 (P)	Let token multiset of prediction be P_i and gold be G_i after the text normalization. Overlap count: $\text{ov}_i = \sum_w \min(\text{cnt}_{P_i}(w), \text{cnt}_{G_i}(w))$.
	rougeL_f1	Coverage-F1: $p_i = \text{ov}_i / P_i $, $r_i = \text{ov}_i / G_i $, $m_i = \frac{2p_i r_i}{p_i + r_i}$ (else 0), and $\text{CoverageF1} = \frac{1}{N} \sum_i m_i$.
	bleu	ROUGE-L F1: $L_i = \text{LCS}(P_i, G_i)$, $p_i = L_i / P_i $, $r_i = L_i / G_i $, $m_i = \frac{2p_i r_i}{p_i + r_i}$; report mean. BLEU: 1-4gram clipped precision with smoothing + brevity penalty; report mean.
	bert_score_f1	BERTScore-F1: mean of BERTScore F1.

F.3 Evaluation Protocols for Discriminative Tasks: gen vs. lm

CHEMREASON-BENCH contains six task types, among which three are discriminative (STEP VALIDATION, CONDITION VALIDATION, and CONTRASTIVE CHOICE). For these tasks we adopt two complementary inference/evaluation protocols: gen (generation-style decoding) and lm (LM-probing via next-token probabilities). The two protocols provide two views of model behavior: end-to-end generation with instruction following (gen) versus token-level preference and probability assignment (lm).

F.3.1 Protocol A: gen (generation-style decoding)

In gen, the model is allowed to generate a free-form response (optionally including a rationale) and a final decision. During evaluation, we parse the generated text into a discrete label.

Binary validation. For STEP/CONDITION VALIDATION, the target is a binary label in {YES, NO}. We parse the final decision from an explicit answer field when present (e.g., <answer>YES</answer>), otherwise from the first unambiguous occurrence of YES/NO in the output.

Multiple-choice selection. For CONTRASTIVE CHOICE, the model may output either an integer index in $\{0, \dots, K-1\}$ or an option string that can be mapped back to an index. We parse the predicted option accordingly; failures to produce a valid index/text mapping are treated as incorrect.

Probability-based metrics under gen. When probability-sensitive metrics (e.g., Brier score or ECE) are reported under gen, we do *not* necessarily have reliable probability vectors from the model. In such cases we use a simple proxy distribution (e.g., a one-hot vector for the parsed decision, or a uniform distribution as a neutral fallback when parsing fails). Therefore, probability-based metrics under gen should be interpreted as coarse approximations rather than faithful calibration measurements.

F.3.2 Protocol B: lm (LM-probing via next-token probabilities)

In lm, we rewrite the task prompt to require a *single-token* decision, and request token log-probabilities (logprobs/top_logprobs). This removes dependence on long-form generation and focuses on the model’s local preference over the decision space.

Binary validation. We construct a constrained prompt of the form:

Respond with ONLY one token: YES or NO.

Let $\mathcal{Y} = \{\text{YES}, \text{NO}\}$ denote the candidate set. We compute probabilities by normalizing over candidates:

$$P(y) = \frac{\exp(\ell(y))}{\sum_{y' \in \mathcal{Y}} \exp(\ell(y'))}, \quad y \in \mathcal{Y}, \quad (1)$$

where $\ell(y)$ is the logit (or log-prob) associated with candidate token y . We then define the score $s = P(\text{YES})$ and obtain a hard label by thresholding (e.g., $\hat{y} = \text{YES}$ iff $s \geq 0.5$). This protocol enables meaningful computation of probability-sensitive metrics such as Brier score and ECE.

Multiple-choice selection. We construct a constrained prompt requiring a single integer token:

Respond with ONLY the integer index (0..K-1).

Let $\mathcal{I} = \{0, 1, \dots, K-1\}$ denote the candidate indices. We compute a categorical distribution over indices:

$$P(i) = \frac{\exp(\ell(i))}{\sum_{j \in \mathcal{I}} \exp(\ell(j))}, \quad i \in \mathcal{I}, \quad (2)$$

and predict $\hat{i} = \arg \max_i P(i)$. The resulting probability vector supports ranking- and calibration-related metrics (e.g., log loss, MRR, ECE).

Sensitivity and robustness. Because lm uses next-token probabilities, it can be more sensitive to prompt phrasing and tokenization details than gen. However, it avoids long-form formatting errors and provides a direct probability view of the decision space.

F.3.3 Why gen vs. lm can change metrics

The two protocols can yield different discrete decisions on borderline examples due to different decision mechanisms. In gen, the final label is determined by the realized generated string (and parsing), whereas in lm the label is derived from $P(\cdot)$ under an explicit decision distribution. As a result, accuracy/F1 may differ between gen and lm, and probability-based metrics are typically more informative under lm.

