ELEMENTARY: SHERLOCKED BY CAUSALITY

From Clues to Causes in Interactive LLM Agents

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive LLM agents may succeed on the current task without recovering causal structure that remains useful when conditions or objectives change. We introduce **ELEMENTARY**, a generator-based escape-room benchmark that separates evidence sufficiency from agent reasoning failure by controlling how solution-relevant evidence becomes available within a shared interactive environment. The benchmark defines four evidence-conditioned reasoning regimes: **Association** exposes explicit relations; **Observation** provides sufficient passive evidence; **Intervention** requires active evidence acquisition; and **Counterfactual** requires reasoning from factual traces through historical revision and replay. Across twelve generator families, every generated question is validated so that the permitted evidence determines the required task-level solution. We evaluate seven proprietary and open-weight LLMs on 120 validated questions. No model is consistently strong across evidence conditions, and the highest average success rate is only 44.2%. More importantly, success on the current task can hide a failure to recover reusable causal structure. In I3 Causal Propagation, all $70/70$ model–question episodes reach the first control target, but only $1/70$ reaches the second target when successful control requires propagation-aware action ordering. Other diagnostics reveal additional failures in evidence search, passive inference, and historical revision. Together, these results show that current agents can often act on available clues without recovering the causal structure needed to succeed when the objective changes.

1 INTRODUCTION

Large language models (LLMs) are increasingly expected to reason from information provided in context, yet relevant information is not always used reliably. Even when answer-relevant evidence is present, performance can depend substantially on where it appears in the context (Liu et al., 2024). Reliance on such positional correlations may not hurt performance in one context, but can lead to failure when the same evidence appears elsewhere. That is, an LLM may answer a question correctly when the relevant evidence appears in one position, yet fail on the same question when the same evidence is moved elsewhere in the context. For interactive agents, this problem is amplified because such spurious correlated evidence is accumulated over time through observations and the consequences of their own actions. To achieve robustness against these changes, given the relevant evidence, an agent should have the ability to pick up and uncover causal mechanisms, which remain stable across distributional variations. As Sherlock Holmes tells Watson, *"You see, but you do not observe. The distinction is clear"* (Doyle, 1955). For interactive agents, the analogous challenge is to move beyond clues that suffice for the task at hand and recover causes that remain useful beyond it. We therefore ask: can an LLM agent move from *clues* to *causes*?

Moving from *clues* to *causes* requires reasoning about what produced an outcome, what change could alter it, and how the outcome would have differed under another past action. We use *causal reasoning* to mean using evidence to infer and apply the cause–effect relationships needed to answer these questions. Formal causal analysis provides precise definitions for interventional and counterfactual questions (Pearl, 2009).

Whether an agent can answer these questions depends on the evidence it can obtain. Passive records may already be enough to determine the answer. When passive evidence is insufficient, interven-

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

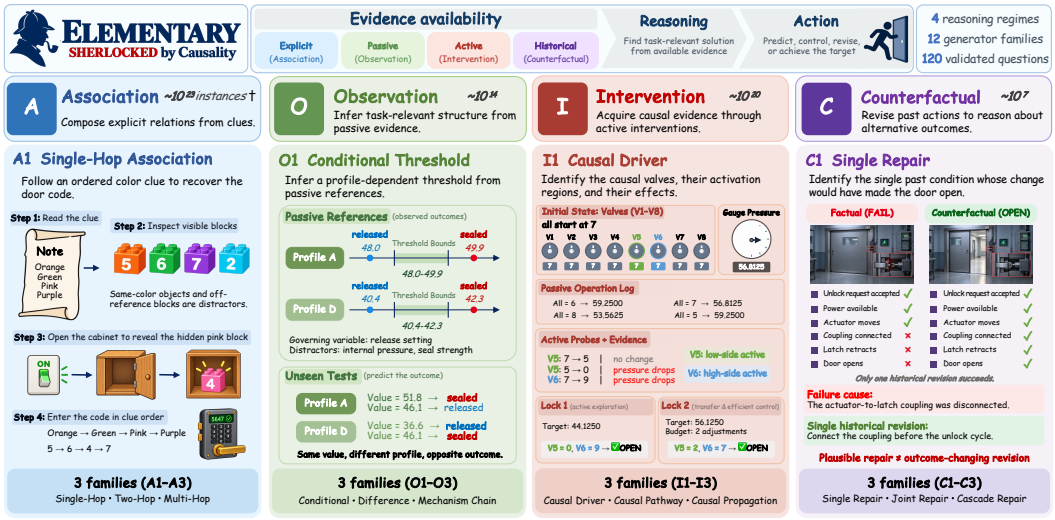


Figure 1: **Overview of ELEMENTARY.** Four reasoning regimes specify different evidence and action requirements within a shared interactive escape-room framework. Each regime contains three generator families, with one representative question illustrated. †Approximate generator capacity; see Appendix E for the counting protocol.

tions can provide the evidence needed to distinguish among competing explanations (Bramley et al., 2015). When an episode has already finished, the agent must instead reason from the record of what actually happened to determine how a different past action would have changed the outcome. We call the evidence sufficient when every explanation consistent with it leads to the same task answer.

Existing benchmarks cover important parts of this problem. Escape-room environments test exploration, tool use, and long-horizon puzzle solving (Qian et al., 2025; Guo et al., 2026), while interactive-discovery benchmarks evaluate hypothesis formation, experimentation, and model learning (Jansen et al., 2024; Gandhi et al., 2025; Warrier et al., 2025). Causal benchmarks further test formal causal questions and causal identification through intervention (Jin et al., 2023; Feng & Rudzicz, 2025; Chen et al., 2026). However, existing benchmarks do not place explicit relations, passive evidence, active intervention, and historical revision within a common interactive framework, making it difficult to compare agent behavior across different ways in which task-relevant evidence becomes available.

In this paper, we introduce **ELEMENTARY**, an escape-room benchmark for reasoning from clues to causes. Figure 1 provides an overview of the benchmark. **ELEMENTARY** is designed around two key properties: *interaction consistency* and *evidence sufficiency*. The former places all evidence conditions within a common interactive environment, while the latter ensures that each task is answerable from the evidence permitted under that condition. Building on these two principles, we organize the benchmark into four evidence regimes that differ in how the information needed for solving the task becomes available.

Association serves as a reference condition in which the relevant relations are explicitly encoded in the environment and agents must identify and compose them. **Observation** requires agents to infer the hidden cause–effect mechanism needed to solve the task using passive evidence that is already sufficient. **Intervention** begins from passive evidence that remains ambiguous and requires agents to gather additional evidence by acting on the environment. **Counterfactual** begins after an episode has finished and requires agents to revise past actions and reason about the outcome after those changes are replayed. Each regime contains three generator families, giving twelve families in total. A family fixes the reasoning structure, while its generator varies the concrete mechanism, target, evidence, and scene realization. This produces many distinct questions without changing the capability being tested. Before evaluation, every generated question is validated to ensure that its intended solution is reachable and that the permitted evidence determines the required answer.

Our contributions are threefold. **First, formulation.** We frame interactive causal reasoning in terms of what evidence an agent can obtain and whether that evidence is sufficient to determine the re-

quired solution. **Second, benchmark.** We build **ELEMENTARY**, a generator-based escape-room benchmark with four reasoning regimes and twelve generator families, and validate each generated question under its assigned evidence condition. **Third, evaluation.** We evaluate seven proprietary and open-weight LLMs on a fixed set of 120 validated questions under a common interaction protocol. The highest average success rate is 44.2%, and no model is consistently strong across evidence conditions. In I3 Causal Propagation, all $70/70$ model–question episodes reach the first control target, but only $1/70$ reaches the second when the new target requires using the hidden propagation structure. This result shows that success on the current task can substantially overstate the extent to which an agent has recovered reusable causal structure through interaction. Additional diagnostics reveal failures in evidence search, passive inference, and historical revision.

2 RELATED WORK

Interactive reasoning in escape-room environments. EscapeBench (Qian et al., 2025) studies exploration and tool use in text-based escape tasks. MM-Escape (Wang et al., 2025) and VisEscape (Lim et al., 2025) extend this setting to multimodal interaction, while AgentEscapeBench (Guo et al., 2026) studies long-range interaction dependencies. **ELEMENTARY** keeps the escape-room setting but varies how the evidence needed to solve a task becomes available: explicitly, through passive records, through intervention, or from a completed history.

Interactive discovery and world-model learning. RULEARN (He et al., 2025) and DiscoveryWorld (Jansen et al., 2024) study hidden-rule and scientific discovery through interaction. BoxingGym (Gandhi et al., 2025) and WorldTest (Warrier et al., 2025) further study experiment design, model learning, and the later use of acquired knowledge. In **ELEMENTARY**, some tasks can be solved from passive evidence alone. Others require the agent to intervene and gather additional evidence.

Causal and counterfactual reasoning. CLADDER (Jin et al., 2023) evaluates causal questions from supplied causal descriptions, while CausalPhys (Tang et al., 2026) studies causal reasoning from images and videos. CausalLink (Feng & Rudzicz, 2025) and CausalGame (Chen et al., 2026) study causal identification through interactive interventions. Conan (Xu et al., 2023) reasons from traces left by a previous agent and includes questions about alternative past actions. **ELEMENTARY** places passive inference, active evidence acquisition, and historical revision in the same interactive framework. Association additionally provides a reference condition in which the relevant relations are explicitly available.

3 THE ELEMENTARY FORMULATION

3.1 CAUSAL REASONING BACKGROUND

Structural causal models provide a formal language for reasoning about relationships in an observed system, the consequences of active interventions, and outcomes under alternative past actions (Pearl, 2009). These distinctions are commonly expressed through associational queries such as $p(Y = y \mid X = x)$, interventional queries such as $p(Y = y \mid \mathrm{do}(X = x))$, and counterfactual queries such as $p(Y_{X=x'} = y' \mid X = x, Y = y)$.

ELEMENTARY builds on these causal semantics but studies a different dimension of reasoning: how an interactive agent obtains enough evidence to determine a task-relevant answer. Rather than receiving all relevant information in advance, the agent may encounter explicit relations, passive records, responses produced by its own interventions, or traces left by a completed factual episode. We therefore organize interactive reasoning by how solution-relevant evidence becomes available and whether that evidence is sufficient.

In this formulation, **Association** serves as an explicit-relation reference condition. **Observation** asks whether passive evidence already determines the required solution, whereas **Intervention** requires active manipulation to resolve ambiguity left by passive evidence. **Counterfactual** retains the factual conditioning of counterfactual reasoning, but realizes alternative-history reasoning through historical revision and replay. Thus, Pearl's framework provides the causal semantics, while **ELEMENTARY** makes evidence availability and acquisition the controlled axis of interactive reasoning.

3.2 THE ELEMENTARY TASK FORMULATION

An **ELEMENTARY** task contains an interactive environment and a target g. At step t, the environment has state S_t, exposes an observation o_t, and accepts an action a_t. We use $M^\star$ to denote the *task-relevant mechanism*: the part of the environment that specifies the attributes, relations, and action-induced transitions needed to determine g. The same environment may contain additional objects and dynamics that are irrelevant to the current target.

For a candidate mechanism M, let $s(M, g)$ denote the task-level solution required by target g under that mechanism. Depending on the task, the solution may be a code, a set of predictions, a validated control solution, or a historical revision. For control tasks, $s(M, g)$ denotes the control solution required to reach the target under the hidden mechanism, rather than the target state itself. Let $\mathfrak{M}$ denote the candidate mechanism space and let D denote the evidence acquired by the agent. The mechanisms that remain compatible with D form the version space

$$\mathcal{V}(D) = \{M \in \mathfrak{M} : M \text{ is consistent with } D\}. \tag{1}$$

We say that the target solution is *identifiable* from evidence D when every mechanism that remains compatible with D implies the same solution:

$$|\{s(M, g) : M \in \mathcal{V}(D)\}| = 1. \tag{2}$$

Evidence can therefore be sufficient for the task even when the complete mechanism is not uniquely recovered. What matters is whether the remaining uncertainty can still change the required solution.

Identifiability is a property of the task evidence, not a guarantee of agent success. Even when the available evidence determines a unique solution, the agent must still locate the relevant information, combine evidence across interactions, and turn its inference into a valid action. This distinction lets us separate whether the task provides enough information from whether the agent can use that information effectively.

3.3 EVIDENCE-CONDITIONED REASONING REGIMES

Using the formulation above, we define four reasoning regimes by how the evidence needed to determine $s(M^\star, g)$ becomes available. The regimes share the same notion of task-level identifiability, but differ in the evidence initially available to the agent and in the interaction required to obtain sufficient evidence.

Association: explicit relations. In Association, the task-relevant relations are explicitly encoded in the environment. The agent must locate, ground, and compose these relations to recover the target solution. Because the relevant structure is available directly rather than inferred from variation in observed outcomes, Association provides a reference condition for reasoning when the required relations are explicit.

