CHEMDEDUCE: BENCHMARKING LLMS FOR UNKNOWN CHEMICAL IDENTIFICATION THROUGH INTERACTIVE SIMULATED EXPERIMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown strong performance in chemical knowledge and reasoning, yet it remains unclear whether they can solve problems whose answers must be discovered through experimentation. We investigate this question through Unknown Chemical Identification, where a model must actively design experiments, interpret observations, update hypotheses, and ultimately identify a hidden substance. We introduce ChemDeduce, an interactive evaluation framework for assessing LLMs through simulated experiments. ChemDeduce formulates chemical identification as a sequential decision process and provides a stateful simulated laboratory with explicit instrument and operational constraints, enabling scalable and reproducible evaluation of iterative experimental reasoning. We benchmark eight frontier LLMs on 200 chemicals, yielding $1{,}600$ model–target pairs and $4{,}800$ episodes over three independent repetitions, and evaluate identification accuracy, operational compliance, interaction efficiency, and trajectory-level reasoning quality. Despite generating seemingly plausible experimental plans, current models achieve an average identification accuracy ranging from 23.83% to 67.33%, revealing substantial limitations in experiment selection, evidence interpretation, and sequential hypothesis refinement.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities in chemical knowledge, molecular reasoning, synthesis planning, and chemistry-oriented tool use (Mirza et al., 2025; Huang et al., 2026; M. Bran et al., 2024; Yu et al., 2024). Yet many laboratory problems cannot be solved by recalling or reasoning over existing knowledge alone; their answers must instead be discovered through real-world experimentation. Unknown chemical identification provides a natural example. Given an unlabeled sample and a set of available reagents and instruments, an experimenter must select informative experiments, execute them under valid conditions, interpret the resulting observations, and progressively refine competing hypotheses based on accumulated evidence. This raises a fundamental question: *can an LLM identify an unknown chemical by autonomously designing and conducting experiments?* Addressing this question requires an LLM to iteratively plan experiments, take actions, and interpret observations within a closed-loop process.

Existing chemistry benchmarks provide limited insight into this capability. Benchmarks such as ChemLLMBench, ChemBench, and ChemEval primarily evaluate static chemical knowledge, offline reasoning, or predefined tool use (Guo et al., 2023; Mirza et al., 2025; Huang et al., 2026). In such settings, the information required for reasoning is largely available beforehand. Unknown chemical identification is fundamentally different: the model must actively acquire missing evidence through a sequence of experiments and use that evidence to progressively reduce uncertainty. Evaluating this process requires tracking the whole interactive experimental trajectory.

Such evaluation requires an environment that can support faithful and reproducible experimental interactions. The environment must respond to each intermediate operation with chemically plausible and state-consistent observations, so that later decisions depend on earlier actions. Real wet-lab experiments offer high physical fidelity but are costly and difficult to scale across many substances and models. Conversely, relying on an unconstrained LLM-only procedure to generate outcomes

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

risks hallucinated phenomena and evaluator-dependent bias. A scalable benchmark therefore needs a constrained simulated environment that preserves the sequential structure of experimentation while controlling states, valid operations, and observable outcomes.

To this end, we introduce **ChemDeduce**, an interactive evaluation framework for benchmarking LLMs on unknown chemical identification through simulated experiments. We formulate the task as a sequential decision process. In each episode, a target chemical is sampled from a pool of 200 chemical substances and its identity is hidden from the model. Acting as an autonomous experimenter, the model proposes experimental operations in natural language, receives the resulting observations or instrument measurements, and iteratively gathers and interprets evidence before making a final identification under a limited interaction budget.

At the core of ChemDeduce is **LabAgent**, a stateful laboratory simulator that integrates virtual instruments, reaction simlution, sample-state tracking, and operational constraints. LabAgent translates natural-language experimental instructions into executable actions and generates observations according to the current experimental state. It also leverages curated chemical knowledge bases and a domain-specific skill library to support reaction and instrument simulation. The stateful design enables reproducible multi-step experimentation while preserving causal consistency across the entire interaction trajectory.

Our contributions are threefold:

- We introduce an interactive benchmark for unknown chemical identification. Unlike existing static chemistry benchmarks, ChemDeduce evaluates whether LLMs can actively design experiments, acquire informative evidence, and iteratively refine hypotheses, providing a controlled testbed for closed-loop experimental reasoning.
- We develop a stateful simulated laboratory for scalable and reproducible evaluation. The environment integrates virtual instruments, reaction simulation, sample-state tracking, and operational constraints, enabling natural-language experimental actions to produce consistent and state-dependent observations.
- We conduct a systematic evaluation of frontier LLMs as autonomous experimental agents. Across $1{,}600$ model–target pairs and $4{,}800$ episodes, our evaluation reveals substantial limitations in current models' ability to select informative experiments, interpret experimental evidence, and consistently update hypotheses over multiple rounds.

2 Related Work

Static Evaluation in Chemistry. A growing body of work has evaluated the scientific and chemical capabilities of LLMs through standardized benchmarks (Hendrycks et al., 2020; Rein et al., 2023; Lu et al., 2022; Wang et al., 2023). General-purpose benchmarks such as MMLU, GPQA, ScienceQA, and SciBench include chemistry questions at undergraduate and graduate levels, while ChemLLMBench (Guo et al., 2023), ChemBench (Mirza et al., 2025), ChemEval (Huang et al., 2026), ChemLLM (Zhang et al., 2024), and FGBench (Liu et al., 2026) more specifically assess chemical knowledge, textual and molecular reasoning, safety awareness, and functional-group understanding. Related work has also explored tool-augmented chemistry agents and LLM-based scientific assistants for chemical reasoning, synthesis planning, and database interaction (Boiko et al., 2023; Yu et al., 2024; M. Bran et al., 2024). These benchmarks and systems provide standardized and reproducible comparisons across models, but their evaluation protocols are predominantly static or task-specific. As a result, they capture important chemical competencies but provide limited insight into whether a model can recognize information gaps, design informative experiments, and acquire the evidence needed to resolve them.

Interactive Evaluation. Interactive benchmarks have increasingly been used to evaluate models in settings where information must be acquired through sequential interaction, tool use, and environment feedback (Yao et al., 2022; Schick et al., 2023; Liu et al., 2024; Zhou et al., 2024; Wang et al., 2022; Jansen et al., 2024). AgentClinic (Schmidgall et al., 2024), for example, evaluates agents in simulated clinical settings where they interact with patients, request examinations, and progressively gather evidence for diagnosis. More general agent benchmarks, including ToolSandbox (Lu et al., 2025), AppWorld (Trivedi et al., 2024), AgentBench (Liu et al., 2024), WebArena (Zhou et al.,

2024), and OSWorld (Xie et al., 2024), evaluate multi-step decision making through interactions with constrained, stateful environments. ScienceWorld (Wang et al., 2022) and DiscoveryWorld (Jansen et al., 2024) move closer to scientific inquiry by requiring agents to perform actions, collect observations, and solve science-oriented tasks in virtual environments. Self-driving laboratory and robotic chemistry studies further demonstrate the potential of closed-loop experimentation for scientific discovery and chemical optimization (Burger et al., 2020; MacLeod et al., 2020; Häse et al., 2018). However, these benchmarks and systems are not designed to systematically evaluate experimental reasoning for chemical identification. To our knowledge, ChemDeduce is the first interactive benchmark specifically designed to evaluate whether LLMs can identify unknown chemicals by sequentially designing simulated experiments, interpreting state-dependent observations, and refining hypotheses over multiple rounds of interaction.

3 METHODS

In this section, we introduce ChemDeduce, an interactive evaluation framework for unknown chemical identification, organized into four main parts: (1) a chemical identification dataset, (2) an interactive evaluation workflow, (3) the LabAgent architecture, and (4) evaluation metrics, as illustrated in Figure 1.

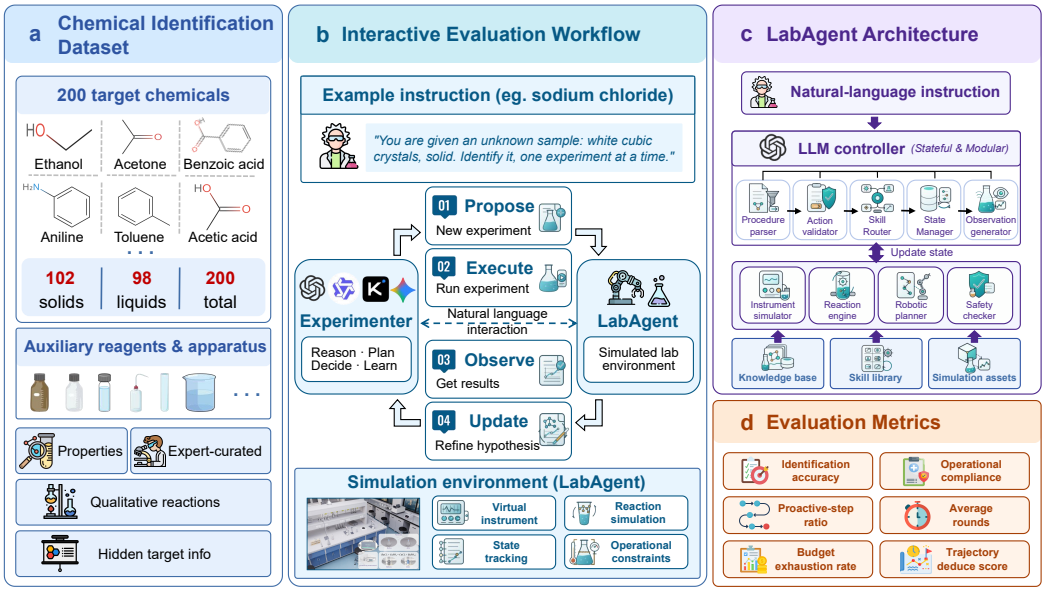


Figure 1: Overview of ChemDeduce. (a) The Chemical Identification Dataset contains 200 target chemicals with curated experimental knowledge and virtual-laboratory resources. (b) The Interactive Evaluation Workflow captures iterative experimentation, observation, and hypothesis refinement. (c) LabAgent integrates an LLM Controller, Chemistry Skills, a Knowledge Base, a Robotic Manipulation Module, and Simulation Assets to execute experiments and generate state-consistent feedback. (d) Evaluation Metrics cover identification accuracy, laboratory-operation validity, successful observation acquisition, interaction efficiency, budget exhaustion, and trajectory-level experimental reasoning quality.

3.1 CHEMICAL IDENTIFICATION DATASET

Data Collection. To support realistic and controlled unknown chemical identification experiments, we collect candidate chemicals, laboratory reagents and associated experimental knowledge from *Analytical Chemistry* (Christian, 1977), *The Systematic Identification of Organic Compounds* (Shriner et al., 2003), *Vogel's Textbook of Practical Organic Chemistry* (Furniss, 2004), and the *CRC Handbook of Chemistry and Physics* (Haynes, 2016). The collected information mainly covers basic physicochemical properties, experimental characteristics, and representative qualitative

reactions, together with the reagents and apparatus required for virtual experiments. These data provide the basis for LabAgent to update laboratory states and generate observable outcomes.

Dataset Generation. We normalize and structure the collected information by standardizing key fields and checking data completeness and consistency of experimental outcomes, and organize the resulting information into target-conditioned records for each chemical. We further retain only substances for which at least one viable identification route exists under the constraints of the simulated laboratory. We then invite three experts with chemistry backgrounds to independently review the samples according to three criteria: Chemical Correctness, Experimental Plausibility, and Record Consistency. Only chemicals approved by the majority vote are retained.

Final Dataset. Following the generation and verification process, we construct a dataset of 200 real chemicals, including 102 solids and 98 liquids, together with validated reagents, apparatus, and experimental records. The dataset covers diverse inorganic and organic chemicals, including several closely related substances that support identification tasks of varying difficulty. During evaluation, the candidate library and target-conditioned records remain hidden from the model under evaluation.

3.2 Interactive Evaluation Workflow

Inspired by the structure of real-world chemical identification workflows, ChemDeduce models unknown chemical identification as a continuous interactive process that progressively acquires evidence through state-dependent experiments. This design moves beyond static chemistry evaluation by modeling the closed loop of experiments, observations, and hypothesis updates to assess iterative experimental reasoning in LLMs.

Agent Roles. Real-world chemical identification involves several participants with clearly delineated responsibilities. To reflect this separation, ChemDeduce defines two core roles in the interactive environment:

$$\mathcal{A} = \{A_{\mathrm{Exp}}, A_{\mathrm{Lab}}\}. \tag{1}$$

The Experimenter, an LLM denoted A_{Exp}, represents the model under evaluation. It is responsible for all deliberative decisions and must identify the hidden target m^* through iterative experimentation and reasoning over experimental observations. In contrast, LabAgent A_{Lab} functions as a responsive experimental environment that interprets, validates, and executes experimental requests, returning observable outcomes according to the current laboratory state. Together, these roles support the evaluation of iterative experimental reasoning.

Interaction Dynamics. Unknown chemical identification is modeled as a temporally ordered multi-turn experimental interaction:

$$H_t = (o_0, u_1, o_1, \ldots, u_t, o_t), \qquad t = 1, 2, \ldots, T, \quad T \le T_{\max}. \tag{2}$$

Here, H_t denotes all information up to round t. Each episode is initialized with a hidden target m^*, and the Experimenter receives its physical state and macroscopic appearance as the initial observation o_0. At each subsequent round, the Experimenter issues a turn-level output u_t based on the history H_{t-1} and receives a new observation o_t from LabAgent A_{Lab}. This recursive process grounds subsequent decisions in accumulated experimental evidence, supporting iterative hypothesis refinement and unknown chemical identification.

Agent Actions. To characterize active experimental decision-making, the Experimenter selects an action from a discrete action space based on the interaction history:

$$\mathcal{A}_{\mathrm{Exp}} = \{\mathrm{Test}, \mathrm{Identify}\}. \tag{3}$$

At round t, the Experimenter produces an action and its associated content conditioned on the interaction history:

$$(a_t^{\mathrm{Exp}}, u_t) = A_{\mathrm{Exp}}(H_{t-1}), \qquad a_t^{\mathrm{Exp}} \in \mathcal{A}_{\mathrm{Exp}}. \tag{4}$$

The action *Test* yields $u_t = q_t$, corresponding to a natural-language experimental request for acquiring new observations. *Identify* produces $u_t = \hat{m}$, representing the final chemical identity and terminating the episode. This abstraction effectively captures the core decision structure of chemical identification. The LabAgent A_{Lab} instead acts as the environment response mechanism, returning feedback according to the Experimenter's action a_t^{Exp} and the current laboratory state s_t.