F.3.4 Reporting gen/lm and computing overall scores

For the three discriminative tasks, we report both gen and lm results as “gen / lm” in tables. To obtain a single scalar per task, we average the primary metric across the two protocols:

$$m_t = \frac{1}{2} (m_t^{\text{gen}} + m_t^{\text{lm}}), \quad t \in \mathcal{T}_{\text{disc}}. \quad (3)$$

For the remaining tasks (ORDERING, STEP COMPLETION, RATIONALIZATION), the scoring does not depend on the gen/lm protocol (i.e., the evaluation is defined on the structured prediction itself rather than on next-token decision probabilities). Hence we treat the two views as identical:

$$m_t^{\text{gen}} = m_t^{\text{lm}} \triangleq m_t, \quad t \in \mathcal{T} \setminus \mathcal{T}_{\text{disc}}, \quad (4)$$

and averaging over gen/lm leaves the score unchanged. Finally, Primary-Overall macro-averages the six task-level primary metrics with equal weights:

$$\text{Primary-Overall} = \frac{1}{6} \sum_{t \in \mathcal{T}} m_t. \quad (5)$$

F.3.5 Interpretation

We recommend interpreting gen as an end-to-end measure of instruction-following and response formation, and lm as a focused probe of token-level preferences and probability quality on discriminative decisions. Reporting both views helps distinguish models that are strong at probability discrimination but brittle in free-form generation (or vice versa).

Unless otherwise noted, benchmark results are obtained with deterministic decoding (temperature = 0) under a fixed evaluation protocol; reported scores are single-run evaluation results. For controlled adaptation experiments, we report results from a single training run with a fixed random seed (seed=42).

G Complete Results Across All Metrics

This appendix reports the complete evaluation results for all models across every metric used in this work. In addition to the primary metrics discussed in the paper, we include the full set of task-level and auxiliary metrics to enable transparent comparison and reproducibility. Tables are organized consistently with the main text (model groups and task order), and we recommend using the primary metrics for headline comparison while treating secondary metrics as diagnostic signals for specific error modes.

G.1 Models Evaluated

Open-source general-purpose LLMs.

- **Llama-2** (Touvron et al., 2023) is an open-weights general-purpose LLM family; we evaluate its instruction-tuned variant. (locally hosted)
- **Llama-3.1** (Grattafiori et al., 2024) is a newer open-weights LLM family with improved general capabilities across model scales. (locally hosted)
- **Mixtral** (Jiang et al., 2024) is a sparse mixture-of-experts LLM; we use the instruction-tuned variant as a strong open-source baseline. (locally hosted)
- **Phi-3-mini** (Abdin et al., 2024) is a small general-purpose model optimized for strong accuracy at compact scale. (locally hosted)
- **Gemma-2** (Team et al., 2024) is an open-weights model family released by Google; we evaluate the instruction-tuned checkpoint. (locally hosted)
- **Qwen-2.5** (Qwen Team et al., 2025) is an open-weights model family spanning multiple sizes; we evaluate instruction-tuned variants. (locally hosted)
- **DeepSeek-v3.2** (DeepSeek-AI et al., 2025) is a large-scale open model; we include it as a high-capacity open baseline accessed via API.

Proprietary API models.

- **Claude Sonnet 4** (Anthropic, 2025) is a proprietary instruction-following model accessed via API.
- **Gemini-2.5-Flash** (Comanici et al., 2025) is a proprietary Gemini-2.5 series model accessed via API.
- **GPT-4o** (Hurst et al., 2024) is a proprietary multimodal model; we use its text interface for all tasks via API.
- **GPT-5.2** (OpenAI, 2025) is a proprietary model accessed via API.
- **Grok-4** (xAI, 2025) is a proprietary model accessed via API.

- **Qwen-3-max** (Team et al., 2025) is a hosted large Qwen3-series model variant accessed via API.

Domain-specific chemistry models.

- **ChemLLM** (Zhang et al., 2024) is a chemistry-specialized dialogue LLM trained with chemistry instruction tuning and accompanied by chemistry-focused data/benchmarks; we include it as a representative chemistry-native baseline. (locally hosted)
- **LlaSMol** (Yu et al., 2024) is a chemistry-oriented instruction-tuned model built on a large, task-diverse small-molecule instruction dataset, targeting broad chemistry understanding and reasoning. (locally hosted)
- **KALE** (Dai et al., 2026) is a knowledge- and logic-enhanced model series for science, with chemistry-specialized variants; we evaluate it as a domain-adapted chemistry baseline. (locally hosted)
- **ChemDFM** (Zhao et al., 2025b) is a chemistry foundation model trained on chemistry-centric corpora and further instruction-tuned for chemistry problem solving; we evaluate it as a strong domain-specific baseline. (locally hosted)

G.2 Complete Evaluation Results Across Every Metric

This subsection presents the full evaluation results for the six task types in CHEMREASON-BENCH. In addition to the primary metric reported in the main paper, we include all secondary metrics for each task to provide a more complete and transparent comparison across models. Tables 8–13 correspond to ORDERING, STEP VALIDATION, CONDITION VALIDATION, CONTRASTIVE CHOICE, STEP COMPLETION, and RATIONALIZATION, respectively.