Observation: identification from passive evidence. In Observation, the task-relevant mechanism is hidden and the agent is restricted to evidence generated under the existing mechanism. Let D_{pass} denote this passive evidence. Observation tasks are constructed so that D_{pass} satisfies the identifiability condition in Eq. 2. The agent must therefore infer enough of the hidden mechanism to determine the target solution from passive records alone.

Intervention: active evidence acquisition. Intervention begins from passive evidence that does not yet determine the target solution. The agent may manipulate task-relevant variables and observe the resulting state changes. For some evidence D_{int} obtainable through legal interventions, the defining information condition is

$$|\{s(M, g) : M \in \mathcal{V}(D_{\text{pass}})\}| > 1, \qquad |\{s(M, g) : M \in \mathcal{V}(D_{\text{pass}} \cup D_{\text{int}})\}| = 1. \tag{3}$$

Thus, passive evidence alone is insufficient, but legal interventions can provide the additional evidence needed to determine the solution.

Counterfactual: historical revision and replay. In Counterfactual, the factual episode has already occurred. Let $\tau^{\text{F}} = (S_0, a_0, S_1, a_1, \ldots, S_T, a_T, S_{T+1})$ denote its factual trajectory, where S_t is the environment state before action a_t. The trajectory is generated under the task-relevant mechanism

$M^\star$, background conditions U, and agent policy π. Let D_{F} denote the evidence left by the factual aftermath. The agent uses D_{F} to infer the task-relevant hidden conditions in U and then selects one or more legally editable past actions to revise.

For example, replacing the factual action a_t with a'_t gives

$$Y^{\mathrm{CF}} = \mathrm{Replay}\left(M^\star, U, S_0, a_0, \ldots, S_t, a'_t, \pi\right). \tag{4}$$

Replay preserves the factual trajectory up to S_t, applies the revised action a'_t, and then regenerates the subsequent states and actions under the same mechanism $M^\star$, background conditions U, and policy π. A counterfactual solution is a legal historical revision whose replay satisfies the target g.

Together, the four regimes vary the path from evidence to solution: relations may be explicit, passive evidence may already be sufficient, additional evidence may need to be acquired through intervention, or the agent may need to reason from factual evidence about an alternative past. The next section turns these evidence conditions into concrete, generator-based interactive tasks.

4 Building the ELEMENTARY Benchmark

4.1 Generator-Based Construction

ELEMENTARY is organized hierarchically into reasoning regimes, generator families, questions, and episodes. Each of the four reasoning regimes contains three generator families, with each family defining a distinct reasoning structure. A generator instantiates that structure into concrete questions by varying the task-relevant mechanism, target, available evidence, and scene realization. Each generated question has a fixed ground-truth solution, and one model attempt on one question constitutes an episode.

We define *generator capacity* as the number of valid task instances supported by the implemented generator constraints after excluding surface-only variations. Two instances are treated as distinct when they differ in the task-relevant mechanism, target, evidence, or interaction structure. The resulting spaces span approximately 10^{23} instances for Association and 10^{7} for Counterfactual, with verified lower bounds of 10^{14} for Observation and 10^{20} for Intervention, as summarized in Figure 1. Family-level capacities and the counting procedure are provided in Appendix E.

Generated questions are admitted only after regime-specific validation. The validators check that the intended solution is reachable, that the available evidence satisfies the information condition defined by the corresponding reasoning regime, and that the required interaction structure is preserved. Association additionally verifies relation-path and interaction-budget constraints, while the other regimes verify passive identifiability, active ambiguity resolution, or historical-replay requirements as appropriate. The validators also reject shortcut instances in which public surface features reveal private mechanism roles or bypass the intended reasoning path. Solution-relevant information must therefore come from the evidence channel defined by the task rather than from generator artifacts. Detailed validation procedures are provided in Appendices A, B, C, and D.

4.2 Generator Families

Across the four regimes, the twelve generator families vary the reasoning structure while preserving the regime-level evidence condition. Table 1 gives a compact overview of the resulting task space.

Detailed family specifications, construction rules, and validation procedures are provided in Appendices A, B, C, and D.

4.3 Regime-Specific Interaction Design

All four regimes use the same interactive escape-room framework, but the interaction protocol determines how the agent can acquire and use solution-relevant evidence. We therefore adapt the runtime to each evidence condition while keeping the agent responsible for deciding what information to inspect and when to commit to a solution.

Association. Relation-carrying attributes are revealed through interaction rather than listed in the initial observation. Natural-language descriptions require the agent to ground attributes before com-

Table 1: The twelve **ELEMENTARY** generator families. Families within a regime share the same evidence condition but differ in the reasoning structure required to determine the solution.

Family	Structure	Reasoning demand
Association	*Explicit task-relevant relations*	
A1 Single-Hop	color → digit	Ground a direct relation from a referenced attribute to its terminal value.
A2 Two-Hop	marker → shape → digit	Compose two explicit relations while preserving the identity of the reference item.
A3 Multi-Hop	symbol → material → color → code mark → digit	Maintain reference-consistent composition across a longer relation chain.
Observation	*Sufficient passive evidence*	
O1 Threshold	Profile-dependent threshold over a hidden governing measurement	Identify the governing variable and transfer the inferred threshold rule across profiles and boundary cases.
O2 Difference	Threshold over a hidden pairwise relation	Recover the relevant derived quantity and apply the inferred rule to unseen cases.
O3 Chain	Multi-stage mechanism with intermediate states and logical gates	Reconstruct a compositional mechanism from passive records.
Intervention	*Additional evidence through active manipulation*	
I1 Driver	Hidden causal drivers with activation-dependent effects	Distinguish inactive causal drivers from controls with no causal effect.
I2 Pathway	Shared mediated pathway with additional direct effects	Separate direct effects from mediated changes through controlled interventions.
I3 Propagation	Directed propagation with hidden channel constraints	Recover causal direction and conditional transmission rules.
Counterfactual	*Factual evidence with historical revision*	
C1 Single Repair	One historical action revision	Identify the unique past change that produces successful replay.
C2 Joint Repair	Two jointly necessary historical revisions	Select a compatible pair of edits without intermediate replay feedback.
C3 Cascade Repair	Early revision with downstream causal propagation	Track how an earlier change alters the later factual trajectory.

posing the relation chain, while structured distractors create locally plausible but globally invalid continuations. A question-specific *Puzzle Interaction Budget* (PIB) defines an allowance on puzzle-evidence selections and separates targeted evidence collection from broad inspection. The budget is soft at runtime: exceeding it does not terminate the episode, and the agent may still recover the correct code. Official success, however, requires the correct solution within PIB; a correct solution reached afterward is recorded separately as over-budget recovery. Grounding, distractor construction, PIB computation, runtime, and validation are detailed in Appendix A.

Observation. Each question provides passive *References* generated by the hidden mechanism and a set of unseen *Tests*. Each question contains eight References; O1, O2, and O3 contain four, three, and six Tests, respectively, reflecting the different coverage required by their hidden structures. The agent chooses which References to inspect before revealing the Tests. Once the Tests are revealed, further environment exploration stops and the agent submits one joint prediction over all Tests. No Test-level correctness feedback is provided before the submission is final. This design tests whether the mechanism inferred from passive records transfers to unseen cases rather than being refined through sequential trial and error. Reference–Test construction, identifiability, runtime, and validation are detailed in Appendix B.

Intervention. The agent can actively manipulate task-relevant controls and use the resulting state changes as evidence. During *Lock 1*, it may inspect and intervene while working toward an initial target. When that target is reached, the environment state is preserved and *Lock 2* reveals a new target with a limited number of additional state-changing adjustments. Lock 2 permits at most two

effective adjustments in I1 and I2 and five in I3; inspections and passive downstream changes do not consume this budget. The second stage therefore requires the agent to use information acquired during exploration rather than repeat extensive exploration from scratch. Family-specific mechanisms, target construction, adjustment semantics, runtime, and validation are detailed in Appendix C.

Counterfactual. Each question opens in the read-only aftermath of a completed factual episode. The agent first inspects distributed Scene traces to recover the factual conditions relevant to the failure, then enters historical replay and selects legal revisions to recorded past actions. C1 and C3 commit one revision, whereas C2 stages two revisions before a single joint submission. The revised history is replayed once, with no intermediate counterfactual feedback that could be used to test alternative repairs. Evidence construction, historical action semantics, replay, runtime, and validation are detailed in Appendix D.

Together, these protocols turn the regime-level evidence conditions into distinct interactive demands: locating explicit evidence, transferring from passive records, acquiring new evidence through intervention, and revising a completed history. They define the benchmark interaction used for agent evaluation in the experiments that follow.

5 Experiments

5.1 Experimental Setup

Models and experimental setting. We evaluate seven LLMs spanning proprietary and open-weight models: GPT-5.6 Sol (OpenAI, 2026), Claude Opus 5 (Anthropic, 2026), Grok 4.6 (xAI, 2026), Gemini 3.8 Flash (Google DeepMind, 2026), DeepSeek-V4.1-Flash (Xu et al., 2026), Kimi K3 (Team et al., 2026), and Qwen3.8-2.4T-A95B (Qwen Team, 2026). For each generator family, we deterministically instantiate ten evaluation questions using base seed 42 and retain the validated questions indexed 000000–000009. The resulting 120-question evaluation set is then fixed and reused unchanged across all models; one model attempt on one question constitutes one episode. Claude Opus 5 is evaluated on the 90 Association, Observation, and Intervention questions; its Counterfactual evaluation is unavailable because the provider-side safety mechanism blocks execution of that protocol. All models receive the same regime-specific prompt and interaction schema and are evaluated under the same benchmark-side configuration, with temperature set to 0 when supported and no model-specific reasoning or effort parameters specified. A 50-step environment horizon applies with regime-specific scope: the full episode in Association and Counterfactual, the Reference phase in Observation, and Lock 1 in Intervention. Detailed model identifiers and execution configurations are provided in Appendix G.5.

Agent interaction. We evaluate each model as an interactive text agent under a unified environment loop. At each decision step, the agent receives the current observation, available objects, legal actions, and previously acquired factual context, and returns one structured action executed by the environment. The environment then provides the resulting observable feedback. All regimes share the same interaction interface while differing in the evidence available and terminal objectives: Association requires relation inspection and one-shot code submission, Observation requires passive evidence acquisition followed by joint Test prediction, Intervention requires active evidence acquisition and limited control transfer, and Counterfactual requires factual inspection followed by historical revision and replay. Private mechanism variables, evaluator-side correctness signals, and next-action guidance are never exposed to the agent. Detailed action schemas and runtime implementation are provided in Appendix G.

Evaluation metrics. We report two types of metrics: *official success criteria* and *diagnostic metrics*. The *official success criteria* define the final benchmark outcome for each reasoning regime. In Association, an episode succeeds when the agent recovers the target solution within PIB. In Observation, success requires all Test predictions to be correct in the final joint submission. In Intervention, success requires Lock 1 target completion and Lock 2 transfer within the corresponding adjustment budget. In Counterfactual, success requires the submitted historical revision to produce the correct replayed outcome.

The *diagnostic metrics* further characterize the sources of success and failure within each regime. These diagnostics measure evidence-search efficiency in Association, Test-level prediction accu-

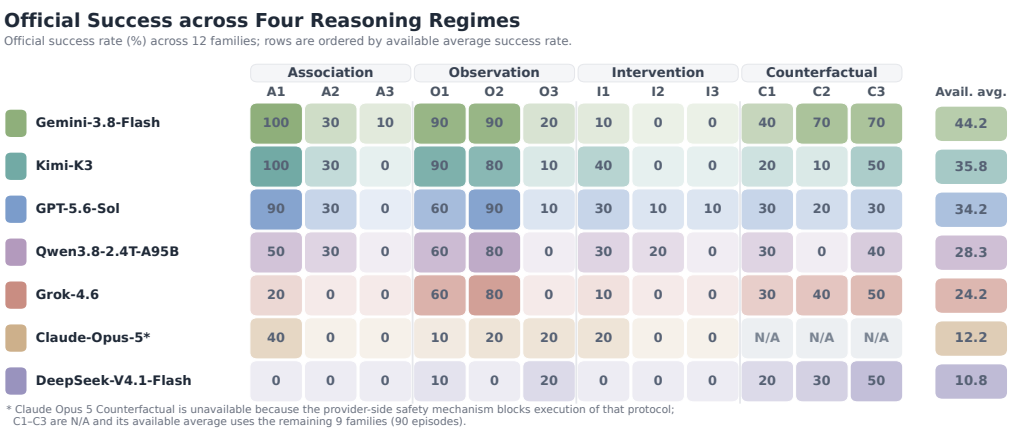


Figure 2: **Performance varies sharply across evidence conditions and reasoning structures.** Official success rates (%) are reported over ten evaluation questions per generator family. Rows are ordered by available average success rate.

racy in Observation, first-target completion and constrained transfer in Intervention, and evidence acquisition, historical revision, and replay impact in Counterfactual. Detailed metric definitions and regime-specific computation procedures are provided in Appendix H.

5.2 Overall Performance Across Evidence Conditions

Figure 2 reports official success across the four reasoning regimes and twelve generator families. Each cell averages over ten evaluation questions from one family.