LabAgent Response. Once the Experimenter issues a *Test* request q_t, LabAgent A_{Lab} checks its executability under the current laboratory state s_t and experimental rules R:

$$c_t = \text{Check}(q_t, s_t, R) \in \{\text{Valid}, \text{Invalid}\}, \tag{5}$$

where R specifies the available resources, safety constraints, and operational requirements. It then produces the observation o_t and, for valid test requests, updates the laboratory state.:

$$o_t = \begin{cases} A_{\text{Lab}}(q_t \mid s_t, m^*), & \text{if } a_t^{\text{Exp}} = \text{Test} \wedge u_t = q_t \wedge c_t = \text{Valid}, \\ \text{Reject}, & \text{if } a_t^{\text{Exp}} = \text{Test} \wedge u_t = q_t \wedge c_t = \text{Invalid}, \\ \text{End}, & \text{if } a_t^{\text{Exp}} = \text{Identify} \wedge u_t = \hat{m}. \end{cases} \tag{6}$$

A valid *Test* updates the laboratory state and returns an observation, whereas an invalid request is rejected and leaves the state unchanged. Each response o_t is constrained by the hidden target m^* and current laboratory state s_t, ensuring state-consistent experimental feedback. This design grounds subsequent Experimenter reasoning in reliable experimental evidence. For valid test requests, LabAgent also updates the laboratory state from s_t to s_{t+1} according to the executed operation; invalid requests return a rejection message and keep $s_{t+1} = s_t$.

History Update. After each interaction, the history is updated recursively as

$$H_t = H_{t-1} \oplus (u_t, o_t). \tag{7}$$

This operation appends the Experimenter's output u_t and the corresponding LabAgent feedback o_t to the interaction history. The updated H_t is then used for subsequent decisions, enabling continuous hypothesis refinement from accumulated experimental results. The interaction terminates when $a_t^{\text{Exp}} = \text{Identify}$ or the interaction budget $T_{\max}$ is reached. The resulting trajectory is used for subsequent outcome evaluation and process analysis.

3.3 LabAgent Architecture

Unknown chemical identification requires state-dependent experimental interactions that provide reliable and state-consistent feedback. To this end, we construct LabAgent as the stateful experimental environment in ChemDeduce. Given an experimental request q_t and the current laboratory state s_t, LabAgent transforms the request into an executable experimental process and returns feedback o_t consistent with the hidden target m^* and the current state. To realize this process, LabAgent integrates five coordinated components: an LLM Controller, Chemistry Skills, a Knowledge Base, a Robotic Manipulation Module, and Simulation Assets.

LLM Controller. The LLM Controller interprets high-level experimental intent and coordinates LabAgent modules. In LabAgent, it parses the Experimenter's natural-language request into a structured experimental procedure specifying experimental objects, key parameters, and operation order, while organizing subsequent validation, tool invocation, and execution according to the current laboratory state. As the system's orchestration core, the LLM Controller coordinates modules under a shared laboratory state, ensuring consistency and coherence across multi-step experiments.

Chemistry Skills. Chemistry Skills encapsulate reusable chemical operations and domain-specific tools. During experiment execution, these skills invoke appropriate tools based on the structured procedure and current laboratory state, combining PubChem data with RDKit-based computation to determine state changes and observable outcomes. This grounds experimental feedback in explicit tool use and chemical computation, ensuring computability and reproducibility.

Knowledge Base. Knowledge Base supplements empirical knowledge that is not fully covered by general-purpose chemistry tools. It stores expert-validated substance properties and qualitative reaction records, which are retrieved when needed according to the hidden target, experimental conditions, and current laboratory state to support reliable outcome generation. To prevent information leakage, all target-specific records remain inaccessible to the Experimenter throughout evaluation.

Robotic Manipulation Module. The Robotic Manipulation Module translates high-level experimental procedures into concrete, executable laboratory operations. In LabAgent, it decomposes validated procedures into basic operations such as sampling, mixing, and measurement, while checking their execution preconditions against the available equipment and current laboratory state.

Simulation Assets. Simulation Assets instantiate the virtual laboratory environment and its observable phenomena. Our simulation environment builds on LabUtopia (Li et al., 2025), which provides the underlying virtual-laboratory framework. They provide experimental scenes, devices, instruments, materials, and corresponding chemical effects, and combine the robotic action sequence with outcomes generated by Chemistry Skills to construct the experimental process and visual feedback. In this way, abstract laboratory state changes are translated into directly observable and recordable phenomena, providing the Experimenter with consistent experimental feedback. More implementation details and task-specific extensions are provided in Appendix D.

The above components jointly form LabAgent's stateful experimental execution pipeline, transforming the Experimenter's natural-language request into a validated experimental procedure, chemically grounded outcome, executable operation, and final observable feedback. Moreover, LabAgent continuously updates the laboratory state and records key execution information, ensuring experimental consistency and supporting subsequent fine-grained analysis.

3.4 Evaluation Metrics

To comprehensively evaluate LLMs for unknown chemical identification, we design a multi-faceted evaluation protocol inspired by real laboratory identification workflows. Beyond final identification accuracy, the protocol further evaluates experimental execution, interaction efficiency, and trajectory-level reasoning quality, providing a comprehensive assessment of both identification outcomes and the underlying experimental process. Specifically, the evaluation metrics are described as follows.

1. **Identification Accuracy:** The accuracy of the final identification, measuring whether the experimenter correctly identifies the hidden target. It provides a primary indicator of the model's identification performance.
2. **Operational Compliance:** The rate at which experimental operations satisfy the required laboratory constraints, reflecting adherence to valid experimental procedures.
3. **Proactive-Step Ratio:** The fraction of attempted operations that are successfully executed and return valid observations, capturing how effectively the model converts experimental requests into usable feedback.
4. **Mean Interaction Rounds:** The number of interaction rounds until episode termination, indicating the efficiency of evidence gathering and decision completion.
5. **Budget Exhaustion Rate:** The rate of episodes that reach $T_{\max}$ without a valid final declaration, reflecting the model's ability to complete identification within the interaction budget.
6. **Trajectory Deduction Score:** The quality of the Experimenter's experimental reasoning trajectory, assessed by "LLM-as-a-Judge" across five dimensions: Candidate and Hypothesis Management (CHM), Diagnostic Experiment Design (DED), Observation Interpretation Fidelity (OIF), Evidential Sufficiency and Uniqueness (ESU), and Evidence-Gated Decision Efficiency (EGDE). Detailed definitions and scoring rubrics for each dimension are provided in Appendix F.

4 Experiments

4.1 Experimental Setup

Models. We evaluate eight frontier models: GPT-5.6-Sol (OpenAI, 2026), Qwen3.8-Max (Qwen Team, 2026), Gemini-3.7-Flash (Gemini Team, 2026), GLM-5.3 (GLM Team, 2026a), GLM-5.2 (GLM Team, 2026b), MiniMax-M3 (Lai et al., 2026), Grok-4.3 (xAI, 2026), and Kimi-K3 (Team et al., 2026). More details about these models are presented in Appendix H.

Evaluation protocol. To ensure a fair and reproducible evaluation, we adopt a unified experimental setting for all models. Specifically, each Experimenter model is evaluated on all 200 target samples, yielding $1{,}600$ model–target pairs, with three independent runs to reduce sampling randomness, resulting in $4{,}800$ episodes in total. LabAgent uniformly uses DeepSeek-V4-Pro (Xu et al.,

Table 1: Unknown chemical identification results (mean ± standard error) on ChemDeduce, including Acc. (identification accuracy), Op. Comp. (operational compliance), PSR (proactive-step ratio), Avg. Rounds (mean interaction rounds), BER (budget exhaustion rate), and TDS (trajectory deduction score).

Model	Acc. ↑	Op. Comp. ↑	PSR ↑	Avg. rounds ↓	BER ↓	TDS ↑
GPT-5.6-Sol	67.33 ± 1.77	88.29 ± 0.42	81.71 ± 0.64	6.97 ± 0.05	16.50 ± 1.80	65.06 ± 0.94
Qwen3.8-Max	60.67 ± 1.36	80.43 ± 0.34	76.14 ± 0.49	6.88 ± 0.08	17.00 ± 1.26	64.11 ± 1.05
Gemini-3.7-Flash	55.00 ± 0.50	71.88 ± 1.14	69.98 ± 0.63	6.94 ± 0.13	12.33 ± 1.45	55.26 ± 0.32
GLM-5.3	48.83 ± 0.93	75.45 ± 0.70	71.13 ± 0.74	8.16 ± 0.06	38.00 ± 1.15	60.34 ± 0.66
GLM-5.2	39.50 ± 0.29	64.96 ± 0.66	61.73 ± 0.64	8.67 ± 0.08	49.50 ± 0.29	54.95 ± 0.69
MiniMax-M3	33.17 ± 0.73	64.08 ± 0.32	59.93 ± 0.47	8.22 ± 0.10	41.00 ± 1.04	46.15 ± 0.18
Grok-4.3	29.17 ± 1.09	60.22 ± 0.16	56.54 ± 0.28	8.33 ± 0.03	37.83 ± 1.02	39.31 ± 0.48
Kimi-K3	23.83 ± 1.02	79.53 ± 0.98	73.47 ± 0.55	8.73 ± 0.02	60.17 ± 0.66	44.17 ± 0.55

2026) as the LLM controller. Model context and laboratory state are reset between episodes to ensure independence across evaluations. During Experimenter inference, the temperature is set to 0.3, with a maximum output length of 8,192 tokens. At the beginning of each episode, the Experimenter receives the public laboratory resources together with the target sample's initial physical state and macroscopic appearance, but not the target identity or target-conditioned records. Each episode ends upon final identification or after $T_{\max} = 10$ interaction rounds. Unless otherwise specified, each reported mean is averaged over the three independent repetitions, and the standard error is computed across these repetitions.

4.2 Main Results

Table 1 presents the unknown chemical identification results of eight LLMs on ChemDeduce. The evaluation reveals a clear hierarchy in performance, with identification accuracy ranging from 23.83% to 67.33%. Notably, even frontier LLMs remain below 70% accuracy under our framework, highlighting the challenge of sequential chemical identification. Interestingly, Kimi-K3 achieves 79.53% operational compliance and a 73.47% proactive-step ratio, exceeding Gemini-3.7-Flash and GLM-5.2 on these execution-oriented metrics, yet attaining the lowest identification accuracy. This indicates that stronger experimental execution does not necessarily translate into better identification without effective interpretation and integration of experimental evidence.

Moreover, interaction rounds are not consistently associated with higher identification accuracy. Qwen3.8-Max was the most efficient, requiring only 6.88 rounds on average, yet achieved only moderate identification accuracy. The top-performing model, GPT-5.6-Sol, demonstrated a more effective and robust strategy, whereas Kimi-K3 required the most interactions while attaining the lowest accuracy. This suggests that interaction efficiency depends on how effectively each experiment narrows the candidate space rather than simply increasing interactions. Meanwhile, budget exhaustion is more pronounced among weaker models, reaching 60.17% in Kimi-K3 and 49.50% in GLM-5.2, substantially exceeding that of stronger models, suggesting that weaker models are more likely to fail to reach a valid final declaration within the limited interaction budget.

Figure 2 further analyzes the experimental reasoning process across five TDS dimensions. A critical finding is that evidence integration and final decision-making remain prominent weaknesses across all evaluated LLMs. In terms of Evidential Sufficiency and Uniqueness (ESU) and Evidence-Gated Decision Efficiency (EGDE), model scores generally fall in the low to moderate range, far below their respective maximum scores. Even the strongest model, GPT-5.6-Sol, achieves only 18.48 out of 30 on

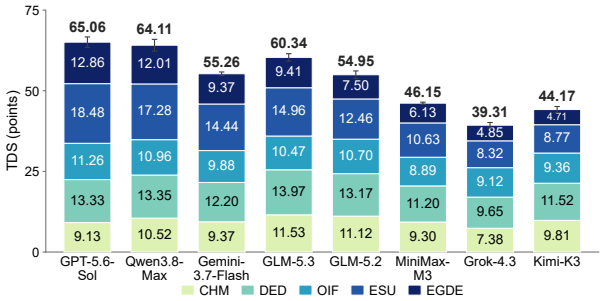


Figure 2: Trajectory Deduction Score (TDS, out of 100) across five dimensions: CHM (15), DED (20), OIF (15), ESU (30), and EGDE (20).

ESU and 12.86 out of 20 on EGDE, indicating that current models still struggle to integrate multi-round observations into sufficiently discriminative evidence and translate that evidence into reliable final decisions. Notably, these two dimensions also exhibit the largest variation across models, with gaps of 10.16 and 8.11 points between the highest and lowest scores, respectively. This suggests that evidential sufficiency and the quality of final decisions are key factors that differentiate models in experimental reasoning. In contrast, most models show relatively stable performance on Candidate and Hypothesis Management (CHM), Diagnostic Experiment Design (DED), and Observation Interpretation Fidelity (OIF), with comparatively small variation across models, showing that current LLMs already possess relatively stable basic experimental reasoning capabilities.

4.3 Failure Analysis

To characterize fine-grained error differences across models, we categorize incorrect declarations into five types: *misread*, *premature*, *near-neighbor*, *off-track*, and *others*, reflecting failures in evidence handling, decision timing, and discrimination among similar candidates. As shown in Figure 3, *misread* and *premature* are the most prevalent error types across models, with average proportions of 37.3% and 42.5%, respectively, substantially higher than the 7.8% and 10.6% observed for *near-neighbor* and *off-track*. Notably, *premature* errors are not driven by explicit pressure from the interaction budget, as the Experimenter is not informed of the exact round limit. These findings indicate that the primary bottleneck of current models lies in accurately interpreting and integrating available evidence and making reliable decisions only when the evidence is sufficient.

A fine-grained analysis across Experimenter models reveals distinct error patterns. Stronger models such as GPT-5.6-Sol and Qwen3.8-Max are dominated by *premature* errors, whereas weaker models exhibit more *misread* errors. In particular, *misread* accounts for 58% and 48% of the errors made by Kimi-K3 and MiniMax-M3, respectively, highlighting difficulties in interpreting observations and resolving contradictory evidence. Interestingly, the GLM models (i.e., GLM-5.3 and GLM-5.2) exhibit the fewest *misread* errors in absolute percentage, yet show a more pronounced tendency to end without a final declaration, indicating that fewer evidence-interpretation errors do not necessarily translate into stronger overall identification performance.