Model	Size	Ordering			
		Primary metric \uparrow	PairAcc \uparrow	EM \uparrow	Kendall- τ_{norm} \uparrow
Open-source Models					
Llama-2	7B	52.27	52.27	18.64	52.27
Mixtral-v0.2	7B	64.58	64.58	37.44	64.58
Phi-3-mini	7B	79.83	79.83	52.13	79.83
Qwen-2.5	7B	78.74	78.74	49.92	78.74
Llama-3.1	8B	73.14	73.14	40.60	73.14
Gemma-2	9B	83.92	83.92	57.58	83.92
Llama-3.1	70B	88.55	88.55	71.17	88.55
Qwen-2.5	72B	91.32	91.32	76.78	91.32
DeepSeek-v3.2	685B	91.53	91.53	76.86	91.53
Proprietary Models					
Claude-Sonnet-4	–	94.00	94.00	84.60	94.00
Gemini-2.5-Flash	–	92.46	92.46	82.07	92.46
GPT-4o	–	92.90	92.90	80.92	92.90
GPT-5.2	–	94.40	94.40	85.39	94.40
Grok-4	–	90.02	90.02	74.29	90.02
Qwen-3-max	–	93.78	93.78	83.49	93.78
Domain-Specific Models					
ChemLLM	7B	56.93	56.93	18.40	56.93
LlaSMol	7B	0.50	0.50	0.32	0.50
KALE	8B	76.25	76.25	46.68	76.25
ChemDFM	13B	14.86	14.86	5.29	14.86

Table 8: Detailed results for Ordering on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

Model	Size	Step Validation						
		Primary metric \uparrow	F1 $^{\dagger\uparrow}$	Acc \uparrow	Brier \downarrow	ECE \downarrow	AUROC \uparrow	AUPRC \uparrow
Open-source Models								
Llama-2	7B	59.36	66.84 / 51.88	55.84 / 49.74	23.22 / 26.23	17.25 / 8.30	73.55 / 50.17	67.36 / 44.03
Mixtral-v0.2	7B	50.46	69.43 / 31.50	68.47 / 62.11	22.59 / 36.22	13.38 / 36.76	72.26 / 80.53	68.63 / 78.51
Phi-3-mini	7B	44.72	72.46 / 16.98	70.99 / 58.28	22.03 / 30.68	16.55 / 33.11	76.27 / 82.45	69.39 / 80.50
Qwen-2.5	7B	72.23	75.10 / 69.35	72.91 / 76.83	19.00 / 20.37	12.24 / 19.22	80.92 / 87.05	74.87 / 85.34
Llama-3.1	8B	58.18	70.90 / 45.45	72.82 / 66.55	22.62 / 22.61	16.41 / 22.73	74.61 / 84.68	68.82 / 81.21
Gemma-2	9B	72.42	69.51 / 75.34	76.31 / 79.36	18.93 / 17.95	15.44 / 16.77	81.84 / 88.18	78.61 / 86.44
Llama-3.1	70B	53.36	62.62 / 44.09	73.69 / 66.64	25.99 / 25.76	24.80 / 28.26	72.15 / 88.14	74.14 / 86.34
Qwen-2.5	72B	79.41	80.47 / 78.36	81.01 / 82.06	13.95 / 15.19	7.50 / 13.15	87.64 / 90.72	83.84 / 89.50
DeepSeek-v3.2	685B	70.60	70.60 / -	77.87 / -	16.46 / -	11.65 / -	85.14 / -	84.00 / -
Proprietary Models								
Claude-Sonnet-4	-	76.43	76.43 / -	81.36 / -	14.13 / -	8.60 / -	88.07 / -	86.55 / -
Gemini-2.5-Flash	-	76.45	76.45 / -	80.84 / -	15.89 / -	13.34 / -	87.41 / -	84.58 / -
GPT-4o	-	78.63	82.54 / 74.73	82.75 / 79.97	13.54 / 16.87	7.34 / 15.40	88.06 / 90.15	81.13 / 88.25
GPT-5.2	-	73.17	80.16 / 66.18	82.40 / 75.61	12.94 / 20.64	4.96 / 19.69	89.72 / 89.14	85.21 / 87.77
Grok-4	-	79.81	81.76 / 77.87	83.71 / 81.53	13.55 / 16.83	7.79 / 15.92	87.81 / 88.65	81.99 / 88.00
Qwen-3-max	-	74.78	73.31 / 76.24	79.70 / 81.27	15.16 / 17.65	11.36 / 17.38	88.09 / 90.39	84.69 / 89.38
Domain-Specific Models								
ChemLLM	7B	34.89	28.13 / 41.66	59.06 / 63.15	40.03 / 22.47	39.51 / 7.55	55.94 / 72.46	54.76 / 68.99
LlaSMol	7B	30.89	61.78 / 0.00	45.47 / 54.36	26.07 / 44.70	6.32 / 44.59	49.30 / 49.80	43.21 / 46.21
KALE	8B	60.32	57.78 / 62.87	66.90 / 68.21	31.18 / 20.72	29.27 / 7.16	65.51 / 75.34	58.43 / 68.74
ChemDFM	13B	38.03	62.76 / 13.31	52.44 / 51.22	25.05 / 28.40	6.80 / 15.12	57.45 / 44.64	53.57 / 41.42