No model is consistently strong across evidence conditions. Gemini-3.8-Flash has the highest average success rate at 44.2%, yet its performance ranges from 100% on A1 and 90% on O1–O2 to 0% on I2–I3. Grok-4.6 shows a different profile: it reaches only 0–20% across Association, but 60–80% on O1–O2 and 30–50% across the Counterfactual families. Thus, aggregate success masks large variation across reasoning regimes.

Models struggle as reasoning becomes more compositional. Within Association, A1 reaches 100% for the strongest models, whereas A3 is at most 10% for every model. A similar pattern appears in Observation: O1 and O2 reach 90%, while no model exceeds 20% on the multi-stage O3 family. Notably, O2 already requires recovering a hidden pairwise relation, yet five models reach 80–90% success. The contrast suggests that models become substantially less reliable when the required inference must be composed across multiple intermediate states and logical operations.

Longer causal chains are not necessarily harder to reason through. C3 requires reasoning through a multi-stage downstream causal trajectory, yet its success rate matches or exceeds C2 for every model with available Counterfactual results. For five of the six models, C3 success is strictly higher than C2. C2 instead requires two jointly necessary historical revisions to be selected before any replay feedback is observed. These results show that causal depth alone is not sufficient to explain the observed difficulty; the structure of the required revision also matters.

The regimes do not form a simple difficulty ladder. Performance does not decrease monotonically from explicit relations to counterfactual reasoning. Grok-4.6 averages only 6.7% across Association but 40.0% across Counterfactual, while DeepSeek-V4.1-Flash rises from 0% to 33.3% across the same two regimes. These reversals reinforce that the four regimes probe distinct ways of turning evidence into a solution rather than successive levels of a single difficulty scale.

5.3 Do Agents Move from Clues to Causes?

Table 2 summarizes current-target success and subsequent control across the three Intervention families. The Intervention regime tests whether an agent can learn how the system works through active

Table 2: **Current-target success and subsequent control in Intervention.** Lock 2 success is conditioned on Lock 1 completion. Results aggregate ten questions per family across seven models.

Family	Lock 1 success	Lock 2 success | Lock 1
I1 Causal Driver	$26/70$ (37.1%)	$14/26$ (53.8%)
I2 Causal Pathway	$21/70$ (30.0%)	$3/21$ (14.3%)
I3 Causal Propagation	$70/70$ (100.0%)	$1/70$ (1.4%)

exploration and use that knowledge when the goal changes. During Lock 1, the agent can probe the system and use the resulting feedback to solve an initial control problem. Lock 2 then keeps the same hidden mechanism, changes the target, and allows only a few additional state-changing actions. If the agent has learned how its actions affect the system during Lock 1, it should be able to use that knowledge to achieve the new target.

The three Intervention families fail at different stages. In I1, only 37.1% of episodes reach Lock 1, but more than half of those successful episodes also complete Lock 2, suggesting that much of the difficulty arises during initial exploration and control. I2 remains difficult at both stages: only 30.0% reach Lock 1, and just 14.3% of those episodes complete Lock 2. I3 shows the opposite pattern. Every episode reaches Lock 1, yet only 1.4% succeeds after the target changes. The low I3 success rate therefore does not come from an inability to reach the initial control target; the failure appears specifically in the second stage.

I3 concentrates the failure in causal transfer. Its Lock 1 target can be reached while the directed Relay graph and hidden channel conditions remain unresolved. Under Lock 2, these causal dependencies become necessary for action: target values activate hidden channels, and setting an upstream Relay can overwrite states that were established earlier downstream. Reaching the new target within five effective adjustments therefore requires the agent to infer which Relays influence which others and to use that structure when ordering its actions. Lock 1 success can thus be achieved without resolving the causal structure that Lock 2 requires. When the new target requires the agent to use this propagation structure to plan its actions, only 1 of 70 episodes succeeds.

Current-target success can overstate reusable causal understanding. The main failure exposed by I3 is that current agents do not reliably extract reusable causal structure from intervention feedback. Their success remains strongly tied to the current objective: when the goal changes, they often fail to use what their interventions revealed about the same underlying mechanism to plan what to do next. In I3, agents can follow the clues far enough to reach the immediate goal, but rarely use those clues to reason about the causes that matter for the next one. Further regime-specific diagnostics are reported in Appendix I.

6 Conclusion

We introduced **Elementary**, an interactive benchmark for testing whether LLM agents can use available evidence to make the right predictions and actions. The benchmark varies how the evidence needed to solve a task is obtained: it may be given explicitly, inferred from passive observations, discovered through intervention, or reconstructed from a completed history. Each generated question is validated so that the evidence available in its setting is sufficient to determine the required solution.

Across seven proprietary and open-weight LLMs, no model performs well across all four regimes, and the highest average success rate is only 44.2%. The clearest failure appears in I3 Causal Propagation: all $70/70$ episodes reach the first target, but only $1/70$ reaches the second when the new target requires the agent to use the hidden propagation structure to plan its actions. Success on the current task can therefore hide a major weakness: the agent may reach the target without learning enough about how the system works to control it again when the goal changes. Other diagnostics reveal additional failures in evidence search, passive inference, and historical revision.

Overall, current agents can often use clues to solve the task in front of them, but they do not reliably turn those clues into causal knowledge that remains useful when the objective changes. This is the gap that **Elementary** is designed to measure: whether the task provides enough evidence, and whether the agent can actually use that evidence to move from *clues* to *causes*.

AI USE STATEMENT

We used generative AI tools primarily for language polishing and editing, and for limited discussion of experimental design during the preparation of this manuscript. All final research decisions, methodologies, implementations, experiments, analyses, and conclusions were developed and verified by the authors. The authors reviewed all AI-assisted content and take full responsibility for the final manuscript and its claims.

REPRODUCIBILITY STATEMENT

We will release the code, prompts, configurations, and the exact 120 evaluation questions used in this paper, together with tools for generating and validating additional tasks. These materials will support reproduction of the reported experiments and extension of the benchmark.

REFERENCES

Anthropic. Introducing Claude Opus 5, July 2026. URL https://www.anthropic.com/news/claude-opus-5. Official model announcement.

Neil R Bramley, David A Lagnado, and Maarten Speekenbrink. Conservative forgetful scholars: How people learn causal structure through sequences of interventions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(3):708, 2015.

Zhenhao Chen, Yongqiang Chen, Chenxi Liu, Junchi Yu, Xiangchen Song, Zijian Li, Jialin Li, Philip Torr, Bo Han, and Kun Zhang. CausalGame: Benchmarking causal thinking of LLM agents in games. *arXiv preprint arXiv:2607.04293*, 2026.

Arthur Conan Doyle. *The Adventures of Sherlock Holmes*. WHITMAN Publishing Company, 1955.

Jinyue Feng and Frank Rudzicz. CausalLink: An interactive evaluation framework for causal reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22313–22326, 2025.

Kanishk Gandhi, Michael Y Li, Lyle Goodyear, Agam Bhatia, Louise Li, Aditi Bhaskar, Mohammed Zaman, and Noah D Goodman. BoxingGym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025.

Google DeepMind. Gemini 3.8 Flash model card, September 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/. Google DeepMind model card.

Zhengkang Guo, Yiyang Li, Lin Qiu, Xiaohua Wang, Jingwen Xv, Dongyu Ru, Xiaoyu Li, Xiaoqing Zheng, Xuezhi Cao, and Xunliang Cai. AgentEscapeBench: Evaluating out-of-domain tool-grounded reasoning in LLM agents. *arXiv preprint arXiv:2605.07926*, 2026.

Kaiyu He, Mian Zhang, Shuo Yan, Peilin Wu, and Zhiyu Zoey Chen. Idea: Enhancing the rule learning ability of large language model agent through induction, deduction, and abduction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13563–13597, 2025.

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.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36: 31038–31065, 2023.

Seungwon Lim, Sungwoong Kim, Jihwan Yu, Sungjae Lee, Jiwan Chung, and Youngjae Yu. VisEscape: A benchmark for evaluating exploration-driven decision-making in virtual escape rooms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16031–16058, 2025.

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

OpenAI. GPT-5.6 System Card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6. OpenAI Deployment Safety Hub.

Judea Pearl. Causal inference in statistics: An overview. 2009.

Cheng Qian, Peixuan Han, Qinyu Luo, Bingxiang He, Xiusi Chen, Yuji Zhang, Hongyi Du, Jiarui Yao, Xiaocheng Yang, Denghui Zhang, et al. EscapeBench: Towards advancing creative intelligence of language model agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 798–820, 2025.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Tianyi Tang, Zhuoyi Lin, Zeyu Feng, Tianyi Ma, Yew-Soon Ong, Ivor Tsang, and Haiyan Yin. Causal scaffolding for physical reasoning: A benchmark for causally-informed physical world understanding in VLMs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9848–9859, 2026.

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 preprint arXiv:2607.24653*, 2026.

Ziyue Wang, Yurui Dong, Fuwen Luo, Minyuan Ruan, Zhili Cheng, Chi Chen, Peng Li, and Yang Liu. How do multimodal large language models handle complex multimodal reasoning? placing them in an extensible escape game. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4807–4817. IEEE, 2025.

Archana Warrier, Dat Nguyen, Michelangelo Naim, Moksh Jain, Yichao Liang, Karen Schroeder, Cambridge Yang, Joshua B Tenenbaum, Sebastian Vollmer, Kevin Ellis, et al. Benchmarking world-model learning with environment-level queries. *arXiv preprint arXiv:2510.19788*, 2025.

xAI. Introducing Grok 4.6, August 2026. URL https://x.ai/news/grok-4-6. Official model announcement.

Anyi Xu, B Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, CC Yu, et al. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Manjie Xu, Guangyuan Jiang, Wei Liang, Chi Zhang, and Yixin Zhu. Active reasoning in an open-world environment. *Advances in Neural Information Processing Systems*, 36:11716–11736, 2023.

APPENDIX CONTENTS

A Association Benchmark Details **14**
A.1 Generator Specifications . . . 14
A.2 Description-Based Attribute Grounding . . . 14
A.3 Structured Distractor Construction . . . 15
A.4 Puzzle Interaction Budget . . . 15
A.5 Runtime Protocol . . . 16
A.6 Structural Validation . . . 17

B Observation Benchmark Details **17**
B.1 Generator Specifications . . . 17
B.2 Reference and Test Construction . . . 18
B.3 One-Shot Runtime Protocol . . . 19
B.4 Prediction Identifiability . . . 19
B.5 Structural Validation . . . 19
B.6 Anti-Shortcut Construction . . . 19

C Intervention Benchmark Details **20**
C.1 Generator Specifications . . . 20
C.2 Exploration-to-Transfer Construction . . . 21
C.3 Intervention and Adjustment Semantics . . . 21
C.4 Family-Specific Target Construction . . . 22
C.5 Structural Validation . . . 22
C.6 Runtime Protocol . . . 22

D Counterfactual Benchmark Details **23**
D.1 Generator Specifications . . . 23
D.2 Factual Scene Evidence and Abduction . . . 23
D.3 Historical Action Space and Replay . . . 23
D.4 Competing Factual Explanations . . . 24
D.5 One-Shot Runtime Protocol . . . 24
D.6 Structural Validation . . . 25

E Generator Capacity **25**

F Representative Question Instances **26**
F.1 Association Examples . . . 26
F.2 Observation Examples . . . 27
F.3 Intervention Examples . . . 28
F.4 Counterfactual Examples . . . 30

G Agent Evaluation Details **31**

G.1 Agent Interface and Action Schemas 31

G.2 Prompt Templates 31

G.3 Interaction Context and Memory 32

G.4 Regime-Specific Runtime Protocols 33

G.5 Experimental Configuration 33

H Diagnostic Metric Definitions **34**

H.1 Association Diagnostics 34

H.2 Observation Diagnostics 34

H.3 Intervention Diagnostics 35

H.4 Counterfactual Diagnostics 36

I Regime-Specific Failure Diagnostics **37**

J Family-Level Diagnostic Results **38**

J.1 Association 38

J.2 Observation 38

J.3 Intervention 39

J.4 Counterfactual 39

Table 3: Representative examples of description-based attribute grounding.

Attribute	Canonical value	Example description
Marker	`chevrons`	repeated V-shaped strokes
Shape	`triangle`	a three-sided angular outline
Material	`wood`	a surface with visible natural grain
Color	`green`	a muted leafy-green tone

A Association Benchmark Details

A.1 Generator Specifications

The three Association generator families share the same escape-room environment and interaction protocol, but differ in the number of explicit relations that must be composed. Each generated question contains an ordered reference specifying four target labels. The remaining puzzle objects instantiate the relations that connect these labels to four digits, together with distractors that create locally plausible but globally invalid alternatives. Concrete labels, relation assignments, object realizations, and scene placements vary across questions.

A1: Single-Hop Association. A1 instantiates a direct relation color $\rightarrow$ digit. The reference specifies an ordered set of target colors, and the agent must identify the digit associated with each color. Candidate objects include both solution-relevant objects and distractors that match a target color without providing a valid digit, as well as digit-bearing objects whose colors do not belong to the reference. The four recovered digits are assembled in the order specified by the reference.