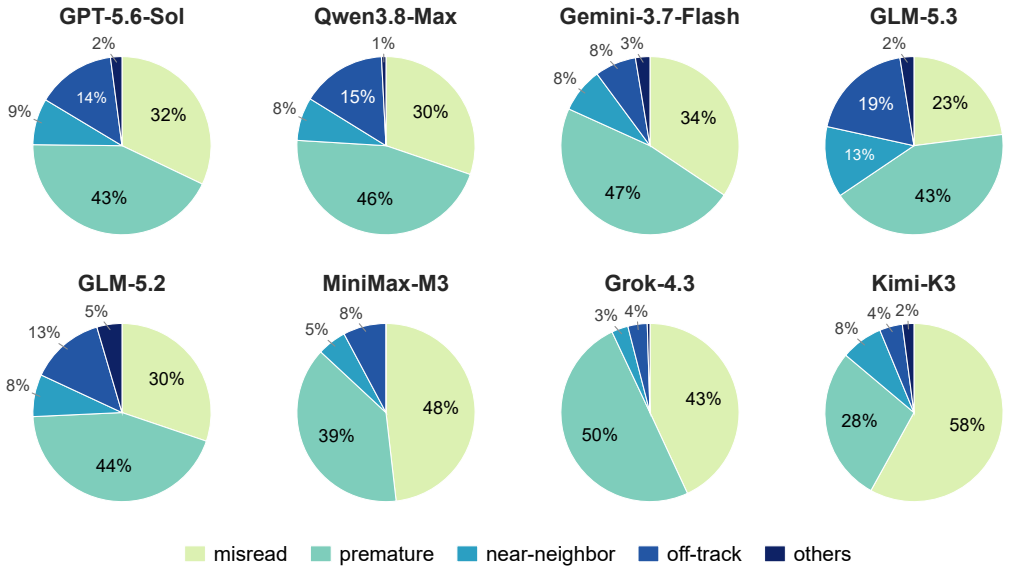


Figure 3: Error type distribution among incorrect identifications across eight Experimenter models. Each pie shows the proportions of five categories: *misread* (contradicted evidence), *premature* (premature conclusion), *near-neighbor* (close-relative error), *off-track* (off-target search), and *others*.

4.4 Sensitivity of LabAgent Controller

To further analyze the sensitivity of ChemDeduce to the LabAgent controller, we keep all other evaluation settings unchanged and evaluate three Experimenter models under three LabAgent con-

troller configurations: DeepSeek-V4-Pro, StepFun-3.7-Flash, and Qwen3.8-Flash. As shown in Figure 4 (a), the relative ranking of the three models remains unchanged across all controllers, while the accuracy variation of each model is within 5.0 percentage points. These results suggest that ChemDeduce is robust across different LabAgent controllers and provides consistent comparative conclusions across Experimenter models.

4.5 Budget Sensitivity

To further examine the effect of interaction budget on ChemDeduce, we evaluate three Experimenter models, Kimi-K3, Grok-4.3, and MiniMax-M3, under interaction budgets of $T_{\max} = 10$, 20, and 30. These models show high budget exhaustion and low identification accuracy under the default $T_{\max} = 10$ setting. As shown in Figure 4 (b), all three models exhibit a consistent trend. Increasing the budget from $T_{\max} = 10$ to $T_{\max} = 20$ improves identification accuracy and reduces budget exhaustion, whereas further increasing it to $T_{\max} = 30$ yields little additional gain. This pattern suggests that increasing the interaction budget improves task completion when the budget is restrictive, whereas beyond this point performance is limited more by experiment selection and evidence integration. This supports the use of $T_{\max} = 10$ as our default resource-constrained setting: the budget is restrictive enough to expose differences in trajectory quality, yet not so restrictive that success is primarily determined by the interaction cap. The sensitivity analysis is included to show how the evaluated budget-limited models respond when the constraint is relaxed.

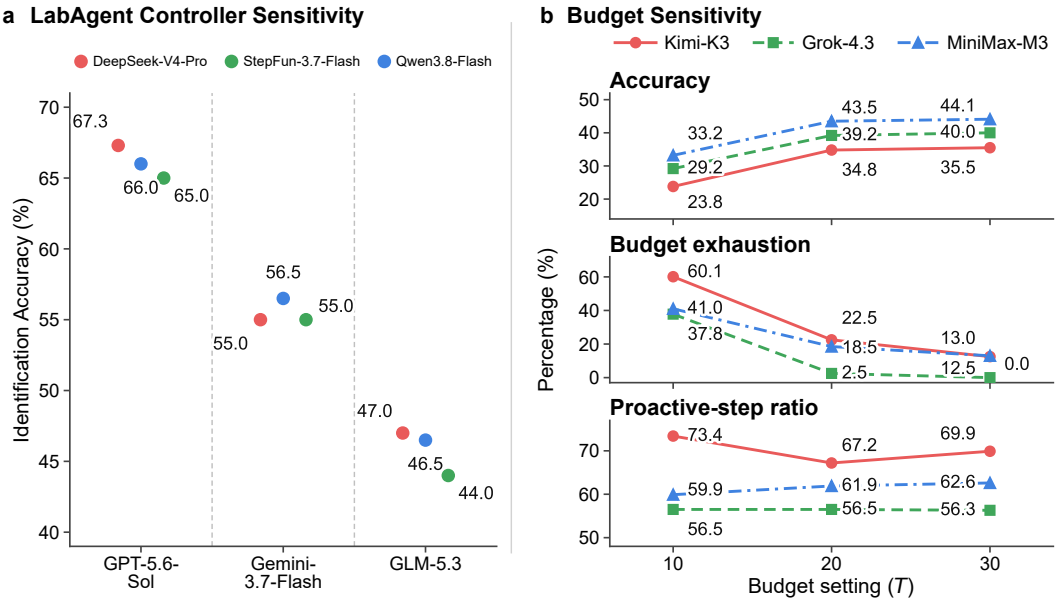


Figure 4: Sensitivity analyses. (a) Identification accuracy of Experimenter models across three LabAgent controllers (DeepSeek-V4-Pro, Qwen3.8-Flash, and StepFun-3.7-Flash). (b) Identification performance of three Experimenter models under interaction budgets of $T_{\max} = 10$, 20, and 30.

5 Conclusion

In this work, we propose an interactive simulated-experiment framework for evaluating LLMs on unknown chemical identification. ChemDeduce moves beyond static chemistry benchmarks by formulating identification as a sequential experimental process in which models must actively design experiments, interpret observations, and update hypotheses from accumulated evidence. With 200 chemicals and a stateful simulated laboratory, ChemDeduce enables controlled and reproducible evaluation of iterative experimental reasoning. Our experiments reveal that current frontier LLMs can generate plausible experimental plans but still struggle with informative experiment selection, evidence interpretation, and multi-round hypothesis refinement, highlighting a substantial gap between static chemical reasoning and closed-loop experimental reasoning.

AI USE STATEMENT

The language models evaluated in this work are the objects of study and were used within the research system under the experimental protocol described in the paper; this is separate from the use of generative AI tools for manuscript preparation. In preparing the manuscript, generative AI tools were used only for drafting assistance, language polishing, and formatting, and all AI-assisted text was reviewed and revised by the authors. These manuscript-preparation tools were not used to create, recompute, or modify the reported measurements, which were computed from recorded evaluation traces generated under the stated protocol. The authors take full responsibility for the correctness, integrity, and presentation of the final work.

ETHICS STATEMENT

This research aims to improve the reliability and safety of language models in scientific reasoning and laboratory decision-making. All experiments in this work are conducted in a text-mediated simulated laboratory environment and do not involve human participants, personally identifiable information, or physical chemical experiments. While the benchmark is designed to reflect realistic laboratory operations and may help inform future real-world experimental workflows, it does not replace professional training, expert supervision, or institutional safety review. Any deployment in a physical laboratory should be conducted only under qualified human oversight and appropriate safety controls. We therefore foresee no significant ethical issues arising from this study.

REPRODUCIBILITY STATEMENT

We are committed to supporting the reproducibility of our research. We will release our source code containing the simulated laboratory, the 200-sample library, prompts, configuration files, and evaluation scripts. The paper provides details on the evaluated models, experimental protocol, prompts, and evaluation metrics. These resources will enable the community to inspect the benchmark protocol, reproduce the simulator-side evaluation pipeline, and replicate the reported analyses to the extent permitted by access to the same or comparable model APIs.

REFERENCES

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Benjamin Burger, Phillip M Maffettone, Vladimir V Gusev, Catherine M Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M Alston, Buyi Li, Rob Clowes, et al. A mobile robotic chemist. *Nature*, 583(7815):237–241, 2020.

Gary D Christian. *Analytical chemistry*. Wiley New York, 1977.

Brian S Furniss. *Practical organic chemistry*. Pearson Education India, 2004.

Gemini Team. Gemini 3.7 flash model card. Google DeepMind, 2026.

GLM Team. GLM-5.3: Frontier coding and emergent cybersecurity capabilities. Zhipu AI, 2026a.

GLM Team. GLM-5.2: The 1m-context flagship glm model. Zhipu AI, 2026b.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in neural information processing systems*, 36:59662–59688, 2023.

Florian Häse, Loïc M Roch, and Alán Aspuru-Guzik. Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories. *Chemical science*, 9(39):7642–7655, 2018.

William M Haynes. *CRC handbook of chemistry and physics*. CRC press, 2016.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv:2009.03300*, 2020.

Yuqing Huang, Rongyang Zhang, Xuesong He, Xuyang Zhi, Hao Wang, Nuo Chen, Zongbo Liu, Xin Li, Feiyang Xu, Deguang Liu, et al. Chemeval: A multi-level and fine-grained chemical capability evaluation for large language models. In *International Conference on Learning Representations*, volume 2026, pp. 21365–21392, 2026.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in neural information processing systems*, 37:10088–10116, 2024.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, et al. Minimax sparse attention. *arXiv:2606.13392*, 2026.

Rui Li, Zixuan Hu, Wenxi Qu, Jinouwen Zhang, Zhenfei Yin, Sha Zhang, Xuantuo Huang, Hanqing Wang, Tai Wang, Jiangmiao Pang, et al. Labutopia: High-fidelity simulation and hierarchical benchmark for scientific embodied agents. *Advances in Neural Information Processing Systems*, 38, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Xuan Liu, Siru Ouyang, Xianrui Zhong, Jiawei Han, and Huimin Zhao. Fgbench: A dataset and benchmark for molecular property reasoning at functional group-level in large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.

Benjamin P MacLeod, Fraser GL Parlane, Thomas D Morrissey, Florian Häse, Loïc M Roch, Kevan E Dettelbach, Raphaell Moreira, Lars PE Yunker, Michael B Rooney, Joseph R Deeth, et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20): eaaz8867, 2020.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, 2025.

OpenAI. GPT-5.6 system card. OpenAI, 2026.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Alibaba Cloud, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv:2311.12022*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv:2405.07960*, 2024.

Ralph L Shriner, Christine KF Hermann, Terence C Morrill, David Y Curtin, and Reynold C Fuson. *The systematic identification of organic compounds*. John Wiley & Sons, 2003.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv:2607.24653*, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv:2307.10635*, 2023.

xAI. Grok 4.3: Model card. xAI, 2026.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv:2210.03629*, 2022.

Botao Yu, Frazier N Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. Chemtoolagent: The impact of tools on language agents for chemistry problem solving. *arXiv:2411.07228*, 2024.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv:2402.06852*, 2024.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

APPENDIX

A LIMITATIONS

While ChemDeduce provides a simulated environment for evaluating laboratory-oriented scientific reasoning, several limitations remain. First, ChemDeduce abstracts laboratory identification into text-based interaction, improving reproducibility but excluding multimodal perception from images, videos, instrument displays, and subtle visual cues. Second, observations are represented through qualitative phenomena and partial instrument readings whose precision is constrained by the simulator settings, which reduces over-interpretation but does not fully capture real-world variability such as contamination, reagent aging, instrument drift, calibration error, and manual handling. Future work could incorporate multimodal observations, finer-grained measurement models, and more realistic physical noise to further bridge the gap between simulated and real-world laboratories.

B COMPARISON WITH EXISTING BENCHMARKS

Table A1 summarizes the relationship between ChemDeduce and representative benchmarks for chemical reasoning, scientific question answering, and interactive agent evaluation. Existing chemistry benchmarks primarily evaluate static knowledge, molecular reasoning, or predefined tool-use tasks. General interactive benchmarks evaluate sequential decision-making but are not designed for unknown chemical identification. ChemDeduce differs by requiring a model to actively acquire evidence through stateful simulated experiments and to identify a hidden chemical under operational constraints.

Table A1: Comparison between ChemDeduce and representative chemistry, scientific, and interactive-agent benchmarks. "Simulated Env." indicates whether the benchmark provides a simulated environment for interactive task execution. "Stateful" indicates whether the environment maintains persistent task state across turns. "Experiment Design" indicates whether the model must choose informative experiments rather than answer static questions.

Benchmark	Domain	Simulated Env.	Static QA	Interactive	Stateful	Experiment Design	Chemical ID
MMLU (Hendrycks et al., 2020)	General science	–	✓	–	–	–	–
GPQA (Rein et al., 2023)	Graduate science	–	✓	–	–	–	–
ChemLLMBench (Guo et al., 2023)	Chemistry	–	✓	–	–	–	–
ChemBench (Mirza et al., 2025)	Chemistry	–	✓	–	–	–	–
ChemEval (Huang et al., 2026)	Chemistry	Limited	✓	Limited	Limited	Limited	–
ChemLLM (Zhang et al., 2024)	Chemistry	–	✓	–	–	–	–
FGBench (Liu et al., 2026)	Functional groups	–	✓	–	–	–	–
AgentBench (Liu et al., 2024)	General agents	✓	–	✓	✓	–	–
WebArena (Zhou et al., 2024)	Web agents	✓	–	✓	✓	–	–
ToolSandbox (Lu et al., 2025)	Tool use	✓	–	✓	✓	–	–
AppWorld (Trivedi et al., 2024)	App agents	✓	–	✓	✓	–	–
ScienceWorld (Wang et al., 2022)	Science tasks	✓	–	✓	✓	Partial	–
DiscoveryWorld (Jansen et al., 2024)	Scientific discovery	✓	–	✓	✓	Partial	–
AgentClinic (Schmidgall et al., 2024)	Clinical diagnosis	✓	–	✓	✓	Diagnostic tests	–
ChemDeduce	**Chemical identification**	✓	–	✓	✓	✓	✓

The comparison highlights three distinguishing properties of ChemDeduce. First, the answer is hidden and must be inferred from newly acquired experimental evidence rather than recalled from a question statement. Second, the environment is stateful: an operation changes sample, vessel, apparatus, and observation states, which affect subsequent interactions. Third, evaluation covers not only final-answer accuracy but also operational compliance, interaction efficiency, budget exhaustion, and process-level evidence use.

C DATASET DETAILS

C.1 DATASET CONSTRUCTION PIPELINE

The ChemDeduce dataset is constructed to support controlled unknown chemical identification in a simulated laboratory. The construction pipeline consists of four stages.

Source collection. We collect candidate substances and experimental knowledge from standard analytical and practical chemistry references, including *Analytical Chemistry*, *The Systematic Identification of Organic Compounds*, *Vogel's Textbook of Practical Organic Chemistry*, and the *CRC Handbook of Chemistry and Physics*. The collected information includes physical state, appearance, basic physicochemical properties, qualitative reactions, available reagents, apparatus requirements, operating conditions, and observable experimental outcomes.