Table 9: Detailed results for Step Validation on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

Model	Size	Condition Validation						
		Primary metric \uparrow	F1 † \uparrow	Acc \uparrow	Brier \downarrow	ECE \downarrow	AUROC \uparrow	AUPRC \uparrow
Open-source Models								
Llama-2	7B	74.81	74.93 / 74.70	62.13 / 65.80	22.33 / 21.37	14.28 / 8.25	73.08 / 73.48	74.36 / 75.91
Mixtral-v0.2	7B	52.22	80.55 / 23.88	73.41 / 49.78	20.08 / 47.05	13.53 / 48.41	73.49 / 85.18	78.27 / 85.07
Phi-3-mini	7B	51.86	83.33 / 20.39	77.62 / 48.97	16.26 / 31.38	10.12 / 37.18	85.14 / 88.27	86.68 / 89.96
Qwen-2.5	7B	82.95	80.85 / 85.06	73.23 / 83.17	16.87 / 13.98	14.40 / 12.22	86.58 / 90.01	86.59 / 89.73
Llama-3.1	8B	71.30	80.80 / 61.79	74.22 / 66.79	16.99 / 21.05	6.93 / 23.55	81.32 / 87.51	83.56 / 89.24
Gemma-2	9B	82.29	82.09 / 82.49	80.39 / 80.57	15.64 / 16.30	11.06 / 14.51	85.59 / 88.37	86.95 / 88.77
Llama-3.1	70B	72.99	82.71 / 63.27	81.47 / 68.40	17.64 / 22.08	14.41 / 27.53	83.94 / 91.60	88.69 / 92.24
Qwen-2.5	72B	87.24	88.60 / 85.88	86.03 / 84.78	11.26 / 12.34	11.36 / 11.06	91.91 / 93.04	91.43 / 93.05
DeepSeek-v3.2	685B	79.93	79.93 / -	79.59 / -	15.59 / -	11.69 / -	85.48 / -	88.17 / -
Proprietary Models								
Claude-Sonnet-4	-	80.73	80.73 / -	80.13 / -	15.13 / -	9.47 / -	86.02 / -	88.76 / -
Gemini-2.5-Flash	-	86.89	86.89 / -	85.41 / -	12.67 / -	8.00 / -	89.20 / -	90.05 / -
GPT-4o	-	83.34	87.21 / 79.48	84.24 / 78.96	11.91 / 16.45	10.43 / 16.12	91.30 / 90.75	92.12 / 91.38
GPT-5.2	-	84.40	83.74 / 85.07	81.29 / 84.06	13.88 / 13.31	6.14 / 12.18	87.34 / 92.70	86.91 / 93.16
Grok-4	-	85.78	87.01 / 84.56	85.14 / 83.62	12.02 / 14.83	5.20 / 13.92	89.48 / 91.06	90.30 / 92.37
Qwen-3-max	-	80.14	78.03 / 82.26	78.07 / 81.47	15.34 / 17.51	10.39 / 17.35	86.59 / 91.67	89.16 / 92.11
Domain-Specific Models								
ChemLLM	7B	57.55	45.09 / 70.02	55.95 / 70.64	42.57 / 18.13	40.22 / 3.95	59.88 / 83.98	71.40 / 86.22
LlaSMol	7B	35.04	70.09 / 0.00	54.61 / 42.70	27.86 / 56.30	10.58 / 56.36	47.64 / 31.23	58.96 / 45.05
KALE	8B	71.54	65.11 / 77.97	65.26 / 73.59	35.45 / 19.46	36.17 / 11.66	66.75 / 79.50	71.00 / 81.03
ChemDFM	13B	39.68	73.15 / 6.22	60.70 / 37.87	25.58 / 34.05	11.47 / 30.72	56.96 / 40.25	66.03 / 48.81

Table 10: Detailed results for Condition Validation on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