A2: Two-Hop Association. A2 introduces an intermediate relation, marker $\rightarrow$ shape $\rightarrow$ digit. The reference specifies an ordered set of target markers. The agent must first identify the shape associated with each marker and then follow that shape to its digit. Distractors may agree with the reference at one stage of the chain while failing at a later stage, or may resemble valid terminal objects without being reachable from the active relation chain. Solving A2 therefore requires maintaining reference identity across both relation steps.

A3: Multi-Hop Association. A3 extends the relation chain to symbol $\rightarrow$ material $\rightarrow$ color $\rightarrow$ code mark $\rightarrow$ digit. The reference specifies an ordered set of target symbols, and each symbol must be propagated through four explicit relations before reaching a digit. The longer chain introduces ambiguity at multiple intermediate stages: a candidate may match the currently relevant value while leading to a dead end, or an apparently complete chain may originate from a value that is not part of the reference. The agent must therefore preserve the identity of each reference item across the complete relation chain.

A.2 Description-Based Attribute Grounding

Association does not reduce to literal string matching between the reference and candidate objects. Reference objects expose the canonical labels that define the relation to be solved, whereas candidate objects express the corresponding attributes through natural-language descriptions. The agent must therefore ground a linguistic description to the underlying attribute before it can follow the relation chain.

For example, a reference may contain the marker `chevrons`, while a candidate object describes the same marker as "repeated V-shaped strokes." Likewise, shapes, materials, colors, symbols, and code marks can be expressed through surface descriptions rather than by directly exposing their canonical matching keys. Table 3 shows representative examples.

Each canonical value is associated with multiple possible surface realizations. The realization used in a particular question is fixed when the question is generated, so all agents evaluated on that question receive the same description. When the same underlying value appears more than once within a question, surface realizations can differ while preserving the same semantic relation. This

Table 4: Structured distractors used in the Association regime.

Family	Distractor type	Why it is locally plausible	Why it does not solve the reference
A1	Reference-matching nonterminal	Matches a target color	Does not provide the digit required to complete the relation
A1	Off-reference terminal	Provides a digit and resembles a solution object	Its color is not among the ordered reference targets
A2	Early dead end	Matches a target marker	The resulting intermediate value does not continue to a valid terminal
A2	Off-reference intermediate path	Produces a plausible intermediate value	Its starting marker does not belong to the active reference
A2	Unreachable terminal	Resembles a valid terminal object	No active intermediate relation reaches it
A3	Off-reference complete chain	Forms an apparently valid multi-hop path	The chain originates from a value outside the ordered reference
A3	Intermediate chain break	Matches the active path through one or more stages	The path terminates before reaching a valid digit
A3	Locally matching continuation	Matches the currently relevant intermediate value	Its continuation cannot complete the reference-consistent chain

prevents the solution from depending only on exact lexical identity and requires the agent to connect natural-language descriptions to the explicit relational structure.

A.3 Structured Distractor Construction

Distractors are designed to create ambiguity within the explicit relation chain rather than to add unrelated objects to the room. A distractor is therefore typically plausible under the evidence available at one stage of reasoning, but fails to form a complete path from the ordered reference to a valid digit. Table 4 summarizes the main distractor structures used in the three Association generator families.

As shown in Table 4, the distractors are placed at different points of the relation chain as the family depth increases. A1 mainly tests whether the agent can distinguish reference-consistent evidence from superficially relevant objects. A2 introduces ambiguity at the intermediate relation. A3 extends this ambiguity across multiple stages, including locally valid continuations and complete-looking chains that are disconnected from the ordered reference. Solving these questions therefore requires the agent to track both the current relation and its origin in the reference, rather than relying on local attribute matching alone.

A.4 Puzzle Interaction Budget

The *Puzzle Interaction Budget* (PIB) measures how many puzzle-evidence objects an agent needs to select while solving a question. It is defined independently of the environment-step horizon. The purpose of PIB is to distinguish targeted evidence collection from broad search: two agents may both recover the correct code, while one follows the relevant relation chain and the other inspects most available puzzle objects.

Counted interactions. PIB increases when the agent selects an object that can provide puzzle evidence. The reference object is counted, as are selections of relation-carrying candidate objects. Re-selecting a counted object consumes PIB again. In contrast, actions that only change physical accessibility or submit the final answer do not consume PIB.

Table 5: Examples of interactions that do and do not consume PIB.

Counts toward PIB	**Does not count**
Reference selection	Using the Wall Switch
Selecting a puzzle carrier	Opening the Cabinet
Re-selecting a puzzle carrier	Submitting the final code

Per-question allowance. The PIB allowance is computed separately for each generated question from the sequence of evidence that becomes visible or accessible during interaction. The computation follows the intended reference-guided solution process. For an unresolved reference item, the currently available evidence may leave several candidate objects compatible with the relation observed so far. The allowance includes the worst-case number of candidate selections needed to identify the valid continuation. After a selection reveals new evidence, visibility, accessibility, and the set of plausible continuations are updated before the next part of the allowance is computed.

Once a reference item has reached its terminal digit, it is treated as solved. Later selections that are unnecessary for resolving that item are therefore not included in its allowance. This prevents the PIB calculation from treating exhaustive inspection of all matching objects as necessary evidence collection.

Conceptually, the allowance is obtained by the following procedure:

1. acquire the ordered reference;
2. choose an unresolved reference item;
3. determine the currently accessible objects that remain plausible under the relation path observed so far;
4. allow for the worst-case number of selections needed to identify the valid continuation;
5. update the evidence and accessible state after that continuation is found;
6. repeat until the item reaches a digit, then continue with the remaining reference items.

The resulting allowance is therefore question-specific: questions with different ambiguity or accessibility patterns can receive different PIB values even when they belong to the same generator family.

Soft-budget semantics. PIB is a diagnostic budget rather than a termination condition. Let $B(q)$ denote the PIB allowance for question q and let $U(e)$ denote the number of counted selections made in episode e. We distinguish

$$U(e) \leq B(q) \quad \text{and} \quad U(e) > B(q), \tag{5}$$

as solving *within PIB* and operating *over PIB*, respectively. Exceeding the allowance does not block later observations or terminate the episode. Instead, the excess

$$\max\left(0, U(e) - B(q)\right) \tag{6}$$

is recorded as PIB overage. This allows final success and evidence-collection efficiency to be evaluated separately.

A.5 Runtime Protocol

Each Association question is executed in a fresh environment state. The reference is not automatically provided to the agent: it must be selected during interaction before its ordered target labels are revealed. Likewise, relation-carrying attributes of candidate objects become available only after the corresponding object has been legally selected.

The environment maintains two distinct interaction quantities. The *environment-step count* tracks the overall interaction horizon, whereas PIB counts only puzzle-evidence selections as defined in Appendix A.4. An episode may therefore consume environment steps without consuming PIB, for

Table 6: Structural validation criteria for Association questions.

Criterion	Validation requirement
Unique solution	The ordered reference induces exactly one valid final code.
Reachability	Every solution-relevant relation can be reached through legal interaction from the initial environment state.
No alternative terminal path	Distractor relations cannot form a second reference-consistent path to a different valid code.
Observation masking	Attributes intended to require interaction are not exposed before the corresponding legal reveal.
Reference consistency	The final code follows the order specified by the reference rather than an unrelated ordering of otherwise valid digits.
Distractor validity	Each structured distractor satisfies its intended role while remaining outside the valid reference-to-digit solution path.
PIB reproducibility	The PIB allowance can be independently recomputed from the final instantiated question and its sequential accessibility structure.

example when an action only changes access to another part of the room. Association episodes use a maximum environment horizon of 50 steps.

The final four-digit code can be submitted only once. A correct submission solves the question, whereas an incorrect submission immediately terminates the episode as a wrong-code failure. This prevents repeated code guessing from substituting for evidence collection. Exceeding PIB does not itself terminate the episode; an over-PIB agent retains access to the same environment and may still reach a correct final solution.

A.6 Structural Validation

Every generated Association question is validated before evaluation. Validation operates on the instantiated question structure and checks whether the public puzzle evidence supports the intended solution. The main criteria are summarized in Table 6.

Family-specific checks additionally verify that the instantiated relation graph has the intended depth and structure. For A1, each reference item must resolve through a valid direct relation to one terminal digit. For A2, the reference-consistent two-hop paths must be complete while distractor paths fail to produce an alternative solution. For A3, the same requirement is applied across the full four-hop chain, including distractors whose failure occurs at different intermediate stages. Questions that fail any required structural check are excluded before model evaluation.

B Observation Benchmark Details

B.1 Generator Specifications

All Observation families share the same high-level structure: the agent first inspects a set of passive historical References, infers a hidden task-relevant relation, and then applies that relation to held-out Tests. The three families differ in the structure that must be recovered.

O1: Conditional Threshold. O1 hides one governing measurement, a comparison direction, and profile-specific thresholds. Each Reference contains several numeric measurements, a profile label, and the resulting outcome. Only one measurement governs the outcome. The hidden relation can be written conceptually as

$$Y = \mathbb{I}[X_j \gtrless \theta_c], \tag{7}$$

where X_j is the governing measurement, c denotes the profile, and both the comparison direction and threshold θ_c are hidden. The References need not identify an exact real-valued threshold. They must constrain each threshold sufficiently to determine the held-out Test outcomes.

Table 7: Reference and Test construction in the Observation regime.

Family	References	Tests	Role of the Tests
O1	8	4	Two profile-specific pairs, with one Test on each side of the corresponding threshold.
O2	8	3	Two Tests probe opposite sides of the hidden difference threshold; a third provides an additional unseen transfer case.
O3	8	6	Four Tests use unseen input combinations, while two support Tests provide coverage of mechanism components not guaranteed by the four unseen cases alone.

O2: Difference Threshold. O2 moves from a threshold on one observed quantity to a threshold on a relation between quantities. Several numeric measurements are visible, but the outcome is governed by one hidden pairwise difference:

$$Y = \mathbb{I}[X_i - X_j \gtrless \theta]\,. \tag{8}$$

The agent must determine which pairwise difference is relevant, infer its comparison direction, and constrain the corresponding threshold from passive evidence. The raw measurements themselves are constructed so that no individual measurement provides an equivalent threshold explanation.

O3: Mechanism Chain. O3 replaces a single threshold relation with a multi-stage hidden mechanism. Three input conditions are composed through two intermediate states:

$$Z_1 = C_1(X_1), \qquad Z_2 = G_1(Z_1, C_2(X_2))\,, \qquad Y = G_2(Z_2, C_3(X_3))\,, \tag{9}$$

where each C_k is a hidden threshold-and-direction condition and G_1 and G_2 are hidden logical gates. References expose the three inputs, the two intermediate states, and the final outcome. The agent must therefore reconstruct the mechanism stage by stage rather than rely only on an end-to-end mapping from raw inputs to the final outcome.

B.2 Reference and Test Construction

Each Observation question contains eight passive References. References and Tests are constructed jointly so that the available evidence supports the intended inference while the held-out cases test whether that inference transfers beyond the observed examples. The number of Tests is family-specific because the three hidden structures require different diagnostic coverage.

O1: profile-specific boundary coverage. The four O1 Tests form two profile-specific pairs. Within each pair, the two Test cases lie on opposite sides of the corresponding hidden threshold. Correctly predicting all four cases therefore requires the inferred relation to transfer across both profiles and both sides of their decision boundaries.

O2: boundary diagnosis and unseen transfer. O2 uses two core diagnostic Tests on opposite sides of the governing difference threshold. These cases test whether the agent has recovered both the relevant derived quantity and its direction. A third Test evaluates the same inferred relation on an additional unseen case rather than repeating another boundary example.

O3: sparse compositional transfer. O3 deliberately avoids presenting the complete combination table of its three input conditions. The eight References contain multiple numeric realizations of a sparse set of diagnostic input combinations. Four Tests then evaluate complementary combinations that were not present in the References.

These four unseen cases alone do not guarantee that every component of the hidden mechanism affects the prediction set. O3 therefore includes two additional support Tests with held-out numeric values. The complete six-Test set is constructed so that the different input conditions and both hidden logical gates are exercised by at least one Test. The support cases are selected independently of the private ground-truth mechanism, preventing the Test layout itself from revealing the hidden structure.

B.3 One-Shot Runtime Protocol

Each Observation question begins with eight visible Reference objects. Their historical records are not automatically included in the model observation. The agent must select a Reference to reveal its contents and may inspect References in any order.

The Reference phase ends when the agent explicitly reveals the Tests or when the 50-step Reference horizon is reached. An explicit reveal consumes one environment step. A forced reveal at the horizon does not consume an additional step and is not itself treated as a task failure.

All Tests are revealed simultaneously. The previously acquired Reference evidence remains available, but no further environment exploration is allowed after the Test phase begins. The agent must then submit one complete prediction map covering every Test in the question.

The first complete schema-valid submission is final. No Test-level correctness feedback is provided before termination, and the agent cannot revise individual predictions after a valid submission. The question is solved only when every Test prediction is correct; otherwise the episode terminates as a prediction failure. This protocol prevents sequential feedback and repeated guessing from substituting for passive-rule inference.