Normalization and structuring. Raw records are converted into a unified structured schema. Chemical names, aliases, physical states, class labels, reagent names, apparatus names, and observation descriptions are normalized to reduce ambiguity. When multiple sources describe the same test, we reconcile differences by prioritizing standard qualitative-analysis conditions and expert-verified observations.

Expert review. Three reviewers with chemistry backgrounds independently evaluate candidate records according to chemical correctness, experimental plausibility, and record consistency. A chemical is retained only if it receives majority approval. Records with ambiguous identities, unsupported observations, conflicting test outcomes, or insufficient distinguishing information are revised or removed.

Simulator integration. The verified records are converted into target-conditioned knowledge profiles used by LabAgent. These profiles specify hidden target identity, public initial appearance, allowable observations, relevant qualitative reactions, instrument responses, safety constraints, and waste-handling requirements. During evaluation, the complete target-conditioned profiles remain hidden from the Experimenter.

C.2 Dataset Statistics

The final benchmark contains 200 real chemical substances, including 102 solids and 98 liquids. The library covers inorganic substances, elemental samples, and common organic functional-group families. Table A2 reports the mean identification accuracy of each model by chemical class. These results provide a complementary view of the dataset by showing that different categories induce substantially different levels of difficulty. For example, elemental substances and inorganic salts are generally easier to identify, whereas halogenated compounds, amines/amides, and several organic functional-group classes are more challenging. This suggests that ChemDeduce contains not only diverse chemical categories, but also heterogeneous identification difficulty across categories.

Table A2: Identification accuracy (%) by chemical class at $T_{\max} = 10$, averaged over the three repeats. n is the number of targets in the class and *mean* averages the eight models. Solid and Liquid split the same 200 targets by the sample's physical state. For each row, the best model score is shown in bold and the second-best score is underlined; ties are all marked.

Class	n	GPT-5.6	Qwen3.8	Gemini	GLM-5.3	GLM-5.2	MiniMax	Grok-4.3	Kimi-K3	mean
Halogenated	11	**57.6**	48.5	42.4	21.2	24.2	12.1	9.1	6.1	27.7
Amine / amide	22	**45.5**	39.4	39.4	30.3	22.7	10.6	9.1	7.6	25.6
Phenol	10	**63.3**	40.0	36.7	46.7	30.0	16.7	30.0	16.7	35.0
Ether	8	**58.3**	50.0	37.5	37.5	37.5	33.3	20.8	16.7	36.5
Ester	13	**69.2**	**69.2**	43.6	20.5	28.2	43.6	23.1	10.3	38.5
Carboxylic acid	22	**60.6**	50.0	47.0	47.0	37.9	21.2	24.2	19.7	38.4
N,S-heterocycle	10	**66.7**	43.3	50.0	40.0	36.7	23.3	10.0	42.1
Alcohol	22	**78.8**	71.2	63.6	50.0	33.3	39.4	27.3	16.7	47.5
Ketone	11	**78.8**	63.6	69.7	39.4	15.2	30.3	6.1	15.2	39.8
Aldehyde	10	**63.3**	**63.3**	53.3	60.0	36.7	40.0	26.7	23.3	45.8
Carbohydrate	8	**79.2**	70.8	75.0	62.5	50.0	33.3	45.8	33.3	56.3
Hydrocarbon	12	**91.7**	69.4	63.9	61.1	58.3	47.2	41.7	38.9	59.0
Inorganic salt	36	67.6	**71.3**	67.6	70.4	60.2	44.4	49.1	47.2	59.7
Elemental	5	86.7	80.0	**93.3**	80.0	80.0	80.0	86.7	66.7	81.7
Solid	102	**62.7**	56.2	54.6	51.0	42.8	30.4	35.3	29.1	45.3
Liquid	98	**72.1**	65.3	55.4	46.6	36.1	36.1	22.8	18.0	44.0
All targets	200	**67.3**	60.7	55.0	48.8	39.5	33.2	29.2	23.8	44.7

C.3 Dataset Record Examples

The ChemDeduce dataset contains two types of structured records: target-conditioned chemical records and laboratory resource records. Target-conditioned records describe hidden substances, including their verified names, aliases, physical states, macroscopic appearances, physicochemical properties, qualitative reactions, and supported measurements. Laboratory resource records describe available instruments, apparatus, reagents, accessories, storage locations, usage constraints, and safety information. The following examples illustrate the storage format used in the dataset.

Solid Sample Profile: Sodium Chloride

Basic Identity

- **Name**: Sodium chloride
- **Aliases**: Table salt; NaCl; common salt
- **State**: Solid

Physical Properties

- **Melting Point**: $801^\circ\mathrm{C}$
- **Boiling Point**: Not available
- **Specific Rotation**: Not applicable
- **Refractive Index at** $20^\circ\mathrm{C}$: Not available
- **Conductivity at** $25^\circ\mathrm{C}$: $85.0\,\mathrm{mS/cm}$
- **IR Characteristic Peaks**: Not available

Macroscopic Appearance

- White cubic crystals

Curated Qualitative Reactions

- **Reagent / Operation**: Platinum wire flame test
  - **Phenomenon**: The flame appears yellow.
- **Reagent / Operation**: Silver nitrate solution
  - **Phenomenon**: Adding silver nitrate solution to the aqueous solution produces a white precipitate. The precipitate does not dissolve after adding dilute nitric acid.

Liquid Sample Profile: Methanol

Basic Identity

- **Name**: Methanol
- **Aliases**: Methanol; wood alcohol; industrial alcohol
- **State**: Liquid

Physical Properties

- **Melting Point**: $-98^\circ\mathrm{C}$
- **Boiling Point**: $64.7^\circ\mathrm{C}$
- **Refractive Index at** $20^\circ\mathrm{C}$: 1.3288
- **Conductivity at** $25^\circ\mathrm{C}$: Not available
- **IR Characteristic Peaks**: Not available

Macroscopic Appearance

- Colorless liquid with an ethanol-like odor

Curated Qualitative Reactions

- **Reagent / Operation**: Metallic sodium
  - **Phenomenon**: Adding metallic sodium produces colorless gas bubbles.
- **Reagent / Operation**: Lucas reagent
  - **Phenomenon**: No turbidity is observed. As a primary alcohol, methanol does not react even upon heating.
- **Reagent / Operation**: Iodoform test
  - **Phenomenon**: Negative result. The sample does not contain the required structural motif for a positive iodoform reaction.

Instrument Profile: Refractometer

Basic Information

- **Name**: Refractometer
- **Quantity**: 1
- **Storage Location**: Center of the instrument bench

Required Accessories

- Lens tissue
- Rubber dropper

Sample Preparation

- Use a rubber dropper to place 2–3 drops of the liquid sample onto the prism surface of the refractometer.
- Close the prism cover.
- Adjust the reflector toward the light source until a clear bright–dark boundary appears in the field of view.

Usage Constraints

- The refractometer is applicable only to liquid samples.
- The prism surface must not contact hard objects or strongly corrosive liquids.
- After use, the prism surface must be gently cleaned with lens tissue moistened with ethanol.

Notes

- The test temperature is fixed at $25^{\circ}\mathrm{C}$.
- The measurement range is $n_D = 1.3000$–1.7000.

General Apparatus Profile: Beaker$(250\,\mathrm{mL})$

Basic Information

- **Name**: Beaker, $250\,\mathrm{mL}$
- **Quantity**: 8
- **Storage Location**: Middle shelf of the apparatus cabinet
- **Material**: Borosilicate glass

Role in the Virtual Laboratory

- The beaker is treated as a general-purpose container for operations such as dissolving, mixing, temporary liquid transfer, and observation of macroscopic phenomena.
- Its availability is checked by LabAgent when an experimental procedure requires a generic vessel.

Reagent Profile: Dilute Hydrochloric Acid

Basic Information

- **Name**: Dilute hydrochloric acid
- **Aliases**: Dilute HCl; HCl(aq); hydrochloric acid
- **State**: Liquid
- **Quantity**: Sufficient
- **Concentration**: Approximately $2\,\mathrm{mol/L}$

Container and Storage

- **Container**: Transparent glass reagent bottle, $500\,\mathrm{mL}$, with a ground-glass stopper
- **Storage Location**: Acidic reagent area of the reagent rack

Hazards

- Corrosive liquid.
- Avoid contact with skin and eyes.
- Reacts with metals and releases hydrogen gas.

Role in the Virtual Laboratory

- This reagent can be selected by the Experimenter for supported acid-addition or acid-confirmation tests.
- LabAgent checks whether its use is compatible with the sample state, requested operation, apparatus, and safety constraints before execution.

C.4 Information Isolation

The dataset contains both public and hidden information. Public information includes the available laboratory resources, their usage constraints, and the initial physical state and appearance of the unknown sample. Hidden information includes the target identity, verified aliases, target-conditioned reaction records, diagnostic cues, and answer-resolution records. During evaluation, the Experimenter can only access public laboratory information and observations returned by LabAgent, while LabAgent can access the hidden target-conditioned records required for state-consistent simulation.

D Simulated Laboratory Environment

All experiments in ChemDeduce are conducted in a simulated chemistry laboratory built in NVIDIA Isaac Sim. The environment builds on the LabUtopia simulation platform (Li et al., 2025) and recreates a real laboratory bench with the glassware and instruments that qualitative identification procedures require, so that each requested experiment can be executed and recorded in a controlled setting.

The simulation infrastructure, the robotic arm, and the layered interface for defining and executing tasks all come from LabUtopia. Its chemistry assets, however, are limited to a few common items such as beakers and measuring cylinders, and its action vocabulary consists of generic manipulations such as shaking and pouring, which do not cover the qualitative procedures we evaluate. On top of the platform we add the chemistry layer: 30 chemical actions in the action catalog that the robot can execute, and 70 pieces of equipment and 31 experiment scenes in the asset library, all assembled by generation scripts from shared assets rather than modeled one by one.

The Cartesian controller provided by the platform (RMPflow) is not sufficient on its own, because operations such as pushing a test tube back into its rack or lowering a dropper into a bottle neck have to land within about a millimeter. We therefore add an inverse-kinematics engine that solves each target pose and validates the solution before the arm moves. Chemical phenomena are likewise computed rather than replayed: color, liquid level, bubbles, precipitates and layered phases are up-

dated for every frame from the experiment parameters and the current configuration of the arm. The same scene can thus produce different outcomes under different parameters, while a fixed parameter set reproduces the same video.

The asset library includes the equipment required by our bench protocols: laboratory benches with shelves of reagent bottles, sinks, gas and power outlets; glassware such as test tubes and racks, beakers, conical flasks, funnels, droppers, glass rods, spatulas, and weighing bottles; instruments including a balance, a refractometer, a conductivity meter, pH paper, and thermometers; heating devices such as an alcohol lamp, a Bunsen burner, and a hot plate; and the reagents used in the qualitative tests. The chemical substances are drawn from the sample library described in Section 3.1. Because the apparatus is kept as a reusable asset library, experiments that require common equipment are assembled from shared assets rather than implemented as separate scenes.

The environment supports five categories of operations: instrumental measurement, thermal manipulation, flame and combustion tests, wet-chemistry procedures, and physical inspection. Instrumental measurements directly return readings, such as the call-out from the conductivity meter. Wet-chemistry and thermal operations produce observable macroscopic phenomena, including dissolution, color change, gas evolution, and the formation or dissolution of precipitates. Flame tests return the characteristic color imparted to the flame by the sample. Because many of these phenomena occur at the centimeter scale and can be difficult to observe from a wide-angle view, the environment records them using dedicated close-up cameras in addition to an overview camera. The flame test and precipitation reaction shown in Figure A1 illustrate two such close-up views.



Figure A1: The simulated laboratory environment of ChemDeduce. The overview camera covers the bench, its glassware and its instruments; the three call-outs show phenomena that the environment can produce and record in close-up, namely a conductivity reading, the flame of a flame test before and after the sample is introduced, and a precipitation reaction with an illustrative equation shown only for visualization.

E Case Study

E.1 Case Study Structure

This section presents three complete cases taken from the recorded evaluation episodes, which show what a run of ChemDeduce looks like in practice: two successful identifications, on a solid sample (sodium salicylate) and on a liquid sample (2-propanol), and one failure, on a solid sample (resorcinol) that the model finally declares to be phenol.

Each case is presented in two parts. The first part is a dialogue-style record, which reports the Experimenter's reasoning and the natural-language instruction it issues in every round, the primitive actions that LabAgent maps the instruction to together with their execution result or rejection message, and the observation returned by the simulated experiment. The second part is the corresponding keyframe figure (Figures A2–A4): every round occupies one row, which shows four keyframes of that round's experiment video from left to right, with the executed action labelled under each frame and the evidence gained from the round tagged at the right of the row. A round rejected by the LabAgent safety check is shown as a single bar without frames, and the last round reports the identity declared by the model together with whether the answer is correct.

E.2 Successful Identification Case

Dialogue History of Solid Identification

Target: Sodium Salicylate (Solid)

Round 1

- **Experimenter (Reasoning):** Sample is a white crystalline solid. First test its water solubility and check the pH of the aqueous solution to quickly determine its acid-base properties.
- **Experimenter (Instruction):** Take a test tube and a surface dish. Add a small amount of sample, add 2 mL of distilled water, and shake to observe dissolution. Dip a piece of wide-range pH paper into the solution using a glass rod and observe the color change.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (Distilled Water) → `instrument_operation` (pH paper).
- **Observation:** White crystalline powder gradually dissolves to form a colorless transparent solution. The pH paper turns green to blue-green.

Round 2

- **Experimenter (Reasoning):** The aqueous solution is weakly alkaline. Add dilute hydrochloric acid to observe if gas is released or precipitate forms to test the anion type.
- **Experimenter (Instruction):** Take a test tube, add sample, add 2 mL distilled water, and dissolve. Add 1 mL of dilute hydrochloric acid dropwise, shake, and observe.
- **LabAgent:** `validate_operation` (FAIL).
- **Observation:** Safety check failed. The requested operation involves a corrosive liquid (dilute hydrochloric acid) but lacks mandatory personal protective equipment (goggles and gloves).

Round 3

- **Experimenter (Reasoning):** Added safety protection. Add dilute hydrochloric acid to the sample solution to observe if bubbles are produced to test for carbonate/bicarbonate.
- **Experimenter (Instruction):** Wear goggles and nitrile gloves. Take a test tube, add sample, add 2 mL distilled water. Add 1 mL of dilute hydrochloric acid dropwise, shake gently, and observe.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (Dilute HCl).
- **Observation:** After adding dilute hydrochloric acid, a white precipitate forms in the test tube.