Model	Size	Contrastive Choice				
		Primary metric↑	Top1Acc↑	LogLoss↓	MRR↑	ECE↓
Open-source Models						
Llama-2	7B	19.41	24.05 / 14.76	2229.14 / 188.64	51.73 / 44.24	65.16 / 41.43
Mixtral-v0.2	7B	54.18	55.71 / 52.65	1493.40 / 207.51	60.74 / 70.78	55.32 / 34.42
Phi-3-mini	7B	55.15	59.24 / 51.07	232.06 / 110.15	58.24 / 70.74	43.77 / 12.37
Qwen-2.5	7B	64.53	63.42 / 65.65	281.81 / 261.19	61.27 / 79.53	51.25 / 29.31
Llama-3.1	8B	61.98	62.58 / 61.37	683.39 / 109.15	57.14 / 76.84	56.36 / 8.45
Gemma-2	9B	69.13	67.50 / 70.75	2137.46 / 128.73	53.60 / 83.05	73.77 / 21.34
Llama-3.1	70B	79.25	77.25 / 81.24	463.37 / 56.59	83.64 / 89.15	16.08 / 6.59
Qwen-2.5	72B	78.88	77.25 / 80.50	363.51 / 118.37	83.11 / 88.94	17.78 / 15.92
DeepSeek-v3.2	685B	81.52	81.52 / -	97.74 / -	88.50 / -	4.26 / -
Proprietary Models						
Claude-Sonnet-4	-	73.82	73.82 / -	526.00 / -	83.80 / -	9.76 / -
Gemini-2.5-Flash	-	82.27	82.27 / -	145.37 / -	88.62 / -	4.51 / -
GPT-4o	-	84.91	83.94 / 85.89	79.00 / 74.24	88.97 / 91.98	8.76 / 10.37
GPT-5.2	-	88.72	88.67 / 88.77	45.45 / 63.85	92.88 / 93.75	2.10 / 9.09
Grok-4	-	78.27	78.09 / 78.46	140.56 / 145.92	85.73 / 87.60	8.91 / 17.48
Qwen-3-max	-	80.78	79.20 / 82.36	128.75 / 144.60	87.19 / 89.94	6.77 / 16.11
Domain-Specific Models						
ChemLLM	7B	48.42	44.38 / 52.46	2495.64 / 112.44	51.97 / 71.88	70.62 / 12.86
LlaSMol	7B	18.48	1.58 / 35.38	3387.83 / 139.48	48.45 / 58.31	17.24 / 8.52
KALE	8B	63.37	60.17 / 66.57	1354.96 / 90.20	74.57 / 80.33	39.53 / 15.17
ChemDFM	13B	32.64	25.72 / 39.55	2565.55 / 129.60	58.39 / 63.16	30.73 / 2.47

Table 11: Detailed results for Contrastive Choice on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

Model	Size	Step Completion			
		Primary metric↑	Action-EM↑	Slot-F1↑	FormatErr↓
Open-source Models					
Llama-2	7B	6.04	6.04	5.39	8.69
Mixtral-v0.2	7B	4.68	4.68	5.19	2.61
Phi-3-mini	7B	5.30	5.30	5.39	5.33
Qwen-2.5	7B	7.25	7.25	8.16	5.15
Llama-3.1	8B	9.22	9.22	11.19	2.05
Gemma-2	9B	11.54	11.54	12.81	6.82
Llama-3.1	70B	16.06	16.06	19.02	6.00
Qwen-2.5	72B	25.73	25.73	29.67	12.10
DeepSeek-v3.2	685B	36.32	36.32	43.02	10.11
Proprietary Models					
Claude-Sonnet-4	–	43.63	43.63	51.18	13.43
Gemini-2.5-Flash	–	35.74	35.74	41.74	11.86
GPT-4o	–	23.72	23.72	27.98	6.70
GPT-5.2	–	51.65	51.65	60.25	17.25
Grok-4	–	35.48	35.48	40.56	15.39
Qwen-3-max	–	30.54	30.54	35.17	12.26
Domain-Specific Models					
ChemLLM	7B	4.19	4.19	4.92	1.35
LlaSMol	7B	0.00	0.00	0.00	0.00
KALE	8B	5.26	5.26	6.54	3.10
ChemDFM	13B	0.60	0.60	0.54	0.86

Table 12: Detailed results for Step Completion on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

Model	Size	Rationalization				
		Primary metric↑	Cov-F1↑	ROUGE-L F1↑	BLEU↑	BERTScore-F1↑
Open-source Models						
Llama-2	7B	19.57	19.57	20.71	2.72	1.80
Mixtral-v0.2	7B	20.28	20.28	20.23	2.30	15.92
Phi-3-mini	7B	24.19	24.19	23.28	3.08	24.09
Qwen-2.5	7B	17.96	17.96	18.48	1.03	20.26
Llama-3.1	8B	22.87	22.87	23.21	3.29	22.48
Gemma-2	9B	21.15	21.15	20.82	1.06	23.38
Llama-3.1	70B	26.21	26.21	24.66	4.42	25.90
Qwen-2.5	72B	25.31	25.31	24.40	3.02	26.50
DeepSeek-v3.2	685B	31.33	31.33	27.20	5.41	29.80
Proprietary Models						
Claude-Sonnet-4	–	30.74	30.74	27.05	4.95	29.56
Gemini-2.5-Flash	–	28.83	28.83	25.88	4.97	27.72
GPT-4o	–	27.87	27.87	26.18	4.41	28.10
GPT-5.2	–	29.45	29.45	25.19	3.30	27.37
Grok-4	–	30.05	30.05	25.19	5.90	26.73
Qwen-3-max	–	30.46	30.46	26.17	5.24	29.20
Domain-Specific Models						
ChemLLM	7B	15.84	15.84	17.72	1.96	0
LlaSMol	7B	1.31	1.31	2.12	0.16	0
KALE	8B	18.52	18.52	20.85	2.04	0
ChemDFM	13B	7.88	7.88	8.18	1.04	0

Table 13: Detailed results for Rationalization on CHEMREASON-BENCH. Primary metric is the normalized task composite used in Table 2; the remaining columns report the primary and secondary metrics (Section 5).