B.4 Prediction Identifiability

Observation is built around prediction identifiability rather than exact recovery of every hidden scalar parameter. Let D_{ref} denote the complete Reference evidence and let $\mathcal{V}(D_{\mathrm{ref}})$ denote the mechanisms consistent with that evidence. For every Test x^{test}, a generated question must satisfy

$$\left|\left\{s(M, x^{\mathrm{test}}) : M \in \mathcal{V}(D_{\mathrm{ref}})\right\}\right| = 1. \tag{10}$$

Thus, several parameter settings may remain compatible with the References as long as they are equivalent for the predictions required by the question.

For O1, the governing measurement and comparison direction must be determined, while each profile-specific threshold may remain within an admissible interval. Every threshold compatible with that interval must agree on the four Test predictions.

For O2, the governing pairwise difference and comparison direction must be distinguished from the alternative derived relations. Exact threshold point identification is unnecessary when all Reference-compatible threshold values produce the same predictions on the three Tests.

For O3, the complete Reference evidence must identify a unique structural chain, including the input directions and logical composition through the two intermediate stages. Some numeric threshold settings can remain observationally equivalent because intermediate logical operations may mask distinctions that are irrelevant to the held-out predictions. All surviving numeric realizations must nevertheless agree on all six Test outcomes.

B.5 Structural Validation

Every generated Observation question is validated before model evaluation. The validator checks both solvability and whether the available evidence satisfies the intended information condition. Table 8 summarizes the family-specific requirements.

O3 additionally tests whether the intermediate states are necessary for the intended reconstruction. When the intermediate observations are removed and only the raw inputs and final outcomes are retained, multiple structural explanations must remain compatible with the References and must disagree on held-out predictions. The final six-Test set is also checked for component coverage so that the different input conditions and both logical gates affect the evaluated predictions.

B.6 Anti-Shortcut Construction

A valid Observation question should require reasoning from passive outcomes rather than exploiting artifacts of the generator. We therefore check whether the public input structure itself reveals the private role of a variable or mechanism.

Table 8: Structural validation requirements for Observation questions.

Family	Required identification	Validation requirement
O1	Governing measurement and comparison direction	The References must exclude alternative governing measurements and constrain both profile-specific thresholds sufficiently for all compatible thresholds to agree on the four Tests.
O2	Governing pairwise difference and comparison direction	The intended difference must explain the Reference outcomes, while individual raw measurements and alternative pairwise differences cannot provide an equivalent threshold rule. All compatible thresholds must agree on the three Tests.
O3	Multi-stage structural chain	The complete References must identify the ground-truth structural chain. Multiple compatible numeric threshold settings may remain, but every surviving mechanism must agree on all six Tests.

For O1, the visible numeric fields are constructed so that the governing measurement is not identifiable from simple marginal patterns such as unique-value counts, duplicate structure, or sorted numeric gaps. The Test layout likewise should not reveal which field is causal before the Reference outcomes are considered.

For O2, no individual raw measurement may reproduce the Reference labels through a valid threshold rule, and alternative pairwise differences are excluded as equivalent explanations. The intended rule must therefore be recovered from the relationship among measurements rather than from a simpler raw-field shortcut.

O3 uses a mechanism-independent sparse Reference and Test construction. Public numeric layouts are not selected according to the hidden logical gates or directions. In particular, question acceptance does not rely on mechanism-dependent rejection that could make the distribution of visible numeric inputs informative about the private structure. The intended signal comes from the Reference outcomes and intermediate states, not from the sampling pattern itself.

C Intervention Benchmark Details

C.1 Generator Specifications

The three Intervention families use different causal mechanisms, but all require the agent to learn from state changes produced by its own actions.

I1: Causal Driver. I1 contains eight controllable Valves. The generator assigns a private subset of two Valves as true causal drivers. Each driver has a hidden activation threshold, an activation side, and a signed effect. Non-driver Valves have no effect on the Gauge.

Let $\mathcal{D}$ denote the set of true drivers. The Gauge follows

$$P = P_0 + \sum_{i \in \mathcal{D}} \beta_i f_i(V_i; \tau_i), \qquad |\mathcal{D}| = 2, \tag{11}$$

where P_0 is the baseline, V_i is the level of Valve i, τ_i is its hidden threshold, and β_i is its signed effect. For a high-side driver, $f_i(V_i; \tau_i) = \max(0, V_i - \tau_i)$. For a low-side driver, $f_i(V_i; \tau_i) = \max(0, \tau_i - V_i)$.

A driver can therefore produce no observable change when it is tested outside its active region. Solving I1 requires identifying the causal Valves and learning where and how their effects become active.

I2: Causal Pathway. I2 contains eight Controllers. Every Controller affects a shared aggregate quantity, while a private subset of two Controllers also has a direct effect on the final outcome. The

mechanism is

$$S = \sum_i C_i, \qquad M_1 = M_0 + kS, \qquad M_2 = G(S) + \sum_i d_i C_i, \tag{12}$$

where C_i is Controller i, M_1 and M_2 are the two observable Meters, and G is a hidden nonlinear calibration. Exactly two coefficients d_i are nonzero.

Changing one Controller changes S, so the mediated and direct effects are mixed. A controlled comparison can instead change two Controllers while keeping S fixed. For example, changing C_i by $+\delta$ and C_j by $-\delta$ gives $\Delta S = 0$. The mediated term $G(S)$ then remains fixed, while the change in M_2 isolates the direct-effect contrast between the two Controllers.

I3: Causal Propagation. I3 contains five Relays connected by a hidden directed tree. Each Relay with outgoing edges has a private channel $c_i \in \{0, 1, 2, 3\}$. When Relay R_i is set to level x, the new value propagates to its children only when $x \bmod 4 = c_i$. The same signal value continues downstream, and each downstream Relay applies its own channel condition before propagating further.

The agent must recover both the direction of propagation and the hidden channel conditions. A missing downstream response can result from an inactive channel, while an observed downstream response can pass through several intermediate Relays.

C.2 Exploration-to-Transfer Construction

Every Intervention question contains two consecutive control stages.

Lock 1. Lock 1 is the exploration stage. The agent may repeatedly inspect and manipulate the system while trying to reach the first target. Each legal intervention returns the resulting observable state changes. Lock 1 has a horizon of 50 environment steps and no separate adjustment budget.

The first target is constructed together with the hidden mechanism. Reaching it requires interacting with the causal structure of the corresponding family rather than changing an unrelated control.

Lock 2. Lock 2 begins immediately after Lock 1 succeeds. The environment is not reset. The state reached at the end of Lock 1 becomes the initial state of Lock 2, and a new target is revealed only at this transition.

Lock 2 limits the number of additional state-changing adjustments. The private adjustment budgets are two for I1, two for I2, and five for I3. These budgets make the second stage a transfer problem: the agent must use information acquired during Lock 1 to reach a new target with limited further experimentation.

Family-specific transfer targets. In I1, both true drivers contribute to the generated targets. The Lock 2 target requires both driver settings to change relative to the Lock 1 solution.

In I2, the second target changes the required outcome while preserving the same causal pathway. The limited adjustment budget makes repeated exploration impractical.

In I3, the Lock 2 target assigns a new value to every Relay. Internal target values activate their hidden propagation channels. Setting an upstream Relay can therefore overwrite downstream Relays, so the target must be executed in an order consistent with the recovered propagation structure.

C.3 Intervention and Adjustment Semantics

The Intervention runtime distinguishes environment steps from effective adjustments. An environment step records a legal interaction with the environment. An effective adjustment is a legal control action that changes the selected control from its current value.

A state-changing control action consumes one adjustment. Any causal changes that propagate from that action do not consume additional adjustments. Reading or inspecting the environment also consumes no adjustment.

Table 9: Structural validation requirements for Intervention questions.

Family	Hidden structure	Validation requirement
I1	Threshold-gated causal drivers	The generated transition function must match the sampled driver identities, thresholds, activation sides, and signed effects. Passive evidence must leave multiple driver mechanisms compatible, while both control targets must be reachable through the true drivers.
I2	Shared mediated pathway with additional direct effects	The Reference state must remain compatible with multiple direct-effect structures. Controlled interventions that preserve the shared aggregate must distinguish these structures, and both Lock targets must be reachable under the sampled mechanism.
I3	Directed Relay tree with hidden propagation channels	Historical evidence must leave multiple graph–channel mechanisms compatible. Legal active interventions must be able to distinguish them. The Lock 1 target must preserve this ambiguity, while the Lock 2 target must require valid downstream control under the true propagation structure.

Setting a control to its current value is a full-state no-op. It consumes an environment step but no adjustment, and it does not trigger downstream propagation. This rule keeps the Lock 2 budget tied to actual state changes.

Only effective adjustments count toward the family-specific Lock 2 budget.

C.4 Family-Specific Target Construction

Targets are generated from the hidden mechanism and checked for reachability before a question is admitted.

For I1, the target is chosen so that both true drivers are relevant to the required Gauge value. Changing only one driver cannot reproduce the complete target configuration.

For I2, the target specifies the required observable Meter state but does not prescribe which Controller to test or how to construct a controlled comparison. The experiment design remains part of the agent's task.

For I3, the Lock 1 target is generated so that directly trying the visible target values does not identify the hidden graph and channels. The Lock 2 target has a different role. Its internal Relay values activate the corresponding channels, so successful control requires acting consistently with the directed propagation structure.

C.5 Structural Validation

Every Intervention question is validated before evaluation. Validation checks the hidden mechanism, the observable transition function, and the two-lock control problem.

The validator also checks transfer-specific constraints. In I1, the target construction requires both true drivers. In I2, the benchmark does not encode a prescribed matched intervention into the visible target. In I3, the Lock 2 target is reachable within five effective adjustments and requires an ordering consistent with the directed propagation structure.

C.6 Runtime Protocol

An Intervention episode begins in Lock 1 with the current state, the available controls, and the first target. The agent may inspect the environment and apply legal control actions. After each intervention, the environment returns the observable consequences of the resulting state transition.

Lock 1 succeeds when the current state satisfies its target. The episode then enters Lock 2 without resetting the environment. The second target is revealed at this point.

During Lock 2, every effective state-changing control consumes one adjustment from the private family-specific budget. Passive causal propagation does not consume extra adjustments. The episode succeeds when the actual environment state reaches the Lock 2 target. If the available adjustments are exhausted before the target is reached, the episode terminates as a Lock 2 failure.

The model-facing feedback contains observable environment consequences only. Hidden driver identities, pathway roles, graph structure, channels, and evaluator-side mechanism states are not exposed.

D Counterfactual Benchmark Details

D.1 Generator Specifications

All Counterfactual families begin from a completed factual episode whose final outcome is failure. The agent observes traces left by this factual history, infers the episode-specific conditions that caused the failure, and then revises the recorded past. The same hidden episode background is used when the revised history is replayed.

C1: Single Repair. C1 requires one historical revision. The generator constructs a factual failure and a legal set of historical edits such that exactly one single edit changes the replayed outcome to success.

The agent must first infer which factual condition made the observed failure possible. It then selects one past object and one action that was legally available at that historical step. The first valid state-changing revision is committed and the episode is replayed once.

C2: Joint Repair. C2 requires two jointly necessary historical revisions. Every legal single edit fails. Exactly one compatible unordered pair succeeds.

The two edits are independent of ordering. Neither edit becomes legal only because the other has already been applied. The agent therefore selects both revisions from the factual evidence rather than using the first revision as a counterfactual probe.

C3: Cascade Repair. C3 requires one early historical revision whose consequences propagate through a multi-stage factual trajectory. The factual causal chain contains four to six downstream hops. The successful edit occurs in the earlier part of this chain.

After the revision, the downstream trajectory is recomputed from the first point of divergence. The successful replay changes at least three later states or events before reaching the final outcome. A late direct repair does not provide an alternative successful solution.

D.2 Factual Scene Evidence and Abduction

Each question opens in the aftermath of a factual failed episode. The current Scene is read-only. Objects expose their basic descriptions, while their factual traces are revealed only after the agent legally inspects them.

The evidence is distributed across multiple Scene objects. A trace may describe a physical state, a residual mark, a device condition, an object displacement, or another consequence of the completed episode. The agent combines these observations to infer the hidden factual conditions that are relevant to the repair.

The full private hidden world does not need to be uniquely recovered. The collected evidence must determine the repair-relevant hidden state strongly enough that the corresponding successful historical revision is unique across the hidden worlds that remain compatible with the evidence.

D.3 Historical Action Space and Replay

Historical revisions operate on actions that were legally available during the factual episode. The agent does not directly assign a desired private state. It selects a historical object and chooses one of the actions that object could have taken at the corresponding time.

Table 10: Distractor roles in Counterfactual questions.

Role	Scene evidence	Reasoning effect
Neutral	A factual trace unrelated to the repair	Adds scene content without supporting a competing causal explanation.
Plausible anomaly	A locally unusual trace	Suggests a possible cause but does not form a complete alternative explanation.
Historical ambiguity	A trace compatible with more than one past event	Requires the agent to combine it with evidence from other objects.
Coherent false cause	Several traces jointly support an alternative factual explanation	The alternative remains possible under partial evidence and implies a different historical repair.