Round 4

- **Experimenter (Reasoning):** Adding acid precipitates a white solid, suspected to be a weak acid sodium salt. Use ferric chloride solution to test for characteristic color to identify the acid radical.
- **Experimenter (Instruction):** Wear goggles and gloves. Take a test tube, add sample, add 2 mL distilled water. Add 2-3 drops of ferric chloride solution, shake, and observe the color change.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (Ferric Chloride Solution).
- **Observation:** After adding ferric chloride solution and shaking, the solution turns purple.

Round 5

- **Experimenter (Reasoning):** Purple color indicates salicylate or phenol. Perform a flame test to confirm the metal cation.
- **Experimenter (Instruction):** Wear goggles. Light the alcohol lamp. Clean a platinum wire with dilute HCl and heat until colorless. Dip the wire into the sample solution and place it in the outer flame. Observe the flame color.
- **LabAgent:** `validate_operation` (PASS) → `instrument_operation` (Flame test).
- **Observation:** The flame is yellow.

Round 6

- **Experimenter (Reasoning):** Sample is a white crystalline solid, soluble in water, weakly alkaline; acidification yields a white precipitate, indicating a slightly soluble weak acid salt; characteristic purple color with $FeCl_3$ indicates salicylic acid structure; flame test shows bright yellow, confirming sodium ion (Na^+). Comprehensively locked as sodium salicylate.
- **Experimenter (Instruction):** Identification: Sodium salicylate.
- **LabAgent:** End.
- **Observation:** Experimenter conclusion: Sodium salicylate. Rounds = 6, Reason = experimenter concluded.

Final Result: Sodium Salicylate ✓ (Correct)



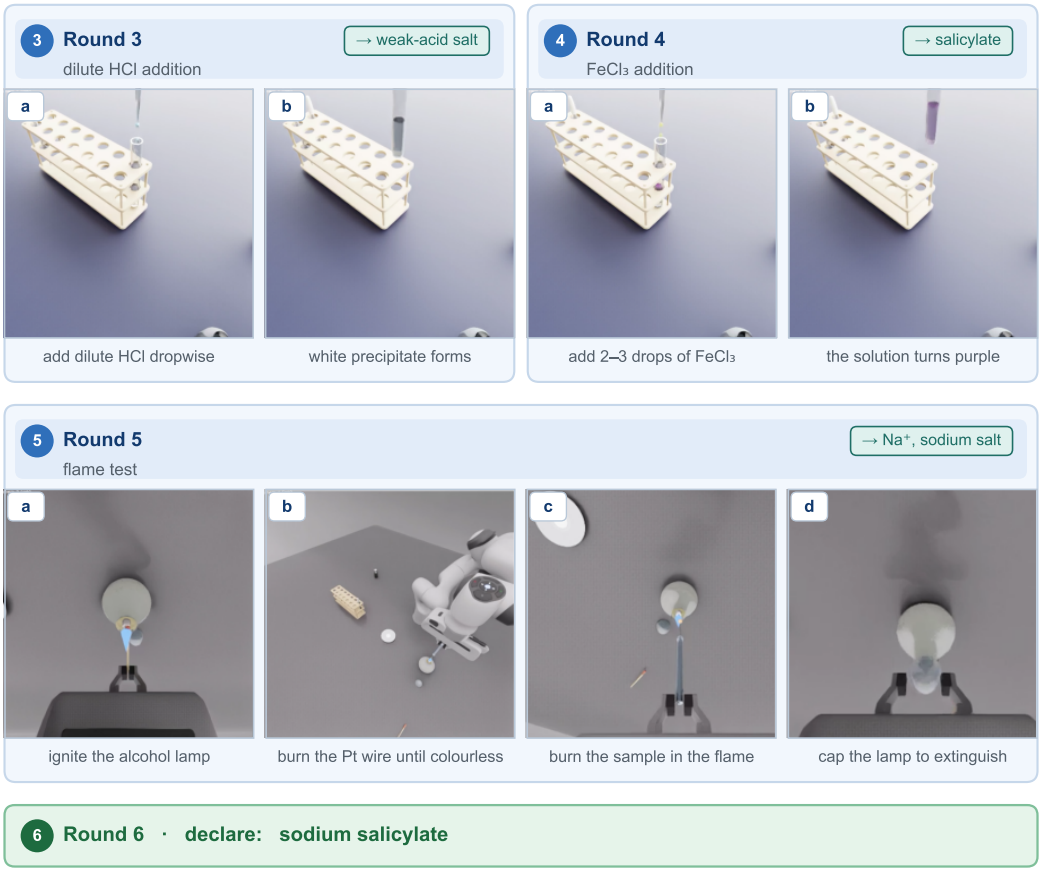


Figure A2: Case study of a successful solid identification in ChemDeduce: the hidden target is sodium salicylate, which the Experimenter declares correctly in six rounds. Round 1 covers solubility and pH, Round 2 is rejected by the safety checker, Rounds 3 and 4 are two reagent additions, Round 5 is a flame test, and Round 6 declares the identity.

Dialogue History of Liquid Identification

Target: 2-Propanol (Liquid)

Round 1

- **Experimenter (Reasoning):** Unknown sample is a colorless liquid. Measuring the refractive index efficiently obtains a characteristic physical constant to narrow the scope.
- **Experimenter (Instruction):** Approach the refractometer at the center of the instrument bench. Use a rubber dropper to place 2-3 drops of the liquid sample onto the prism surface, close the cover, and adjust the reflector until a clear boundary appears. Record the refractive index n_D at 25°C. Clean the prism with lens tissue moistened with ethanol.
- **LabAgent:** `validate_operation` (PASS) → `refractometer`.
- **Observation:** Measured refractive index: 1.38 (25°C).

Round 2

- **Experimenter (Reasoning):** $n_D \approx 1.38$ is common for lower aliphatic alcohols or carboxylic acids. Test pH first to rule out carboxylic acids.
- **Experimenter (Instruction):** Take wide-range pH paper, dip a clean glass rod into the liquid sample, and touch it to the center of the pH paper. Compare the color change with the standard chart.
- **LabAgent:** `validate_operation` (PASS) → `instrument_operation` (pH paper).

- **Observation:** The pH paper shows no significant color change upon contact with the sample.

Round 3
- **Experimenter (Reasoning):** Neutral liquid with $n_D \approx 1.38$ could be an alcohol, ketone, or ester. Use 2,4-DNPH to test for carbonyl groups.
- **Experimenter (Instruction):** In a fume hood, take a test tube, add $0.5\,\mathrm{mL}$ of sample. Add 3-4 drops of 2,4-dinitrophenylhydrazine solution, shake, and observe for yellow or orange-red precipitate.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (2,4-DNPH solution).
- **Observation:** No significant visible change. No yellow or orange-red precipitate is observed.

Round 4
- **Experimenter (Reasoning):** Aldehydes and ketones are excluded. $n_D \approx 1.38$ points to an aliphatic alcohol. Use Lucas reagent to identify the class of alcohol.
- **Experimenter (Instruction):** In a fume hood, take a test tube, add $0.5\,\mathrm{mL}$ of sample. Add $2\,\mathrm{mL}$ of Lucas reagent, shake, and record whether the mixture immediately turns cloudy and separates into layers, or turns cloudy within a few minutes at room temperature.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (Lucas reagent).
- **Observation:** After standing for 5-10 minutes, the mixture gradually turns cloudy and separates into layers.

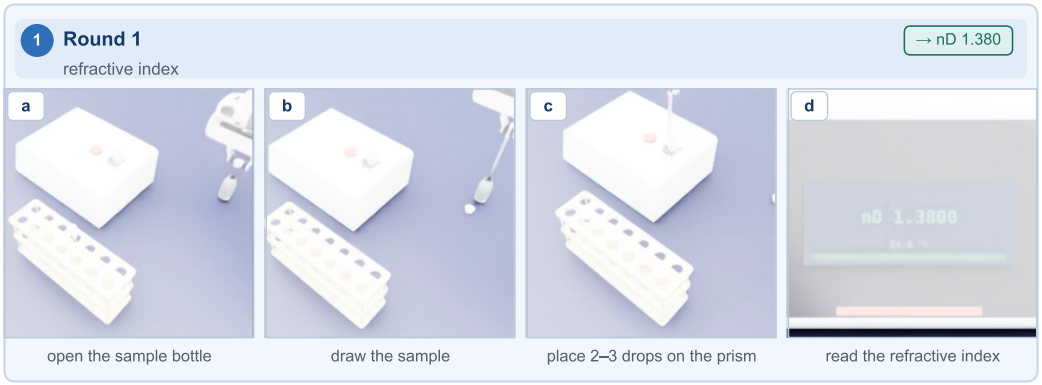
Round 5
- **Experimenter (Reasoning):** Secondary alcohol characteristics. $n_D \approx 1.38$ makes isopropanol a plausible candidate. Perform an iodoform reaction to verify it.
- **Experimenter (Instruction):** Take a test tube, add $0.5\,\mathrm{mL}$ of sample, add $1\,\mathrm{mL}$ of $2\,\mathrm{mol/L}$ sodium hydroxide solution, and add iodine-potassium iodide solution dropwise until a pale yellow color persists. Heat in a water bath at 50–$60^\circ\mathrm{C}$ and observe for a yellow precipitate with a characteristic odor.
- **LabAgent:** `validate_operation` (FAIL).
- **Observation:** Safety check failed. The requested procedure involves corrosive (NaOH) and toxic (iodine solution) reagents but lacks mandatory personal protective equipment (goggles and gloves).

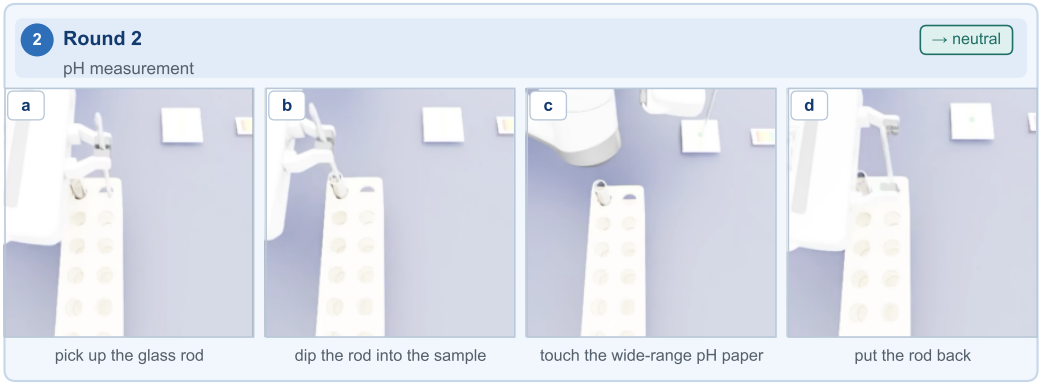
Round 6
- **Experimenter (Reasoning):** Wearing protective equipment to safely perform the iodoform reaction to verify isopropanol.
- **Experimenter (Instruction):** Wear goggles and nitrile gloves. Take a test tube, add $0.5\,\mathrm{mL}$ of sample. Add $1\,\mathrm{mL}$ of $2\,\mathrm{mol/L}$ NaOH, then add iodine-potassium iodide solution dropwise until a pale yellow color persists, shaking gently. Place the test tube in a 50–$60^\circ\mathrm{C}$ water bath for several minutes to observe for yellow crystalline precipitate (iodoform).
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (NaOH solution) → `mixture_reaction` (Iodine-Potassium Iodide solution).
- **Observation:** No significant change after adding NaOH. After adding iodine solution dropwise to a pale yellow color and heating in a water bath, a yellow precipitate forms.

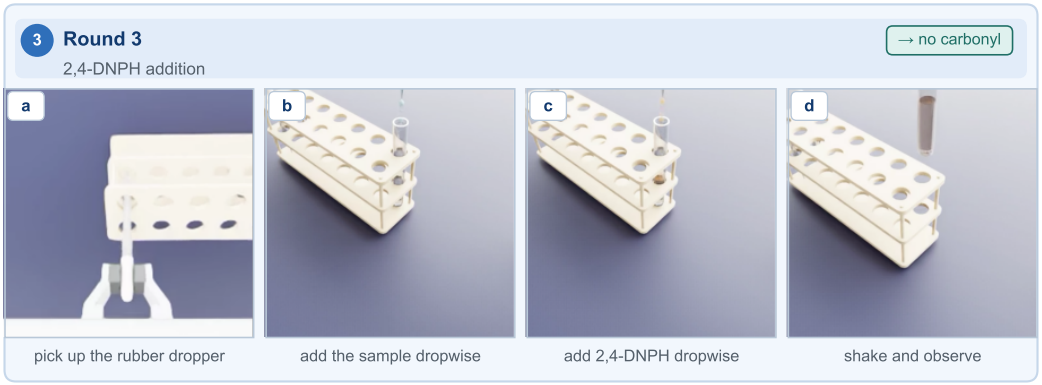
Round 7
- **Experimenter (Reasoning):** Sample is a colorless neutral liquid, $n_D \approx 1.38$. No precipitate with 2,4-DNPH excludes aldehydes and ketones. Turns cloudy with Lucas reagent in 5-10 minutes confirming a secondary alcohol. Positive iodoform reaction producing yellow precipitate proves the presence of a $CH_3-CH(OH)^-$ structure. Combining the refractive index 1.377-1.380 and the above results uniquely identifies the sample as isopropanol.
- **Experimenter (Instruction):** Identification: Isopropanol (2-Propanol).
- **LabAgent:** End.

- **Observation:** Experimenter conclusion: Isopropanol (2-Propanol). Rounds = 7, Reason = experimenter concluded.