H Fine-Tuning Details: CHEMREASON-TUNE, SFT, and GRPO

This appendix documents the fine-tuning data construction, splits, and training objectives used to adapt LLMs using CHEMREASON’s verifiable task interface.

H.1 Fine-Tuning Data Construction and Splits

Reaction pools and instance scale. We construct two reaction pools (Fig. 3, Sector 1): (i) a 500-reaction benchmark pool for evaluation, and (ii) a $> 20\text{k}$ -reaction training pool for adaptation. From the training pool, we instantiate $> 120\text{k}$ verifiable task instances (denoted CHEMREASON-TUNE) by reusing the same expert-designed template library and unified schemas that define evaluation (Fig. 3, Sectors 2–3).

Template-based instantiation with unified schemas. Each task type is defined by a template that specifies: (i) the evidence shown to the model (context, candidates/options, and constraints such as schema and legend), (ii) the required output type, and (iii) the evaluation/reward protocol. Instances are serialized into structured targets (e.g., JSON fields for label/index/order, or action+slots for step completion), enabling deterministic parsing and validator-based checking.

Controlled re-instantiation. To improve robustness and reduce overfitting to a single prompt surface, we allow controlled surface-level re-instantiation for training: prompt contexts may be paraphrased or lightly diversified, while preserving the underlying constraints (schema validity, legend/placeholder bindings, and gold labels). All generated variants are passed through the same rule-based validators; invalid or constraint-violating variants are filtered out before training.

H.2 Training Details

Hardware. All fine-tuning experiments were run on $4 \times$ NVIDIA A800 GPUs with mixed-precision training (bf16).

Model and parameterization. We fine-tune instruction-tuned backbones using LoRA adapters (rank $r=16$, $\alpha=32$, dropout = 0.05) applied to the attention and MLP projection modules (q/k/v/o and gate/up/down projections). We use bfloat16 parameters and standard SDPA attention.

Data. We train on CHEMREASON-TUNE JSONL splits for SFT and GRPO. RL/GRPO uses task instances paired with ground truth so that rewards can be computed by deterministic parsers/validators.

Supervised fine-tuning (SFT). SFT is run for 4 epochs with AdamW, learning rate 5×10^{-6} , weight decay 0.01, warmup ratio 0.1, and gradient clipping at 1.0. We use per-device batch size 4 with gradient accumulation 4 (effective batch size 16 per step) and a maximum sequence length of 2048. We evaluate and save checkpoints every 500 steps and log every 50 steps.

GRPO stage. GRPO is run for 2000 optimization steps with learning rate 5×10^{-7} , per-device batch size 4. For each prompt, we sample 4 candidate generations and compute verifiable rewards using deterministic parsers and validators aligned with each task type (Appendix Table 14). Prompts are truncated to 2048 tokens and generations are capped at 512 tokens. We use temperature 0.7 with nucleus sampling ($p=0.9$) and top- $k=50$, KL penalty coefficient $\beta=0.1$, and PPO-style clipping with clip range 0.2. We evaluate every 100 steps, save every 50 steps, and log every 10 steps. All runs use a fixed random seed (42).

Splitting and leakage control. We split data at the reaction-record level (not at the instance level): all task instances derived from the same reaction record are assigned to the same split. This prevents near-duplicate leakage across splits and ensures that generalization is measured over reactions rather than reaction-specific artifacts. Unless otherwise specified, evaluation results in the main paper are reported on the 500-reaction benchmark pool only; the larger pool is used exclusively for training and auxiliary analyses.

H.3 Supervised Fine-Tuning (SFT)

Objective. Let (x_i, y_i) be the prompt and the reference structured output (serialized as a token sequence) for instance i . We apply standard teacher-forcing with a causal language modeling objective and compute the loss only over the target region:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^{|y_i|} \log p_{\theta}(y_{i,t} \mid x_i, y_{i,<t}). \quad (6)$$

SFT primarily teaches (i) instruction following for each task format and (ii) schema-valid, parseable structured outputs under the unified interface.

H.4 GRPO Fine-Tuning with Verifiable Rewards

Verifiable rewards. For each task type, we define a scalar reward $r(x, y) \in [0, 1]$ computed by deterministic parsing and validation of the model output, followed by task-specific scoring. Invalid or schema-violating outputs receive zero reward by construction. A consolidated specification of reward functions and edge cases is provided in Appendix Table 14.