Let τ^{F} denote the factual action history and let ρ denote the selected historical revision. Replay replaces the specified factual action or actions while keeping the same episode-specific background. The simulator then recomputes every downstream state and event affected by the revision.

For C1 and C3, the first valid state-changing historical edit is committed immediately. The revised episode is replayed once and the resulting Door state is terminal.

For C2, the first edit is staged without replay feedback. The second edit is then selected from the remaining historical candidates. Both revisions are applied jointly after submission, followed by one replay of the episode.

D.4 Competing Factual Explanations

Counterfactual questions combine solution-relevant evidence with structured distractors. The distractors play four roles, summarized in Table 10.

A coherent false-cause bundle is tied to an alternative hidden world represented by the same deterministic simulator. Under the bundle and an incomplete subset of the relevant evidence, this alternative world remains compatible with the observed Scene and leads to a different repair.

The complete Scene also contains evidence that rules out this alternative. Restoring the full factual evidence therefore removes the false explanation and leaves the intended repair uniquely supported. The false-cause construction makes abduction depend on combining evidence across objects rather than following a single anomalous trace.

D.5 One-Shot Runtime Protocol

A Counterfactual episode has two phases.

During *scene exploration*, the agent may inspect the read-only current Scene. Each successful inspection reveals the factual observation attached to that object. The agent chooses when to stop collecting evidence and enter historical replay.

All factual evidence obtained through legal inspection remains available after this transition. The current Scene is then frozen, and the agent interacts with the recorded historical objects and their legal past actions.

For C1 and C3, one valid state-changing historical revision commits the answer. The simulator performs one deterministic replay and immediately terminates the episode.

For C2, the first revision is staged without counterfactual feedback. The second revision is also staged without feedback. The pair is then submitted jointly, applied to the same factual history, and replayed once.

A replay that opens the Door is a success. A valid committed repair whose replay leaves the Door closed terminates the episode as a counterfactual failure. No family permits repeated counterfactual replay followed by answer revision.

Table 11: Family-specific validation requirements for Counterfactual questions.

Family	Required repair	Validation requirement
C1	One historical edit	Exactly one legal single revision produces a successful replay. Every other legal single revision leaves the factual failure unresolved.
C2	One unordered pair of historical edits	Every legal single revision fails. Exactly one compatible unordered pair succeeds, and applying the two revisions in either order produces the same counterfactual replay.
C3	One early historical edit	Exactly one early revision produces success. The factual chain contains four to six causal hops, the successful revision changes at least three downstream states or events, and no late-stage revision provides a successful shortcut.

D.6 Structural Validation

Every Counterfactual question is validated before evaluation. The factual simulation must end with the Door closed, and each candidate historical action must be legal at its recorded point in the factual episode.

The complete factual evidence must identify the repair-relevant state and the successful repair. Hidden worlds that remain compatible with the complete evidence must agree on the repair required by the question. No single Scene observation is admitted as a complete shortcut to that repair.

Coherent false-cause bundles are validated separately. The corresponding alternative hidden world must remain compatible with the intended partial evidence and must imply a repair different from the ground-truth repair. Evidence included in the complete Scene must then eliminate that alternative and restore a unique supported repair.

E Generator Capacity

Generator capacity measures the number of distinct valid task instances supported by each implemented generator family after collapsing surface-only variations. We count instances separately when they differ in task-relevant mechanism, target, evidence, or interaction structure. Changes only to identifiers, presentation order, or verbalization do not define a new task instance. The resulting counts characterize the space supported by the generators rather than the number of questions materialized for evaluation.

Table 12 reports the family-level capacities. For families whose complete constrained spaces can be enumerated under the implemented generator and validation rules, we report rounded capacities. For O2 and I2, the full constrained space is not exhaustively enumerated, so we report conservative lower bounds. Lower-bound values are rounded downward and therefore never exceed the verified counts.

Aggregating across the three families in each regime gives approximately 1.3×10^{23} instances for Association and 7.1×10^{7} for Counterfactual, with verified lower bounds of 9.69×10^{14} for Observation and 3.87×10^{20} for Intervention.

Benchmark release. We will release the exact 120 questions used in our evaluation, comprising 10 validated questions from each of the twelve generator families. We will also provide an interface for generating and validating additional questions from the same families.

Table 12: Generator capacity of the twelve benchmark families. $\geq$ denotes a verified conservative lower bound.

Generator family	Capacity
A1 Single-Hop Association	2.2×10^{10}
A2 Two-Hop Association	2.8×10^{17}
A3 Multi-Hop Association	1.3×10^{23}
O1 Conditional Threshold	9.7×10^{14}
O2 Difference Threshold	$\geq 1.53 \times 10^{6}$
O3 Mechanism Chain	1.3×10^{12}
I1 Causal Driver	4.9×10^{11}
I2 Causal Pathway	$\geq 2.05 \times 10^{11}$
I3 Causal Propagation	3.9×10^{20}
C1 Single Repair	8.2×10^{6}
C2 Joint Repair	7.3×10^{6}
C3 Cascade Repair	5.6×10^{7}

F Representative Question Instances

F.1 Association Examples

A1: Single-Hop Association (`A1_000000`). Model-facing evidence. The Note specifies the ordered colors `orange`, `green`, `pink`, and `purple`. Inspecting the corresponding numbered Blocks reveals

$$\texttt{orange} \to 5, \quad \texttt{green} \to 6, \quad \texttt{pink} \to 4, \quad \texttt{purple} \to 7.$$

The pink Block is inside the Cabinet and becomes visible only after the Cabinet is opened. The room also contains ordinary objects with matching colors and numbered Blocks with off-reference colors.

Ground-truth solution. Following the Note order gives the code

$$\boxed{5647}.$$

A2: Two-Hop Association (`A2_000001`). Model-facing evidence. The Panel specifies `crosshatch`, `checkerboard`, `chevrons`, and `concentric_rings`. The valid two-hop chains are

$$\begin{aligned} \texttt{crosshatch} &\to \texttt{triangle} \to 1, \\ \texttt{checkerboard} &\to \texttt{square} \to 5, \\ \texttt{chevrons} &\to \texttt{circle} \to 7, \\ \texttt{concentric_rings} &\to \texttt{star} \to 9. \end{aligned}$$

The `chevrons`→`circle` Box is hidden inside the Cabinet. Other Boxes create partial matches. For example, another `crosshatch` Box points to `arrowhead`, while off-reference markers point to shapes that also appear on numbered Buttons.

Ground-truth solution. Composing marker→shape→digit in Panel order gives

$$\boxed{1579}.$$

A3: Multi-Hop Association (`A3_000000`). Model-facing evidence. The Board specifies `triangle`, `diamond`, `circle`, and `star`. The four valid chains are

$$\begin{aligned} \texttt{triangle} &\to \texttt{stone} \to \texttt{red} \to \texttt{waves} \to 1, \\ \texttt{diamond} &\to \texttt{wood} \to \texttt{green} \to \texttt{chevrons} \to 9, \\ \texttt{circle} &\to \texttt{glass} \to \texttt{yellow} \to \texttt{vertical_stripes} \to 5, \\ \texttt{star} &\to \texttt{metal} \to \texttt{blue} \to \texttt{crosshatch} \to 7. \end{aligned}$$

Several objects agree with only part of a chain. For example, a second diamond Token maps to `leather`, another glass Holder maps to `purple`, and another blue Display maps to `concentric_rings`. These alternatives do not complete a valid path to a digit.

Ground-truth solution. Following the Board order through all four relations gives

$$\boxed{1957}.$$

F.2 Observation Examples

O1: Conditional Threshold (`O1_000000`). Model-facing References.

	Profile	Internal pressure	Seal strength	Release setting / Outcome
R1	D	85.2	50.6	34.7 / released
R2	D	80.6	69.3	42.3 / sealed
R3	A	69.1	57.4	49.9 / sealed
R4	A	94.4	55.7	55.6 / sealed
R5	A	80.6	59.1	44.2 / released
R6	D	87.5	59.1	40.4 / released
R7	A	78.3	62.5	48.0 / released
R8	D	76.0	64.2	44.2 / sealed

The four held-out Tests are:

	Profile	Internal pressure	Seal strength	Release setting
T1	D	71.4	52.3	36.6
T2	A	89.8	65.9	51.8
T3	A	82.9	60.8	46.1
T4	D	82.9	60.8	46.1

Ground-truth mechanism and predictions. The governing variable is `release_setting`, with larger values producing `sealed`. The References constrain the threshold to $(48.0, 49.9]$ for Profile A and $(40.4, 42.3]$ for Profile D. The Test predictions are

$$(\texttt{released}, \texttt{sealed}, \texttt{released}, \texttt{sealed}).$$

T3 and T4 have identical numeric inputs and differ only in profile, so their opposite outcomes require a profile-conditioned threshold.

O2: Difference Threshold (`O2_000000`). Model-facing References.

	Chamber	Reference	Buffer	Outcome
R1	39.1	59.9	58.6	tripped
R2	62.5	74.2	70.3	tripped
R3	84.6	101.5	102.8	stable
R4	87.2	93.7	100.2	stable
R5	30.0	43.0	36.5	tripped
R6	31.3	46.9	46.9	stable
R7	58.6	67.7	72.9	stable
R8	85.9	100.2	95.0	tripped

The Tests are

	Chamber	Reference	Buffer
$T1$	49.5	69.0	66.4
$T2$	96.3	104.1	108.0
$T3$	70.3	88.5	80.7

Ground-truth mechanism and predictions. The governing quantity is

$$\texttt{reference_pressure} - \texttt{buffer_pressure}.$$

Larger differences produce `tripped`, and the Reference evidence places the transition between 0.0 and 1.3. The three Test differences are 2.6, -3.9, and 7.8, giving

$$(\texttt{tripped}, \texttt{stable}, \texttt{tripped}).$$

The individual pressure values do not define the rule; the agent must first construct the relevant difference.

O3: Mechanism Chain (`O3_000001`). Model-facing References.

	Thermal	Pressure	Flow	Stage 1	Stage 2	Final
R1	36.5	58.0	5.8	ON	ON	OFF
R2	24.5	60.0	8.2	OFF	OFF	OFF
R3	29.0	64.0	5.0	OFF	OFF	OFF
R4	27.5	68.0	20.2	OFF	OFF	OFF
R5	32.0	62.0	7.4	ON	ON	OFF
R6	35.0	70.0	21.0	ON	OFF	OFF
R7	30.5	66.0	18.6	ON	OFF	OFF
R8	23.0	72.0	17.8	OFF	OFF	OFF

All eight Reference outcomes are `OFF`. The intermediate states nevertheless distinguish the hidden chain.

Ground-truth mechanism. Thermal load is active on its high side, line pressure on its low side, and flow rate on its high side. Both hidden gates are AND:

$$Z_1 = C_{\text{thermal}}, \qquad Z_2 = Z_1 \wedge C_{\text{pressure}}, \qquad Y = Z_2 \wedge C_{\text{flow}}.$$

The six Test predictions are

$$(\texttt{OFF},\ \texttt{OFF},\ \texttt{OFF},\ \texttt{OFF},\ \texttt{ON},\ \texttt{OFF}).$$

In particular, T5 has high thermal load, low line pressure, and high flow, so all three conditions are active. Predicting from the Reference final outcomes alone would miss this case.

F.3 Intervention Examples

I1: Causal Driver (`I1_000000`). Model-facing setup. Initially all eight Valves are at level 7 and the Gauge reads 56.8125. The historical Operation Log contains synchronized settings:

Record	All Valve levels	Gauge
1	6	59.25
2	8	53.5625
3	7	56.8125
4	5	59.25

The Lock 1 target is

$$\texttt{Gauge_1} = 44.125.$$

Ground-truth mechanism. The causal drivers are Valve 5 and Valve 6. Valve 5 is a low-side driver with threshold 2.25 and signed effect -2.75. Valve 6 is a high-side driver with threshold 6.25 and signed effect -3.25.

The synchronized Operation Log remains compatible with 4,704 hidden mechanisms. A local probe of Valve 5 around levels 5–8 produces no effect because its active region lies below 2.25. The Lock 1 target is reached at

$$V_5 = 0, \qquad V_6 = 9.$$

After Lock 1, the second Gauge target is 56.125, reached by

$$V_5 = 2, \qquad V_6 = 7,$$

which requires changing both causal drivers.

I2: Causal Pathway (`I2_000000`). Model-facing setup. The initial Controller state is

$$(2, 3, 8, 2, 2, 9, 1, 3),$$

with

$$M_1 = 91.25, \qquad M_2 = 127.00.$$

The Lock 1 target is

$$M_1 = 91.25, \qquad M_2 = 124.75.$$

The Operation Log contains two synchronized configurations: setting every Controller to 3 gives $(M_1, M_2) = (77.75, 108.75)$, while setting every Controller to 4 gives $(95.75, 124.75)$.