Final Result: Isopropanol (2-Propanol) ✓ (Correct)







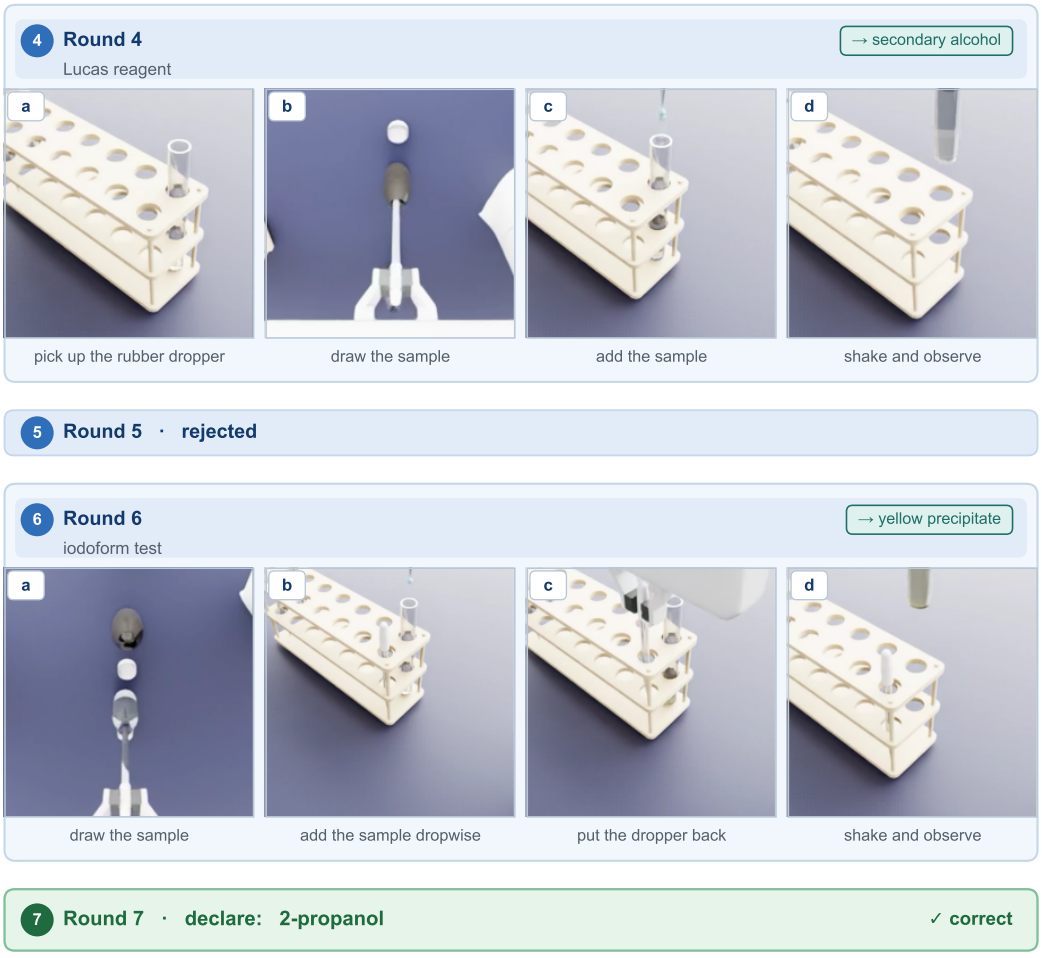


Figure A3: Case study of a successful liquid identification in ChemDeduce: the hidden target is 2-propanol, which the Experimenter declares correctly in the seventh round. Round 1 measures the refractive index ($n_D = 1.38$), Round 2 checks the pH (neutral), Round 3 applies 2,4-DNPH and excludes carbonyl compounds, Round 4 applies Lucas reagent and identifies a secondary alcohol, Round 5 is rejected by the safety checker, Round 6 is a positive iodoform test, and Round 7 declares the identity.

E.3 Failure Case

Dialogue History of Solid Identification

Target: Resorcinol (Solid)

Round 1

- **Experimenter (Reasoning):** Sample is a white needle-like crystal. First test its water solubility and solution pH to determine its category.
- **Experimenter (Instruction):** Take a test tube, add a small amount of sample, add 2 mL of distilled water, stir to observe dissolution, and test the pH of the solution with wide-range pH paper.
- **LabAgent:** `validate_operation` (PASS) → `mixture_reaction` (Distilled Water) → `instrument_operation` (pH paper).
- **Observation:** The white needle-like crystals gradually dissolve to form a colorless solution. The pH paper turns yellow to orange.

Round 2

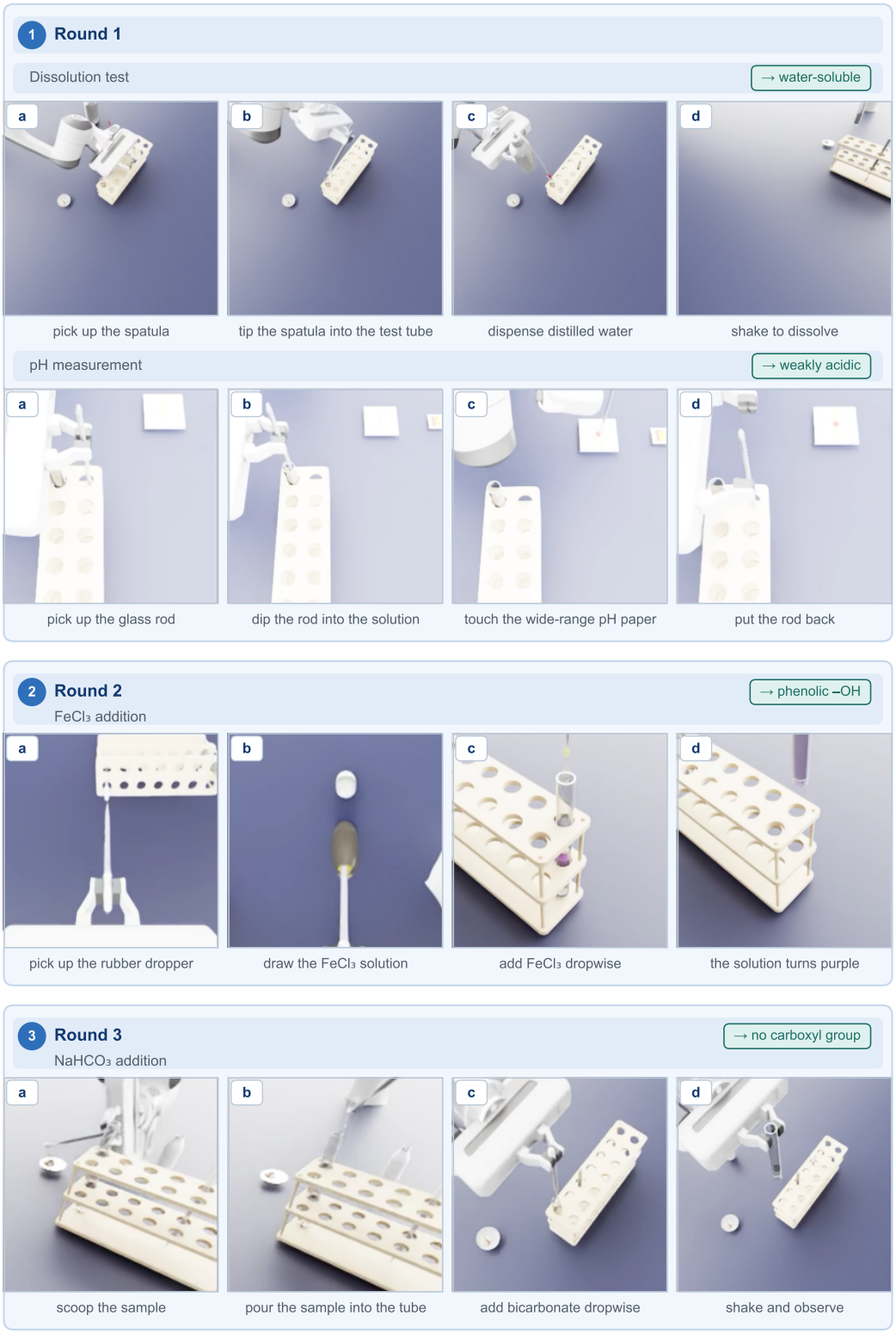
- **Experimenter (Reasoning):** The aqueous solution is weakly acidic. Add ferric chloride solution to test for phenolic hydroxyl or salicylic acid functional groups.
- **Experimenter (Instruction):** Take a test tube, add sample, add $2\,\mathrm{mL}$ distilled water, and dissolve. Add 2-3 drops of ferric chloride solution, shake, and observe the color change.
- **LabAgent:** `validate_operation` (PASS) $\rightarrow$ `mixture_reaction` (Ferric Chloride Solution).
- **Observation:** After adding ferric chloride solution and shaking, the solution turns blue-purple.

Round 3
- **Experimenter (Reasoning):** Blue-purple color with $FeCl_3$ indicates phenolic hydroxyl. Test with $NaHCO_3$ solution to check for a carboxyl group stronger than carbonic acid.
- **Experimenter (Instruction):** Take a test tube, add sample. Add $1\,\mathrm{mL}$ of sodium bicarbonate solution, shake, and observe if bubbles are produced.
- **LabAgent:** `validate_operation` (PASS) $\rightarrow$ `mixture_reaction` (Sodium Bicarbonate Solution).
- **Observation:** No bubbles are produced after adding sodium bicarbonate solution.

Round 4
- **Experimenter (Reasoning):** Carboxylic acid is excluded, confirming phenolic hydroxyl. Add concentrated bromine water to test for a tribromo-substituted white precipitate.
- **Experimenter (Instruction):** Take a test tube, add sample, add $2\,\mathrm{mL}$ distilled water, and dissolve. Wearing protective gloves in a fume hood, add bromine water dropwise, shake, and observe if a white precipitate forms.
- **LabAgent:** `validate_operation` (PASS) $\rightarrow$ `mixture_reaction` (Bromine Water).
- **Observation:** The reddish-brown color of bromine water gradually fades, and a white precipitate forms.

Round 5
- **Experimenter (Reasoning):** Sample is a white needle-like crystal, soluble in water, weakly acidic. Blue-purple color with $FeCl_3$ confirms phenolic hydroxyl. No bubbles with $NaHCO_3$ excludes carboxylic acid-containing phenolic acids (e.g., salicylic acid). Rapid reaction with bromine water to form a white precipitate (2,4,6-tribromophenol) highly matches the typical reaction of a monofunctional phenol. Uniquely locked as phenol.
- **Experimenter (Instruction):** Identification: Phenol.
- **LabAgent:** End.
- **Observation:** Experimenter conclusion: Phenol. Rounds = 5, Reason = experimenter concluded.

Final Result: Phenol $\times$ (Incorrect)


1 Round 1
Dissolution test
→ water-soluble
a
b
c
d
pick up the spatula
tip the spatula into the test tube
dispense distilled water
shake to dissolve
pH measurement
→ weakly acidic
a
b
c
d
pick up the glass rod
dip the rod into the solution
touch the wide-range pH paper
put the rod back
2 Round 2
FeCl₃ addition
→ phenolic –OH
a
b
c
d
pick up the rubber dropper
draw the FeCl₃ solution
add FeCl₃ dropwise
the solution turns purple
3 Round 3
NaHCO₃ addition
→ no carboxyl group
a
b
c
d
scoop the sample
pour the sample into the tube
add bicarbonate dropwise
shake and observe

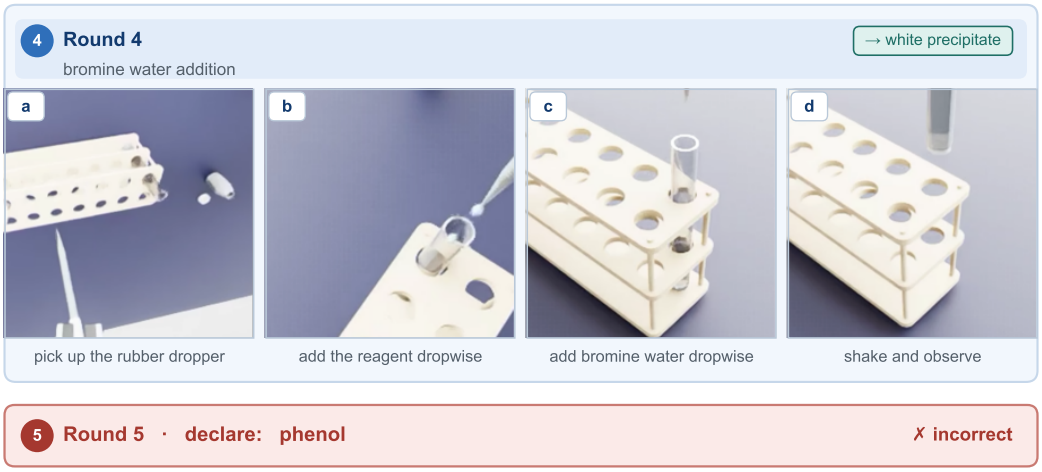


Figure A4: Case study of a failed solid identification in ChemDeduce: the hidden target is resorcinol, but the Experimenter declares phenol in the fifth round and is incorrect. Round 1 covers solubility and pH, Round 2 applies $FeCl_3$, which gives a purple color, Round 3 applies sodium bicarbonate and finds no carboxyl group, Round 4 applies bromine water, which gives a white precipitate, and Round 5 declares the identity. Its error is one of reasoning rather than execution: it does not consider that the same phenomenon can be produced by different substances, and resorcinol gives the same purple color with $FeCl_3$ and the same white precipitate with bromine water.

F Details for Trajectory Evaluation and Failure Annotation

F.1 TDS Dimensions

The Trajectory Deduction Score (TDS) evaluates the quality of the experimental reasoning trajectory beyond final identification accuracy. It consists of five dimensions, with a total score of 100:

- **Candidate and Hypothesis Management (CHM, 15 points):** The extent to which the Experimenter maintains a meaningful set of candidate substances, introduces hypotheses based on observed evidence, removes candidates when observations contradict them, and updates the candidate space throughout the interaction.
- **Diagnostic Experiment Design (DED, 20 points):** The informativeness and discriminative value of the selected experiments, evaluating whether the Experimenter chooses tests that target unresolved uncertainty, distinguish among plausible candidates, and avoid redundant or low-value operations.
- **Observation Interpretation Fidelity (OIF, 15 points):** The faithfulness and chemical correctness of the agent's interpretation of returned observations, ensuring that conclusions are traceable to actual experimental feedback and do not rely on fabricated, exaggerated, or over-interpreted phenomena.
- **Evidential Sufficiency and Uniqueness (ESU, 30 points):** The degree to which the accumulated observations uniquely support the final declared substance and rule out major competing candidates, rather than merely supporting a broad chemical class or an underdetermined set of possibilities.
- **Evidence-Gated Decision Efficiency (EGDE, 20 points):** The extent to which the agent declares the final identity at an appropriate time, avoiding both premature identification before sufficient evidence is available and unnecessary additional testing after the evidence has become sufficient.

F.2 Detailed Scoring Rubric

Each TDS dimension is scored on a six-level anchored ladder. A valid score must fall on the corresponding ladder and be supported by evidence from the recorded trajectory. When two adjacent descriptors both apply, the scoring judge is instructed to assign the lower score. Table A3 summarizes the detailed rubric.