Group sampling and advantage normalization. Given a prompt x , we sample a group of K candidate completions $\{y^{(k)}\}_{k=1}^K$ from the current policy. We compute rewards $\{r^{(k)}\}$ and form normalized within-group advantages:

$$A^{(k)} = \frac{r(x, y^{(k)}) - \mu_r}{\sigma_r + \epsilon}, \quad (7)$$

$$\mu_r = \frac{1}{K} \sum_{k=1}^K r(x, y^{(k)}), \quad (8)$$

$$\sigma_r^2 = \frac{1}{K} \sum_{k=1}^K (r(x, y^{(k)}) - \mu_r)^2. \quad (9)$$

Optimization objective with regularization. We optimize a reward-weighted objective with a KL regularizer to prevent excessive drift from a reference policy (typically the SFT checkpoint):

$$\max_{\theta} \mathbb{E}_x \left[\frac{1}{K} \sum_{k=1}^K A^{(k)} \log p_{\theta}(y^{(k)} | x) - \beta \text{KL}(p_{\theta}(\cdot | x) \| p_{\text{ref}}(\cdot | x)) \right]. \quad (10)$$

Because rewards are computed via strict parsers/validators, GRPO directly reinforces constraint-satisfying procedural behavior: correct discrete decisions for discriminative tasks, accurate structured reconstruction for step completion, and faithful placeholder/legend bindings.

Consistency between training and evaluation. We maintain a consistent interface across the pipeline: the same schemas, parsers, canonicalization rules, and validation logic used for evaluation are also used to compute rewards during GRPO. This alignment ensures that improvements from fine-tuning translate to measurable gains under the benchmark’s scoring protocol.

H.5 GRPO Reward Functions

This subsection presents the GRPO-aligned reward functions for all task types in CHEMREASON-BENCH. As summarized in Table 14, we specify for each task the parsed prediction schema, the corresponding reward definition, and important implementation details such as normalization, canonicalization, and edge-case handling. Overall, the reward design follows the same principle as benchmark evaluation: whenever possible, rewards are grounded in deterministic parsing and validator-based checking, so that policy optimization remains aligned with the benchmark’s verifiable output interface.

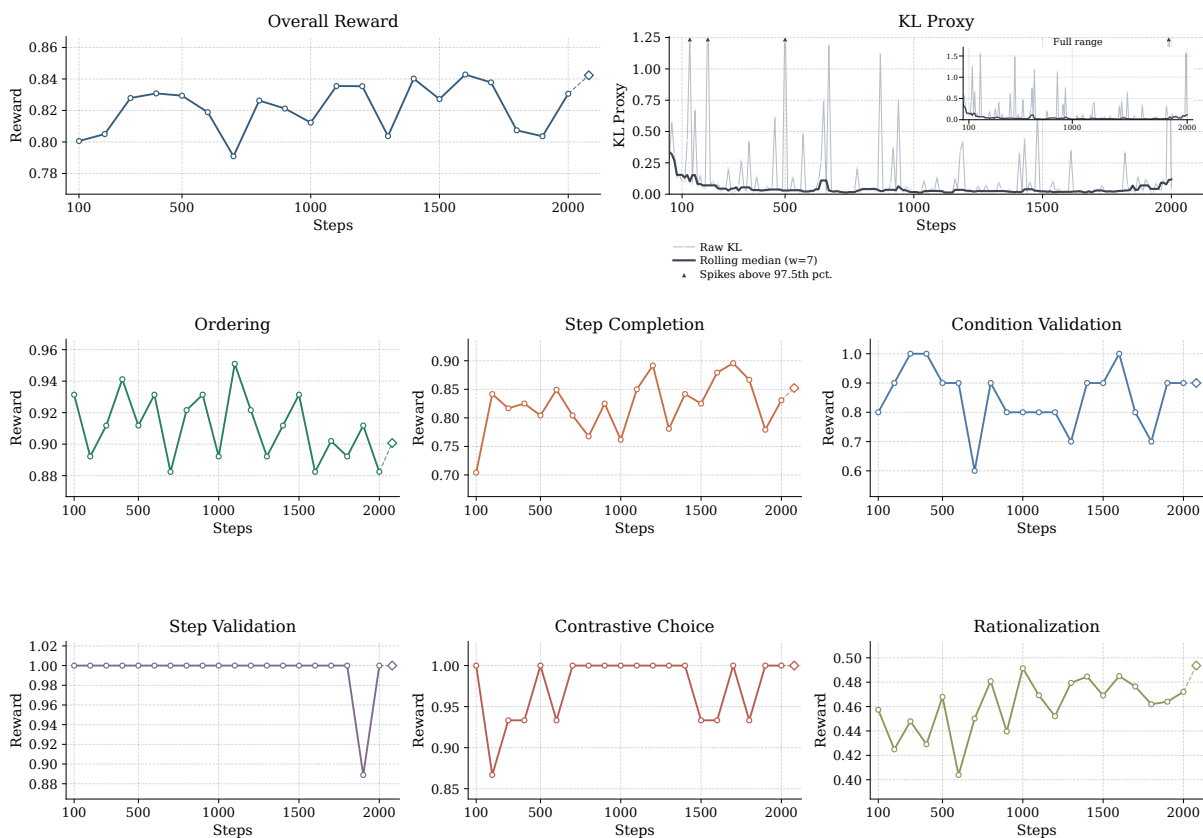


Figure 11: GRPO training dynamics on CHEMREASON-TUNE. The figure shows the overall evaluation reward, per-task rewards for the six subtasks, and a KL proxy over training steps. Reward is computed using task-specific automatic validators and normalized to $[0, 1]$. For the KL proxy, we plot both the raw KL values and a rolling median (window size $w = 7$); extreme spikes above the 97.5th percentile are marked, and the inset shows the full raw range. Overall reward improves modestly and remains relatively stable throughout training, with most gains coming from STEP COMPLETION, CONDITION VALIDATION, and RATIONALIZATION, while STEP VALIDATION and CONTRASTIVE CHOICE remain near saturation.