Ground-truth mechanism. All eight Controllers affect the shared sum

$$S = \sum_i C_i,$$

which changes both Meters through the mediated pathway. Controller 3 and Controller 7 also have direct effects on M_2, with coefficients 2.75 and 1.75.

A matched change such as

$$C_3 \leftarrow C_3 + 1, \qquad C_7 \leftarrow C_7 - 1$$

keeps $\Delta S = 0$. The shared mediated term is therefore unchanged, while

$$\Delta M_2 = 2.75 - 1.75 = 1.$$

This controlled comparison separates the direct pathway from the shared mediated response.

One Lock 1 solution has

$$C_3 = 4, \qquad C_7 = 6,$$

with $M_1 = 91.25$ and $M_2 = 124.75$. Lock 2 keeps M_1 fixed and changes the target to $M_2 = 127.75$. It is reached with

$$C_3 = 7, \qquad C_7 = 3,$$

using the two available state-changing adjustments.

I3: Causal Propagation (`I3_000000`). Model-facing setup. The two historical interventions are

$$R_4 : 6 \to 10$$

with no change to the other Relays, and

$$R_3 : 7 \to 4$$

again with no downstream change. The initial state is

$$(R_1, R_2, R_3, R_4, R_5) = (8, 0, 9, 11, 10),$$

and the Lock 1 target is

$$(9, 6, 8, 2, 0).$$

Ground-truth mechanism. The directed tree is

$$R_5 \to R_2 \to R_4 \to R_1, \qquad R_3 \to R_4.$$

The hidden channels are

$$c_2 = 1, \qquad c_3 = 1, \qquad c_4 = 0, \qquad c_5 = 0,$$

with R_1 a leaf.

The two historical interventions fail to propagate because $10 \bmod 4 \neq c_4$ and $4 \bmod 4 \neq c_3$. Thus, absence of propagation in the log does not identify the absence of an edge.

The Lock 2 target is

$$(R_1, R_2, R_3, R_4, R_5) = (6, 1, 9, 4, 8).$$

Its internal Relay values activate their corresponding channels. The five settings must therefore respect a valid causal order. The three valid orders are

$$R_3, R_5, R_2, R_4, R_1,$$
$$R_5, R_2, R_3, R_4, R_1,$$
$$R_5, R_3, R_2, R_4, R_1.$$

F.4 COUNTERFACTUAL EXAMPLES

C1: Single Repair (`C1_000000`). Model-facing aftermath. The access terminal records that the unlock command was accepted. The branch power is available, and the unlock actuator reports a normal energized response. The latch nevertheless remains extended. The visible latch guide is clean, and the door track is clear.

Other Scene traces support two plausible alternatives: an actuator failure and a jammed latch guide. The corresponding direct checks contradict both explanations.

Ground-truth abduction and repair. The actuator is working, the latch guide is free, the track is clear, and the command has been accepted. The failed transmission is the disconnected actuator-to-latch coupling.

Among the legal historical revisions, the unique successful edit is

$$\boxed{\texttt{SECURE_COUPLING}}.$$

After replay, the actuator moves, the latch retracts, and the Door opens.

C2: Joint Repair (`C2_000006`). Model-facing aftermath. The factual unlock succeeds, but the door branch has no power and the door never starts moving. The motor passes its stand-alone self-test. A service cart has rolled approximately 160 cm toward the Door, and the parking area slopes in that direction. The fixed door rail itself is clear.

Ground-truth abduction and repair. Restoring branch power solves the first failure but creates a second one. With the cart brake still released, the cart rolls into the Door path. The powered Door then moves only partway before the obstacle stops the opening sequence.

No single historical edit succeeds. The unique successful pair is

$$\boxed{\texttt{RESET_DOOR_BREAKER} + \texttt{ENGAGE_BRAKE}}.$$

With both revisions applied, branch power is available, the cart remains in its parking position, and the Door completes its full travel.

C3: Cascade Repair (`C3_000001`). Model-facing aftermath. The access request is accepted and the door-drive self-test passes. An upstream source continues discharging water during the incident. The drain-side temperature is approximately $-10°$C. Crystalline material bridges the first drain slots, the drain-flow sensor remains near baseline despite continuing inflow, standing water reaches approximately 10 cm, and the float indicator rises above its upper reference line.

The Scene also contains evidence compatible with a debris-blocked drain or a failed door drive. Direct observations of the frozen inlet and the successful drive self-test distinguish these alternatives from the factual mechanism.

Ground-truth abduction and repair. The factual causal trajectory is

$$\begin{aligned}\text{frozen drain} &\to \text{blocked drainage} \to \text{high water} \\ &\to \text{high float} \to \text{latched flood safety} \to \text{Door failure}.\end{aligned}$$

The unique successful historical revision is the early action

$$\boxed{\texttt{START_DRAIN_HEATER}}.$$

Replaying the episode from that change produces

$$\begin{aligned}\text{clear drain} &\to \text{flowing drainage} \to \text{normal water level} \\ &\to \text{normal float} \to \text{released flood safety} \to \text{Door OPEN}.\end{aligned}$$

All six downstream events change after the historical revision.

G AGENT EVALUATION DETAILS

G.1 AGENT INTERFACE AND ACTION SCHEMAS

Each episode proceeds through structured interactions with the environment. The model receives the current model-facing state and the choices available at that point. It returns one structured decision, which is checked against the current interface before execution.

Association, Observation, and Intervention use an object-first interaction pattern. The model first selects one visible object. If the object exposes local actions, it then selects one legal action for that object. Object-selection decisions contain an object identifier and a short reason. Local-action decisions contain an action type, its parameters, and a reason.

Observation adds two phase-specific decisions. During the Reference phase, the model may stop inspecting References and call `reveal_tests`. After all Tests are revealed, it calls `submit_predictions` once with predictions for every Test.

Intervention uses the same object–action structure for inspection and control. During Lock 1, both inspection and state-changing control actions may be available. Lock 2 exposes only the task controls needed to reach the new target.

Counterfactual uses schema-checked JSON decisions. During Scene exploration, the model may inspect one current object or choose `ENTER_HISTORICAL_REPLAY`. Historical replay then exposes recorded historical objects and the actions that were legal at those points in the factual episode. C1 and C3 commit one historical revision. C2 stages two revisions before calling `submit_joint_counterfactual`.

Regime	Interactive interface	Final decision
Association	Object selection and local interaction	One code submission
Observation	Reference inspection, followed by Test reveal	One joint prediction over all Tests
Intervention	Inspection and state-changing control	Reach the Lock 2 target
Counterfactual	Read-only Scene inspection, followed by historical revision	Commit the required revision(s) and replay once

G.2 PROMPT TEMPLATES

Each regime uses a fixed system-level instruction. The runtime then constructs the user message from the current environment state, interaction context, and currently legal choices.

Association. The system instruction is:

> You are an LLM agent solving a causal escape room puzzle.
> Your objective is to open the door. A code may be required.
> You may submit a code only once in the episode.
> An incorrect code submission ends the episode.
> Each step has two phases: first choose one visible object to inspect, then choose one local action for that object if actions are available.
> During object selection, call `select_object` with exactly one `object_id` from the visible object list and a non-empty reason.
> During action selection, call `execute_action` with `action_type`, `params`, and a non-empty reason, using only the listed local actions.
> Do not fabricate object IDs, action types, or parameter names.

Each object-selection turn provides the goal, current observation, previous environment feedback, factual interaction history, progress memory, and visible object IDs. After an object is selected, the next message provides its revealed observation and legal local actions.

Observation. The system instruction is:

> You are an LLM agent solving an escape room puzzle.
> Use only information actually revealed by the environment.
> Follow the current tool protocol exactly.
> Do not assume hidden attributes or outcomes that have not been shown.

During the Reference phase, the user message contains the current observation, visible object IDs, factual Reference-phase history, progress memory, and previous environment feedback. The agent may inspect another object or reveal the Tests. After the Tests are revealed, all Test inputs and prediction options are shown together.

Intervention. The system instruction is:

> You are an LLM agent solving an escape room puzzle.
> Your objective is to open the door.
> The puzzle has Lock 1 followed by Lock 2. Lock 2 starts only after Lock 1 is completed; the real environment state is preserved and the new Lock 2 target is revealed only then.
> During Lock 1, use the currently available room interactions and control-setting actions.
> During Lock 2, ordinary exploration is unavailable: only state-changing `set_level` actions on the visible task controls are available. State-changing adjustments are limited. The exact adjustment limit and remaining number of adjustments are not shown.
> Each interaction cycle has two phases: first choose one visible object, then choose one local action if actions are available.
> During object selection, call `select_object` with exactly one visible `object_id` and a non-empty reason.
> During action selection, call `execute_action` with one listed `action_type` and `params`, together with a non-empty reason.
> Do not fabricate object IDs, action types, parameter names, or values.

Each turn provides the current lock state, current target, factual interaction history, progress memory, visible objects, and the legal actions associated with the selected object.

Counterfactual. The Counterfactual instruction depends on the public historical-edit budget B:

> You are acting inside a room-escape benchmark. Follow only the actions and objects exposed by the interface.
> The current scene is read-only: current-scene interactions may inspect evidence but may not change current state.
> This episode permits exactly B historical revision(s).
> A committed counterfactual is replayed once; after commitment the episode ends.
> Do not invent object IDs or actions. Return only the JSON object requested by the current interface.

For C2, $B = 2$, and the prompt additionally states that both revisions must be staged before the single joint submission and that no counterfactual outcome is shown between them.

Scene-exploration messages contain the current Scene, task instruction, discovered observations, recent factual interaction history, and the control for entering historical replay. Historical-replay messages contain the remaining historical objects, staged revisions, recent interaction history, and all factual Scene evidence obtained before the phase transition.

G.3 Interaction Context and Memory

Association, Observation, and Intervention retain the factual interaction history accumulated during the relevant interactive phase. A history entry records the selected object, the observation revealed for that object, the executed action and parameters, the resulting environment feedback, and whether the action succeeded. Earlier model-generated reasons are not included in this factual history.

These three regimes also provide a neutral progress summary over the most recent 15 interaction cycles. The summary records the step, selected object, selected action, and execution status. It does not add new environment information.

Observation retains Reference-phase interactions in its factual history. Once the Tests are revealed, the agent cannot inspect additional objects or acquire further environment observations.

Counterfactual keeps the most recent 15 real interactions in its raw history. Each entry records the phase, selected object, revealed observation, action, environment feedback, and the number of staged revisions.

Counterfactual also preserves all factual Scene evidence obtained through legal inspection. When the agent enters historical replay, this evidence is carried forward as a persistent Scene-evidence ledger. The ledger stores the inspected object and its revealed observation, so evidence obtained early in Scene exploration remains available after older interaction records leave the 15-interaction window.

G.4 Regime-Specific Runtime Protocols

Association. The 50-step horizon applies to the full episode. The agent may continue inspecting and interacting with objects until it submits a code or reaches the horizon. The first code submission is final, and an incorrect code terminates the episode.

Observation. The 50-step horizon applies to the Reference phase. The agent may call `reveal_tests` before reaching the horizon. If the horizon is reached first, the Tests are revealed automatically without consuming an additional environment step. All Tests are revealed together. No further environment interaction is allowed, and the first complete schema-valid joint submission is final.

Intervention. The 50-step horizon applies to Lock 1. Reaching the Lock 1 target transitions the episode to Lock 2 while preserving the actual environment state. The Lock 2 target is revealed only after this transition.

Lock 2 is constrained by effective state-changing adjustments. The budgets are two adjustments for I1, two for I2, and five for I3. Passive state changes caused by a control action do not consume additional adjustments.

Counterfactual. The 50-step horizon covers the full episode, including Scene exploration and historical replay. Entering historical replay freezes the current Scene and exposes the historical action space.

C1 and C3 terminate after one valid state-changing historical revision is committed and replayed. C2 stages two revisions without intermediate counterfactual feedback. The pair is then submitted jointly and replayed once.

G.5 Experimental Configuration

Evaluation-set instantiation. For each generator family, we deterministically instantiate the ten evaluation questions indexed 000000–000009 using base seed 42. The instantiated questions are checked by the corresponding family validators, after which the resulting 120-question evaluation set is frozen and reused unchanged across all model evaluations.

The corresponding provider-facing model identifiers are `gpt-5.6-sol`, `claude-opus-5`, `grok-4.6`, `gemini-3.8-flash`, `deepseek-v4.1-flash`, `kimi-k3`, and `qwen3.8-2.4t-a95b`.

Claude Opus 5 is evaluated on the 90 Association, Observation, and Intervention questions. Its 30 Counterfactual evaluations are unavailable because the provider-side safety mechanism blocks execution of the C1–C3 interaction protocol.

We request temperature 0 where the endpoint supports it. No explicit reasoning or thinking-effort parameter is sent, and each provider retains its default output-token limit.

Each logical API request has a timeout of 900 seconds. An infrastructure failure may trigger at most one additional request. A malformed or illegal structured response may trigger up to five correction attempts after the initial response. Correction attempts do not advance the environment or reveal additional evidence.