Table A3: Rubric for Trajectory Deduction Score Evaluation

Dimension (Max)	**Score Criteria**
Candidate and Hypothesis Management (15)	**15**: Every candidate is traceable to an observation that introduced it; each relevant competitor is either eliminated by a named discriminating observation or explicitly retained as unresolved; eliminations visibly redirect subsequent tests. **12**: The candidate set is built from observations and does shrink, but eliminations are recorded only in aggregate or only when the conclusion is already in view. **9**: Candidates are motivated by observations, but the set mostly grows; eliminations are asserted rather than forced by diagnostic tests. **6**: Candidates are weakly connected to observations, or a candidate that should have been excluded remains active. **3**: The reasoning only restates surface facts without forming a meaningful chemical candidate space. **0**: No candidate analysis is present, or an excluded candidate is kept alive or reintroduced after being ruled out.
Diagnostic Experiment Design (20)	**20**: The design is highly diagnostic: a small number of steps efficiently narrows the candidate space, each experiment targets live candidates, and no redundant tests are performed. **16**: The design is targeted and works through key properties needed to separate plausible candidates. **12**: Most experiments eliminate some candidates, but the ordering or targeting is only moderately effective. **8**: A few experiments are useful, but the overall strategy is loose and does not substantially shrink the candidate space. **4**: Experiments are mostly low-information, repeated, or lack a stated discriminating purpose. **0**: Experiments are invalid, unexecutable, or unrelated to chemical identification.
Observation Interpretation Fidelity (15)	**15**: Every key interpretation is accurate, restrained, and traceable to an observation; negative results and residual uncertainty are handled correctly. **12**: Interpretations are faithful and chemically sound, and key conclusions trace back to observations. **9**: Interpretations are mostly consistent with the observations, with only mild overreach. **6**: The reasoning contains several unsupported claims or a clear chemical interpretation error. **3**: The trajectory frequently refers to phenomena that were not observed, or seriously over-interprets qualitative observations. **0**: Most interpretations contradict the observations, or the reasoning rests on fabricated phenomena.
Evidential Sufficiency and Uniqueness (30)	**30**: The accumulated observations uniquely support the declared substance and explicitly rule out the main competing candidates. **24**: The evidence essentially supports the declared substance, with only minor gaps or slight residual uncertainty. **18**: The evidence supports the declared substance fairly strongly, but an important competing candidate is not excluded. **12**: The evidence supports the declared substance but also supports several reasonable unresolved candidates. **6**: The evidence supports only a broad chemical class rather than a specific substance. **0**: The conclusion contradicts key observations or rests on no experimental evidence.
Evidence-Gated Decision Efficiency (20)	**20**: The model declares one round after t^*, with no unnecessary testing after the evidence becomes sufficient. **16**: The model declares two rounds after t^*. **12**: The model declares three or more rounds after t^*. **8**: The model declares three or more rounds after t^*, and most extra rounds return no new useful observation; a wrong final answer normally receives this score or lower. **4**: No sufficient-evidence round t^* exists, but the model still names a specific identity. **0**: There is no valid final declaration, the answer is a pure guess, or the conclusion contradicts the observations.

Here, t^* denotes the first Test round whose returned observation makes the accumulated evidence sufficient to support a specific identity, and t_{decl} denotes the round in which the model declares the final answer. Because the observation from round t^* is received only after the Experimenter has issued the Test action for that round, the earliest evidence-grounded Identify action can occur on the following Experimenter turn. Thus, $t_{\mathrm{decl}} = t^* + 1$ is the earliest possible supported declaration and receives full EGDE credit. EGDE is therefore not a simple speed reward: a fast unsupported guess receives little or no credit, whereas an efficient evidence-gated decision receives a high score. Unless otherwise specified, TDS is computed for the default $T_{\max} = 10$ evaluation setting.

F.3 Failure-Type Annotation

Failure-type analysis is applied only to episodes that end with a valid but incorrect final declaration. Episodes without a valid final declaration are excluded from this taxonomy and are instead counted by BER. Each eligible episode is assigned exactly one failure category using the full recorded trajectory. The categories are defined as follows:

- **Misread:** The decisive inference contradicts or materially misstates an observation returned by LabAgent.
- **Premature:** The model declares a specific identity before the accumulated evidence discriminates it from plausible alternatives, without a decisive observation-misreading error.
- **Near-neighbor:** The trajectory remains within the appropriate local candidate family but selects a chemically close alternative that shares the observed tests.
- **Off-track:** The search trajectory or final declaration moves to a candidate that is poorly supported by the accumulated observations and is outside the plausible local neighborhood.
- **Other:** The error is not captured by any of the preceding categories.

Each episode was independently labeled by two annotators with chemistry backgrounds. Disagreements were resolved by discussion.

G Data and Simulation Quality Verification

To verify the quality of the simulated laboratory, we conducted an additional human review of LabAgent-generated trajectories. We sampled 40 target substances from the benchmark, including 20 solids and 20 liquids. For all sampled targets, the Experimenter was Gemini-3.7-Flash and the LabAgent LLM controller was DeepSeek-V4-Pro. The complete interaction trajectories, including the Experimenter's experimental requests, LabAgent's executed procedures, rendered experimental videos, and returned textual observations, were provided to two chemistry professors for manual evaluation.

The review focused on whether the LabAgent side of the benchmark produced chemically plausible and evaluation-valid experimental feedback. Specifically, the reviewers examined three aspects:

1. **Request–Video Consistency:** Whether the rendered experimental video matches the Experimenter's requested operation, including the selected apparatus, reagent, sample handling, and intended experimental procedure.
2. **Simulation Outcome Correctness:** Whether the visual outcome in the rendered video is consistent with the expected chemical behavior under the requested experimental condition, such as color change, precipitation, gas evolution, dissolution, heating response, or instrument operation.
3. **Macroscopic Observation Accuracy:** Whether the textual observation returned by LabAgent faithfully describes the macroscopic phenomenon shown in the video without adding unsupported interpretation, hidden identity information, diagnostic hints, or new chemical formulas beyond reagent names already specified in the Experimenter's request.

Each reviewer independently inspected the sampled trajectories and judged whether the LabAgent output was acceptable for benchmark evaluation. A trajectory was considered valid only when the experimental procedure, rendered phenomenon, and textual observation were all judged to be consistent with the requested operation and chemically plausible. Disagreements, if any, were resolved through discussion between the reviewers.

Overall, the reviewed cases passed the manual quality check. The experts found that LabAgent's rendered videos were aligned with the requested experimental procedures, the simulated outcomes

were chemically reasonable for the sampled targets, and the returned textual observations accurately reflected observable macroscopic phenomena. This review supports the reliability of LabAgent as a stateful simulated laboratory for evaluating closed-loop chemical identification, while recognizing that the simulator remains an abstraction of real wet-lab experimentation.

H Model Description

We evaluate eight frontier LLMs as Experimenter agents in ChemDeduce. Each Experimenter model is evaluated on all 200 target substances with three independent repetitions, resulting in 600 episodes per model and $4{,}800$ episodes in total. All Experimenter models are accessed through their official APIs. Unless otherwise specified by the provider, Experimenter inference uses temperature 0.3 and a maximum output length of $8{,}192$ tokens. No model-specific prompt or decoding hyperparameter is tuned on the evaluation targets. The same task instructions, laboratory resources, action format, and interaction budget are used for all Experimenter models.

In addition to the Experimenter models, ChemDeduce uses LLMs on the LabAgent side for interpreting experimental requests and coordinating tool calls. The default LabAgent controller is DeepSeek-V4-Pro. In the LabAgent controller sensitivity analysis, we additionally use Qwen3.8-Flash and StepFun-3.7-Flash as alternative LabAgent controllers. For process-level evaluation, the offline TDS scoring judge is DeepSeek-V4.1-Flash. The TDS judge is separate from the LabAgent controller and is invoked only after an episode has finished. Detailed model specifications are summarized in Table A4.

Table A4: Summary of LLMs used in ChemDeduce. Experimenter models are evaluated as autonomous chemical-identification agents. LabAgent controllers are used inside the simulated laboratory. The TDS scoring judge is used only for post-hoc trajectory evaluation.

Model Name	Creator	Role	Access	URL
GPT-5.6-Sol	OpenAI	Experimenter	Official API	`https://openai.com`
Qwen3.8-Max	Alibaba	Experimenter	Official API	`https://qwenlm.github.io`
Gemini-3.7-Flash	Google	Experimenter	Official API	`https://gemini.google.com`
GLM-5.3	Zhipu AI	Experimenter	Official API	`https://www.zhipuai.cn`
GLM-5.2	Zhipu AI	Experimenter	Official API	`https://www.zhipuai.cn`
MiniMax-M3	MiniMax	Experimenter	Official API	`https://www.minimax.io`
Grok-4.3	xAI	Experimenter	Official API	`https://x.ai`
Kimi-K3	Moonshot AI	Experimenter	Official API	`https://www.moonshot.cn`
DeepSeek-V4-Pro	DeepSeek	Default LabAgent LLM controller	Official API	`https://www.deepseek.com`
Qwen3.8-Flash	Alibaba	Alternative LabAgent LLM controller	Official API	`https://qwenlm.github.io`
StepFun-3.7-Flash	StepFun	Alternative LabAgent LLM controller	Official API	`https://www.stepfun.com`
DeepSeek-V4.1-Flash	DeepSeek	TDS scoring judge	Official API	`https://www.deepseek.com`

Inference configuration. Experimenter models use temperature 0.3 and a maximum output length of $8{,}192$ tokens, except when a provider does not expose the corresponding parameter. The default LabAgent controller, DeepSeek-V4-Pro, uses temperature 0.2 and a maximum output length of $32{,}800$ tokens. Other LabAgent controllers are used only in the controller sensitivity analysis. The offline TDS scoring judge, DeepSeek-V4.1-Flash, is used after the interaction has ended and does not affect the Experimenter's behavior during an episode.

Context isolation. The Experimenter model, LabAgent controller, and TDS scoring judge operate in separate contexts. The Experimenter receives only public laboratory information, the initial physical state and appearance of the unknown sample, and LabAgent's returned observations or rejection messages. The LabAgent controller coordinates tool calls and state-consistent simulation under hidden target-conditioned records. The TDS scoring judge receives completed trajectories and evaluation-only reference information for post-hoc scoring, but its outputs are never returned to the Experimenter.

Repetition protocol. Each model–target pair is evaluated in three independent repetitions. At the start of each repetition, the Experimenter context, laboratory state, sample state, apparatus state, and interaction history are reset. The target identity remains fixed for the same target across repetitions, but the model may select different experiments due to sampling variability. Reported main results are computed over the three repetitions, with standard errors reported where available.

I DETAILED PROMPTS

Experimenter System Prompt

You are a chemical analysis laboratory technician, identifying an unknown pure substance through step-by-step experiments. You know nothing at all about the identity of the sample and may reason only from instrument readings and reaction phenomena. Output exactly one operation instruction per turn.

Available operations

1. Use the polarimeter: optical-activity measurement of chiral substances.
2. Use the refractometer: refractive-index measurement of liquid samples.
3. Mixing reaction: mix the sample with a reagent and observe the resulting phenomenon.
4. Laboratory operation: heating, flame test, combustion, pH paper, magnetism, etc.
5. Use the conductivity meter: conductivity of the sample's aqueous solution.

Each operation must be issued as a complete procedure; the system identifies the operation type automatically.

Core rules

1. Identify in as few rounds as possible. Every round should yield a high-information observation; avoid redundant verification and uninformative probing.
2. One operation per round. A round allows a "compound procedure," which may consist of multiple primitive actions.
3. Do not guess the substance name. The final identification is the only exception.
4. Reason from observations. Do not assume without grounds, and do not get stuck on an unexpected result.

Output format

Every experimental round must use exactly this two-part template:
[Minimal reasoning] at most two sentences, stating only why this experiment was chosen.
[Next operation instruction] one standard-format instruction.
When the evidence is sufficient to determine the sample uniquely:
Identification: substance name.

Hard constraints on reasoning

- Forbidden: multi-step deliberation, reaction mechanisms, molecular-structure analysis, by-product speculation, and background exposition.
- Forbidden: restating the accumulated observations wholesale; extract only the feature that discriminates for the current decision.
- Forbidden: presenting a candidate identity as final before sufficient evidence is obtained; tentative hypotheses may be stated only when they motivate the next discriminating experiment.

Prohibited behavior

- No unsupported final conclusion. Candidate narrowing is allowed only when it is explicitly grounded in observations and followed by a discriminating test if uncertainty remains.
- No multiple parallel experimental plans.
- No laboratory-technique discussion, or background material.
- No explanatory build-ups or redundant modifiers in the reasoning.
- No line-by-line restatement of historical observations.
- When the context grows long, compress reasoning to one sentence and prioritize the completeness of the instruction.
- No blank controls and no simultaneous operation on multiple test tubes.

Requirements on operation instructions
Every instruction must be a complete procedure that can be executed step by step. It should contain, in execution order:

1. which equipment and reagents to take, and from where;
2. how to open the container, with what, and how much to measure out;
3. the operations themselves, in the correct order;
4. finishing steps, such as resealing bottles and returning equipment.

LabAgent Controller System Prompt

You are a laboratory-bench simulator. Your role is the display of an instrument: you output experimental observations and nothing else; you perform no analysis.
You are connected to the following instruments and benches:

- Operation-compliance check: before executing any experimental operation, you must call `validate_operation`. If it returns failure, return the issues to the Experimenter and do not call any other tool.
- Polarimeter: measures optical rotation of a sample.
- Refractometer: measures refractive index of a liquid sample.
- Mixing-reaction bench: mixes the sample with a reagent and observes the phenomenon.
- Instrument-operation bench: heating, flame test, pH paper, and similar operations.
- Conductivity meter: measures conductivity of the sample's aqueous solution.

Workflow

1. On receiving an instruction, first call `validate_operation`, passing the Experimenter's full instruction text.
2. If `validate_operation` returns failure, turn the issues into plain-language feedback for the Experimenter and end this round.
3. If it returns pass, call the instrument or reaction tool requested by the instruction.
4. Render the tool's returned data as concise natural language. Report the phenomenon only, with no reasoning attached.
5. You may obtain data only by calling tools; you may not invent it from your own knowledge.

The two formats a tool may return
Format A: direct text. The tool's `text` field can be output as it stands. Transcribe it verbatim, adding and removing nothing.
Format B: structured data. The tool's dictionary contains `needs_llm_translation: true` and a `translation_hint` field. The `text` field is raw chemical JSON. You must read the hint, extract the key information, and express it as a macroscopic phenomenon. Do not output raw JSON, structured field names, target identities, product identities, diagnostic substance names, chemical formulas, or reaction equations, except for reagent names already provided in the Experimenter's instruction when needed for safety or instrument messages.
Core rule
You are an instrument, not a chemist. The following must never appear in your output:

- chemical reasoning or analysis;
- evaluation of or response to the Experimenter's reasoning;
- chemical explanations;
- target identities, product identities, diagnostic substance names, chemical formulas, or reaction equations, except for reagent names already provided in the Experimenter's instruction when needed for safety or instrument messages;

- suggestions for the next experiment;
- hints about the matching process.

You may output only:

- instrument readings;
- macroscopic phenomena visible to the eye;
- instrument messages or error messages.

If the Experimenter asks for a mixing reaction without naming a reagent, ask them to specify the reagent.