Table 14: ChemReason-Bench (GRPO-aligned) reward functions by task.

Task	Prediction schema (after parsing)	Reward definition and key details / edge cases
		Pairwise accuracy:
		$r = \frac{\#\{(i, j) : i < j, \pi_{\text{pred}}(s_i) < \pi_{\text{pred}}(s_j)\}}{\#\{(i, j) : i < j\}}$
ordering	{predicted_order: [str]}	where (s_i) is the gold correct_order. Cleaning: filter predicted ids to those in gold; if at least one valid id was predicted, append any missing gold ids in gold order. Degenerate: if gold length < 2, reward = 1.0; if prediction is empty / has no valid ids, reward = 0.0.
		Hard reward (exact match):
		$r = \mathbb{1}[\hat{k} = k^*]$
contrastive_choice	{predicted_option_idx: int} (optionally probs, predicted_choice)	(0/1 accuracy on option index). Note: current implementation is exact-match; comments indicate a future version may restore a softer reward using calibrated probabilities.
		Hard reward (label match):
		$r = \mathbb{1}[\hat{y} = y^*]$
step_validation	{label: bool} (score optional; ignored by reward)	(0/1 correctness). Reward does not use calibrated score; only label correctness matters. Missing label \Rightarrow 0.
		Hard reward (label match):
		$r = \mathbb{1}[\hat{y} = y^*]$
condition_validation	{label: bool} (score optional; ignored by reward)	(0/1 correctness). Same reward function as step_validation. Missing label \Rightarrow 0.
		Structured reconstruction:
		$r = 0.8 \cdot \text{EM}(\text{action}) + 0.2 \cdot F1_{\text{slots}}$
step_completion	{action: str, slots: dict}	with fatal error \Rightarrow 0. Protection 1: if gold action is non-empty but predicted action is empty or not in the allowed action space $\Rightarrow r = 0$. Slots canonicalization: merge aliases to reagent; map names to placeholders via legend; parse amounts into amount_value/unit; detect temperature/duration tokens (## / @ @). Slot F1: token fields by exact match; reagent placeholders via (multi)set overlap; numeric fields via tolerance; units are normalized and illegal units trigger a fatal flag $\Rightarrow r = 0$. Protection 2: if both gold and pred slots are empty, then $F1_{\text{slots}} = 1$ iff action EM is 1, else 0.
		Token-based soft reward:
		$r = 0.5 \cdot \text{coverage} + 0.5 \cdot \sqrt{F1}$
rationalization	{gold_rationale: str}	then apply a length prior and clamp to $[0, 1]$. Tokens are normalized (lowercase, strip punctuation) and stopwords are removed. Coverage: fraction of unique gold tokens appearing in prediction. Smoothing: use $\sqrt{F1}$ to avoid tiny F1 collapsing gradients. Length prior: if predicted tokens < 5, reward $\times 0.5$; if > 150, reward $\times 0.9$; empty output \Rightarrow 0.

H.6 Finetuning Results

Model	Size	Overall	Order	Contrast	Step-Val	Cond-Val	Step-Comp	Ration.
		↑	PairAcc.↑	Top1Acc.↑ (lm)	F1+↑ (lm)	F1+↑ (lm)	SC-Score↑	CovF1↑
Qwen-2.5	72B	64.65	91.32	80.50	78.36	85.88	25.73	25.31
GPT-4o	-	65.23	92.90	85.89	74.73	79.48	23.72	27.87
GPT-5.2	-	70.30	94.40	88.77	66.18	85.07	51.65	29.45
Qwen-2.5	7B	53.94	78.74	65.65	69.35	85.06	7.25	17.96
+SFT	7B	65.35	92.04	71.31	66.27	89.51	49.16	23.81
+SFT+GRPO	7B	65.25	91.69	71.40	64.96	89.60	49.27	24.58
Llama-3.1	8B	49.45	73.14	61.37	45.45	61.79	9.23	22.87
+SFT	8B	60.02	92.00	62.02	52.39	75.49	52.91	25.34
+SFT+GRPO	8B	60.70	92.11	62.58	53.81	76.91	53.54	25.24
Gemma-2	9B	56.74	83.92	70.75	75.34	82.49	11.54	21.15
+SFT	9B	67.55	93.00	71.77	74.51	86.96	53.36	25.71
+SFT+GRPO	9B	67.44	92.74	71.87	73.80	86.77	54.08	25.41

Table 15: Comparison of base vs. fine-tuned variants on three representative open-source models, together with the best-performing baseline model for each task. For discriminative tasks, scores are reported in lm mode.