TDS Scoring Judge Prompt

You are a senior analytical chemist with twenty years of experience in qualitative analysis and undergraduate laboratory instruction. You are evaluating the reasoning trace of an autonomous agent that was asked to identify an unknown substance through simulated experiments. Your assessment will be entered into a benchmark report on closed-loop experimental reasoning and will be compared across eight frontier models.
You are scoring the QUALITY OF THE REASONING PROCESS. Two things follow from that, and they pull in opposite directions — hold both at once:

- A correct identity reached by guessing, by prior familiarity, or on evidence that cannot distinguish it from its neighbors is a WEAK process and must score low.
- Whether the final identity is right does NOT by itself set the score. It gates exactly two dimensions, ESU and EGDE, through the caps in Section 6. CHM, DED and OIF must be scored on the reasoning alone, with no regard to whether the answer turned out to be correct. A model that reasoned well and concluded wrongly still earns those three dimensions.

Score only what the trace actually records. Do not assume the agent "probably did it but did not write it down". However, an operation choice and its timing are part of the trace and may be used as evidence of which uncertainty the agent was addressing. A blocked operation is a normal, expected event, not an error by itself.
EFFICIENCY — read this before you start. Your score depends only on the final JSON object, never on how long you deliberate. Reach the `===JSON===` block directly and quickly: do not re-read the trace repeatedly, do not weigh the same evidence more than once. A judgment that never emits the object is a failed judgment, no matter how good the reasoning inside it was.

SECTION 1 — WHAT THE AGENT STARTED WITH
Sample physical state: {`STATE`}
Initial appearance (given to the agent for free): {`APPEARANCE`}
Instruments available in this laboratory:

- polarimeter (optical rotation)
- refractometer (refractive index; liquids only)
- conductivity meter (conductivity, mS/cm)
- mixing bench (reagent reactions, tracked in a virtual test tube)
- operation bench (heating, flame test, pH, magnetism, solubility)

Infrared spectroscopy and any precision melting/boiling-point instrument are NOT available. A manual melting-range or boiling-point observation counts as a valid but coarse measurement.

SECTION 2 — THE SESSION TRACE
Each round lists: the agent's stated reasoning, the instruction it issued, the safety-and-compliance gate verdict (PASS / FAIL with the reason), the operation actually executed, and the observation returned by the bench.
A round may contain SEVERAL "operation:" lines, when the agent bundled multiple steps into one instruction. Under this task prompt, combining several independent operations in a

single round is not allowed and should not be rewarded as efficient design. Batching a step whose design depends on the result of an earlier step in the SAME round is also a design defect.
{TRACE}

SECTION 3 — FACTS COMPUTED FROM THE RECORD (authoritative)
These values were computed from the episode record. They are not estimates, and you must not recompute or override them: several caps in Section 6 are written against exactly these numbers.
Final-answer verdict: {VERDICT}
How the episode spent its rounds:
{TIMING}
Remember how the verdict may be used: it gates ESU and EGDE only. Score CHM, DED and OIF as if you did not know it.

SECTION 4 — REFERENCE MATERIAL (never shown to the agent)
Hidden target substance: {SUBSTANCE}
Its canonical, source-grounded identification route under exactly these instruments:
{ROUTE}
Near-neighbor substances that share one or more of the target's tests (these are the hard cases, where a non-discriminating test still returns a plausible reading):
{CLUSTER}
Use this section to judge how much discrimination was available in this laboratory and which test would have settled the question. It is not itself worth points: a trace is not better because it names more of these substances, and a test is only diagnostic if it separates the candidates the agent was actually still considering.

SECTION 5 — PROCEDURE
Complete Steps 1 to 3 BEFORE assigning any score in Step 4. Writing the evidence first prevents the scores from being reverse-engineered from the known answer.

STEP 1 — CANDIDATE AND EVIDENCE INVENTORY. List, in order, the candidate substances, chemical classes, or discriminating possibilities that are visible in the trace. Visibility may come from the agent's stated reasoning, from the operation it chose after a given observation, or from the final exclusion reasoning. Do not require an explicit candidate list when the task prompt forbids early candidate-set discussion.
For each visible candidate or discriminating possibility, name the observation that introduced it, the observation that eliminated it or left it open, the round where that happened, and — for an elimination — the property of that observation that made it discriminating. Mark any asserted candidate unsupported by observation, and any possibility that the trace keeps using after an observation had already excluded it.

STEP 2 — EVIDENCE SUFFICIENCY OVER TIME. Answer in two or three sentences: at what round did the observations the agent had collected FIRST become sufficient to support a unique or near-unique identity? Call that round t^*.
Judge this on the observations available at that moment, and NOT on how the episode turned out. In particular, do not let the declaration pull t^* backwards:

- if it was the LATER rounds that made the case decisive, t^* is that later round;
- if the agent never reached sufficient evidence, t^* is "none" — even though it declared an identity. Declaring is not evidence that the evidence sufficed, and a specific name announced on weak observations does not make those observations strong.

If a decisive discriminating test was available but never performed, name it.

STEP 3 — DECLARATION TIMING. State `t_decl`, the round in which the agent declared its final identity (or "none"). Compare it with t^*: earlier means the identity was announced before the evidence justified it, much later means the agent kept working after it already knew enough. Note whether any rounds between t^* and `t_decl` produced anything useful.

STEP 4 — SCORE THE FIVE DIMENSIONS. Every score must be an integer, must fall on one of the anchored descriptors below, and must be justified by a verbatim quote from the

trace. When two descriptors both seem plausible, take the LOWER one. The caps in Section 6 are hard.

D1. Candidate and Hypothesis Management (CHM, 0–15)
Did the agent manage the live possibilities in a way that did WORK — using observations to choose discriminating tests and to stop pursuing possibilities that those observations had made untenable?
Judge candidate management by visible effects in the trace: stated reasoning, operation choice, response to observations, and final exclusion logic. Do not require explicit candidate lists, explicit narrowed-range statements, or extended class/functionality discussion when the task prompt forbids them. A terse trace can earn full credit if the sequence of experiments shows that observations redirected the search. Conversely, naming many plausible substances is not management unless those names guide or are pruned by diagnostic observations.

15 observations visibly guide a live possibility space: each major alternative the available tests bear on is either excluded by a discriminating observation, carried as still unresolved, or addressed by a later test; this management may be implicit in experiment choice and final exclusion logic rather than explicitly narrated

12 observations guide the search and the live space shrinks, but some eliminations are only aggregate or implicit, or the pruning becomes clear only near the conclusion

9 the agent's experiments are broadly consistent with observation-based possibilities, but the trace mostly accumulates evidence rather than clearly pruning alternatives

6 the trace shows weak or loose connection between observations and the possibilities being tested, or a later observation should have changed the search but did not

3 the trace mostly restates surface facts or performs generic tests, with little visible management of chemical possibilities

0 no visible candidate or possibility management; jumps straight to a single identity; or continues to rely on a possibility after an observation had already excluded it

D2. Diagnostic Experiment Design (DED, 0–20)
Were the chosen experiments informative and aimed at the current uncertainty?

20 highly diagnostic: few steps efficiently narrow the space, each aimed at the live candidates, nothing redundant

16 targeted design working through the key properties to separate candidates

12 most experiments eliminate something, but the ordering or the targeting is only fair

8 a few useful experiments, but the strategy is loose and does not really shrink the candidate space

4 mostly low-information, repeated, or with no stated discriminating purpose

0 experiments are invalid, unexecutable, or unrelated to the identification

Notes:

- a round that bundles a dependent step — an operation whose choice could only be justified by the result of another operation in the SAME round, not yet observed when the instruction was written — is a design defect. Do NOT apply this to independent measurements batched together (e.g. pH and conductivity on the same sample), which is legitimate practice.
- a non-discriminating test repeated after it already returned its answer counts against this dimension.
- reducing the step count is itself part of good design here: reaching the same discrimination in fewer rounds is better, not merely equivalent.

D3. Observation Interpretation Fidelity (OIF, 0–15)
Did the agent interpret the bench's returns faithfully and chemically correctly?

15 every key interpretation is accurate, restrained and traceable; negative results and residual uncertainty are handled correctly

12 faithful and chemically sound; key conclusions trace back to observations

9 mostly consistent with the observations, with some mild over-reach or loose wording

6 some interpretations correct, but several unsupported claims or a clear chemical error

3 frequent reference to phenomena that were not observed, wrong attributions, or serious over-interpretation

0 most interpretations contradict the observations, or the core reasoning rests on fabricated phenomena

Caps:

- the agent repeatedly cites observations the bench never returned → OIF at most 6
- a "no visible change" or "cannot determine" return is used as a confirmed positive or a confirmed exclusion → OIF at most 9
- a qualitative result is upgraded to a definite structure or a definite number with nothing in the trace to support it → OIF at most 9

D4. Evidential Sufficiency and Uniqueness (ESU, 0–30)
On the observations this episode actually produced, is the declared identity uniquely supported, and are the main competing candidates ruled out? THIS IS NOT ACCURACY. A correct answer on non-unique evidence scores LOW here.

30 the accumulated observations uniquely support the declared substance and explicitly rule out the main competing candidates

24 the evidence essentially supports the declared substance; only minor gaps or slight residual uncertainty

18 the evidence supports it fairly strongly, but an important competing candidate was never excluded

12 the evidence supports the declared substance, but equally supports several reasonable candidates that remain unresolved

6 the evidence supports only a very broad class ("a carboxylic acid", "an aldehyde", "a sodium salt"), not a specific substance

0 the conclusion contradicts the key observations, or rests on no evidence

Judge the evidence itself, not the agent's confidence in it, and not whether the declared name happens to be right. Reliance on prior familiarity or on which substance is common, rather than on this episode's observations, is a reason to stay at a low level. The caps in Section 6 are binding on this dimension.

D5. Evidence-Gated Decision Efficiency (EGDE, 0–20)
Once the evidence was sufficient, did the agent conclude — or did it keep going?
This is NOT a speed reward. A fast guess is worth nothing: a short episode spent on a guess is not better than a long one spent gathering evidence. The whole dimension is the comparison between the two rounds you established in Steps 2 and 3 — t^*, when the evidence first sufficed, and `t_decl`, when the agent said what the substance was. In this benchmark an episode ends when the identity is stated, so `t_decl` is always the last round; what varies is how much work preceded it that t^* had already made unnecessary.

20 concluded one round after t^*, with nothing wasted after t^*

16 concluded two rounds after t^*

12 concluded three or more rounds after t^*

8 concluded three or more rounds after t^*, and most of those extra rounds returned no new observation; a wrong final answer normally sits at this level or below

4 t^* never existed — a specific identity was named on evidence that never became sufficient to support it

0 no valid final declaration; or the conclusion is a pure guess with no evidence; or it plainly contradicts the observations

Think of it as: the agent had {`BUDGET`} rounds. The moment it had enough to name the substance, every further round is a round it did not need. Credit here is for recognizing sufficiency, and it is only available to the extent D4 says the evidence really was sufficient.

SECTION 6 — GLOBAL CAPS

Apply these on top of the per-dimension caps; the tightest cap wins. Where a cap names a condition you can read in the record, take the verdict and the timing in Section 3 as settled fact. Where it names a judgment, make it yourself.

ESU

1. No valid final declaration → ESU at most 6, and usually 0.
2. Declared identity is wrong (Section 3 verdict) → ESU at most 12. If the conclusion also contradicts a key observation → ESU must be 0 or 6.
3. The evidence supports only a broad class rather than a specific substance → ESU at most 12, even when the declared name is correct.
4. A major competing candidate remained unexcluded → ESU at most 18.
5. The agent repeatedly invokes observations the bench never returned → ESU at most 12, and OIF at most 6.

EGDE

1. No valid final declaration → EGDE must be 0.
2. Declared identity is wrong (Section 3 verdict) → EGDE at most 8. If the conclusion also contradicts a key observation → EGDE must be 0 or 4.
3. EGDE may never exceed what the ESU you just awarded can pay for:

ESU 0 or 6 →	EGDE at most 4
ESU 12 →	EGDE at most 8
ESU 18 →	EGDE at most 12
ESU 24 or 30 →	EGDE may be 16 or 20

   A model does not get decision-efficiency credit for concluding quickly on evidence that does not support the conclusion.
4. The declaration came only in the final round ({`BUDGET`}) → EGDE at most 16, even if the answer is correct and the evidence sufficient; at most 8 if the evidence was not sufficient.
5. Note that EGDE carries no cap for the share of rejected or repeated steps. That share is a separate matter with its own place in the benchmark, and it does not enter this dimension: credit here is for concluding when the evidence allowed it, not for the gate's opinion of the work. The timing bands — how many rounds late or early the declaration came — are the descriptors of D5; apply them there rather than restating them here as a cap.

List every cap you actually applied in "caps_applied", using these names: `NO_FINAL_DECLARATION`, `INCORRECT_ANSWER`, `CONTRADICTS_OBSERVATION`, `CLASS_ONLY`, `COMPETITORS_UNEXCLUDED`, `HALLUCINATION`, `DECLARED_AT_BUDGET`, `ESU_GATED_EGDE`. If none applied, use `["NONE"]`. Do not list a cap you did not apply, and do not apply one you did not list.

SECTION 7 — HOW TO REPLY

Reply in exactly TWO parts.

PART 1 — ANALYSIS. Plain prose, at most about 350 words. Work through Steps 1 to 4 above: the visible candidate/evidence or possibility/evidence inventory, the evidence-sufficiency answer with t^*, the declaration timing, then each of the five scores with the trace quote or operation choice you are basing it on. This part is read by human reviewers and is part of the record. Be concise — a long deliberation that never reaches the JSON is a failed judgment.

PART 2 — MACHINE-READABLE RESULT. After the analysis, write the line

`===JSON===`

on a line by itself, then the JSON object below, and NOTHING after it. The JSON object must be the last thing in your reply. No markdown fences, no comments inside the JSON. Every string must be valid JSON; keep each justification under 200 characters.

```
{
  "scores": {
    "candidate_and_hypothesis_management":
      <int 0-15>,
    "diagnostic_experiment_design":
```

```
      <int 0-20>,
    "observation_interpretation_fidelity":
      <int 0-15>,
    "evidential_sufficiency_uniqueness":
      <int 0-30>,
    "evidence_gated_decision_efficiency":
      <int 0-20>
  },
  "tds": <int 0-100, the sum of the
          five scores>,
  "justifications": {
    "candidate_and_hypothesis_management":
      "<one sentence>",
    "diagnostic_experiment_design":
      "<one sentence>",
    "observation_interpretation_fidelity":
      "<one sentence>",
    "evidential_sufficiency_uniqueness":
      "<one sentence>",
    "evidence_gated_decision_efficiency":
      "<one sentence>"
  },
  "t_star": <int round 1-10 at which the
             evidence first became
             sufficient, or null>,
  "t_decl": <int round 1-10 in which the
             identity was declared, or
             null>,
  "caps_applied": ["<NONE, or one or
                   more of the cap names
                   listed in Section 6>"],
  "one_line_verdict": "<one sentence,
                       under 30 words>"
}
```