Setting	Value
Temperature	0 where effective
Reasoning / thinking	Provider default
Output-token limit	Provider default
Request timeout	900 s
Infrastructure retry	At most 1 after the initial request
Format / action correction	At most 5 after the initial response
Recent progress / history window	15 interaction cycles or interactions
Environment horizon	50 steps

Each question is run as an independent episode. The 50-step horizon follows the regime-specific scope defined in Section G.4.

H Diagnostic Metric Definitions

H.1 Association Diagnostics

PIB Exceed Rate. For episode e, let $U(e)$ denote the number of PIB-counted puzzle-evidence selections and let $B(q_e)$ denote the PIB allowance for question q_e. We define

$$z_{\mathrm{PIB}}(e) = \mathbb{I}\left[U(e) > B(q_e)\right]. \tag{13}$$

The PIB Exceed Rate is

$$\text{PIB Exceed Rate} = \frac{1}{N}\sum_{e=1}^{N} z_{\mathrm{PIB}}(e). \tag{14}$$

A higher PIB Exceed Rate indicates that the agent more often uses more puzzle-evidence selections than the question-specific allowance.

Outcome Decomposition. We decompose Association episodes into the following mutually exclusive terminal outcomes:

- **Within-PIB Success**: the submitted code is correct and $U(e) \leq B(q_e)$.
- **Success over PIB**: the submitted code is correct and $U(e) > B(q_e)$.
- **Wrong-Code Failure**: the agent submits an incorrect final code.
- **Horizon Timeout**: the episode reaches the environment horizon without a correct terminal solution.
- **Other Failure**: a valid benchmark failure not covered by the outcomes above.

For outcome category c, we report

$$\text{Outcome}_c = \frac{N_c}{N}, \tag{15}$$

where N_c is the number of episodes assigned to category c. The outcome categories sum to one within each reporting scope. PIB Exceed Rate is computed separately from this terminal decomposition.

H.2 Observation Diagnostics

Test-Level Accuracy. Let $\mathcal{J}$ denote the set of episodes with a valid final joint submission. For episode $e \in \mathcal{J}$, let $\mathcal{T}_e$ denote its Tests, and let $\hat{y}_{e,t}$ and $y_{e,t}$ denote the predicted and ground-truth outcomes for Test t.

Test-Level Accuracy is the micro-average over all valid Test predictions:

$$\text{Test-Level Accuracy} = \frac{\sum_{e\in\mathcal{J}}\sum_{t\in\mathcal{T}_e}\mathbb{I}\left[\hat{y}_{e,t}=y_{e,t}\right]}{\sum_{e\in\mathcal{J}}|\mathcal{T}_e|}. \tag{16}$$

Episodes without a valid final joint submission are excluded from Test-Level Accuracy and are reported separately in the Joint Error Decomposition.

Joint Error Decomposition. For a valid joint submission in episode e, let

$$E(e)=\sum_{t\in\mathcal{T}_e}\mathbb{I}\left[\hat{y}_{e,t}\neq y_{e,t}\right] \tag{17}$$

denote the number of incorrect Test predictions.

Each Observation episode is assigned to one of the following categories:

- **E0**: $E(e)=0$.
- **E1**: $E(e)=1$.
- **E2**: $E(e)=2$.
- **E3+**: $E(e)\geq 3$.
- **Invalid Submission**: the episode ends without a valid final joint submission.

For category c, we report

$$\text{JointError}_c=\frac{N_c}{N}, \tag{18}$$

where N_c is the number of episodes assigned to category c. The **E0** category is identical to Observation official success.

H.3 Intervention Diagnostics

Lock 1 Success Rate. For episode e, let

$$L_1(e)=\mathbb{I}\left[\text{Lock 1 succeeds}\right]. \tag{19}$$

The Lock 1 Success Rate is

$$\text{Lock1 Success}=\frac{1}{N}\sum_{e=1}^{N}L_1(e). \tag{20}$$

This metric records whether the agent reaches the first target during the exploration stage.

Lock 2 Success Rate conditioned on Lock 1. Let

$$L_2(e)=\mathbb{I}\left[\text{Lock 2 succeeds within its adjustment budget}\right]. \tag{21}$$

The conditional Lock 2 success rate is

$$\text{Lock2 Success} \mid \text{Lock1}=\frac{\sum_{e=1}^{N}L_1(e)L_2(e)}{\sum_{e=1}^{N}L_1(e)}. \tag{22}$$

Only episodes that successfully complete Lock 1 enter the denominator. If no episode in a reporting scope completes Lock 1, the conditional Lock 2 success rate is reported as N/A.

Stage Outcome Decomposition. Each Intervention episode is assigned to one of the following mutually exclusive outcomes:

- **Both Success**: Lock 1 and Lock 2 both succeed.
- **Lock 1 Failure**: Lock 1 does not succeed.
- **Lock 2 Failure**: Lock 1 succeeds, but Lock 2 does not succeed within its adjustment budget.
- **Other Failure**: a valid terminal failure not covered by the stage outcomes above.

For outcome category c, we report

$$\mathrm{StageOutcome}_c = \frac{N_c}{N}, \tag{23}$$

where N_c is the number of episodes assigned to category c. **Both Success** is identical to Intervention official success.

H.4 Counterfactual Diagnostics

Relevant Evidence Coverage. For episode e, let $\mathcal{E}_e$ denote the set of annotated repair-relevant evidence items and let $\mathcal{E}_e^{\mathrm{obs}}$ denote the subset acquired by the agent before committing its historical revision. The episode-level coverage is

$$\mathrm{Coverage}(e) = \frac{|\mathcal{E}_e^{\mathrm{obs}}|}{|\mathcal{E}_e|}. \tag{24}$$

Relevant Evidence Coverage is the episode-level mean

$$\text{Relevant Evidence Coverage} = \frac{1}{N}\sum_{e=1}^{N} \mathrm{Coverage}(e). \tag{25}$$

This metric measures how much annotated repair-relevant evidence the agent acquires before committing a historical revision.

Partial Edit Accuracy. For C1 and C3, let $m_o(e)$, $m_a(e)$, and $m_t(e)$ indicate whether the submitted object, action, and historical time match their oracle counterparts. For C2, let $m_2(e)$ denote the number of matched oracle edits, with edit order ignored.

The episode-level score is

$$\mathrm{EditAcc}(e) = \begin{cases} \dfrac{m_o(e) + m_a(e) + m_t(e)}{3}, & e \in \mathrm{C1} \cup \mathrm{C3}, \\ \dfrac{m_2(e)}{2}, & e \in \mathrm{C2}. \end{cases} \tag{26}$$

Partial Edit Accuracy is

$$\text{Partial Edit Accuracy} = \frac{1}{N}\sum_{e=1}^{N} \mathrm{EditAcc}(e). \tag{27}$$

This metric measures how closely the submitted historical revision matches the oracle repair at the edit-component level.

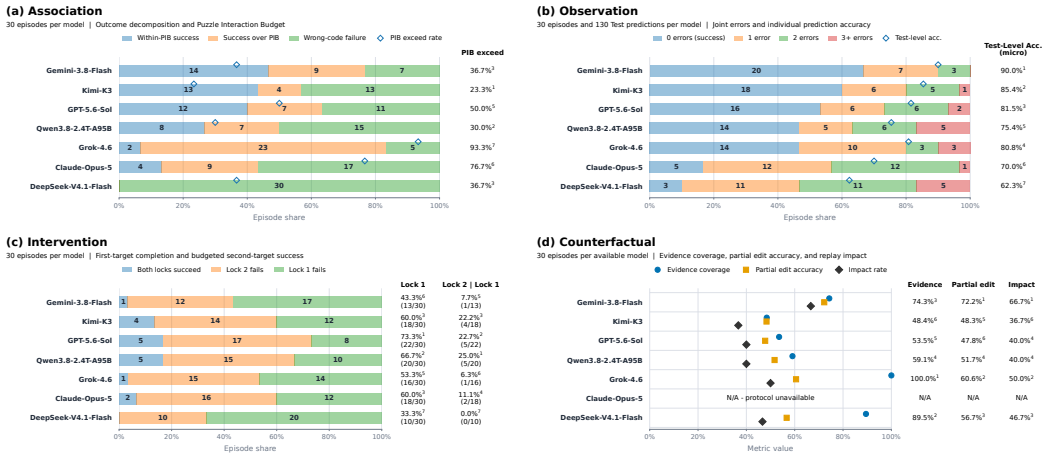


Figure 3: **Different evidence conditions expose different failure modes.** (a) Association decomposes terminal outcomes and reports the PIB Exceed Rate. (b) Observation decomposes joint prediction errors and reports micro-averaged Test-Level Accuracy. (c) Intervention separates Lock 1 completion from Lock 2 success conditioned on Lock 1. (d) Counterfactual reports Relevant Evidence Coverage, Partial Edit Accuracy, and Counterfactual Impact Rate. Numbers inside stacked bars denote episode counts. Superscripts indicate within-metric model ranks; lower is better for PIB Exceed Rate and higher is better for the other metrics. Claude Opus 5 Counterfactual results are unavailable.

Counterfactual Impact Rate. Let Δ_e denote the number of downstream replay events changed by the submitted historical revision. We define

$$I_{\mathrm{cf}}(e) = \mathbb{I}\left[\Delta_e > 0\right]. \tag{28}$$

The Counterfactual Impact Rate is

$$\text{Counterfactual Impact Rate} = \frac{1}{N}\sum_{e=1}^{N} I_{\mathrm{cf}}(e). \tag{29}$$

This metric measures how often the submitted historical revision changes at least one downstream event in the counterfactual replay.

I Regime-Specific Failure Diagnostics

Figure 3 decomposes final success and failure into regime-specific stages. We focus on the resulting behavioral patterns here; complete family-level breakdowns are provided in Appendix J.

Correct solutions can hide inefficient evidence search. Grok-4.6 succeeds within the PIB on only $2/30$ Association episodes, but recovers the correct code on another $23/30$ after exceeding the budget. In contrast, DeepSeek-V4.1-Flash ends all 30 episodes with an incorrect code. The same official failure can therefore reflect two different problems: the agent may eventually recover the solution but search too broadly, or it may fail to recover the solution at all.

High Test-level accuracy can coexist with joint prediction failure. Gemini-3.8-Flash predicts $117/130$ individual Tests correctly (90.0%), yet only $20/30$ episodes contain a fully correct joint submission. Its remaining ten episodes contain seven one-error and three two-error submissions. Many Observation failures therefore arise from a small number of residual prediction errors rather than uniformly incorrect predictions across the held-out Tests.

First-target success does not imply constrained transfer. Across all evaluated Intervention episodes, $117/210$ reach the Lock 1 target, but only $18/117$ of those episodes also reach Lock 2 within its adjustment budget. This gap appears across models: conditional Lock 2 success remains

Table 13: Family-level diagnostic results for Association. PIB Exceed reports the fraction of episodes exceeding the puzzle interaction budget. Outcome columns report episode counts.

Model	Family	PIB Exceed	Within PIB	Over PIB	Wrong	Horizon	Other
Gemini-3.8-Flash	A1	0.0%	10/10	0/10	0/10	0/10	0/10
Gemini-3.8-Flash	A2	40.0%	3/10	4/10	3/10	0/10	0/10
Gemini-3.8-Flash	A3	70.0%	1/10	5/10	4/10	0/10	0/10
Kimi-K3	A1	0.0%	10/10	0/10	0/10	0/10	0/10
Kimi-K3	A2	60.0%	3/10	4/10	3/10	0/10	0/10
Kimi-K3	A3	10.0%	0/10	0/10	10/10	0/10	0/10
GPT-5.6-Sol	A1	0.0%	9/10	0/10	1/10	0/10	0/10
GPT-5.6-Sol	A2	60.0%	3/10	3/10	4/10	0/10	0/10
GPT-5.6-Sol	A3	90.0%	0/10	4/10	6/10	0/10	0/10
Qwen3.8-2.4T-A95B	A1	0.0%	5/10	0/10	5/10	0/10	0/10
Qwen3.8-2.4T-A95B	A2	60.0%	3/10	6/10	1/10	0/10	0/10
Qwen3.8-2.4T-A95B	A3	30.0%	0/10	1/10	9/10	0/10	0/10
Grok-4.6	A1	80.0%	2/10	8/10	0/10	0/10	0/10
Grok-4.6	A2	100.0%	0/10	8/10	2/10	0/10	0/10
Grok-4.6	A3	100.0%	0/10	7/10	3/10	0/10	0/10
Claude-Opus-5	A1	30.0%	4/10	3/10	3/10	0/10	0/10
Claude-Opus-5	A2	100.0%	0/10	4/10	6/10	0/10	0/10
Claude-Opus-5	A3	100.0%	0/10	2/10	8/10	0/10	0/10
DeepSeek-V4.1-Flash	A1	10.0%	0/10	0/10	10/10	0/10	0/10
DeepSeek-V4.1-Flash	A2	80.0%	0/10	0/10	10/10	0/10	0/10
DeepSeek-V4.1-Flash	A3	20.0%	0/10	0/10	10/10	0/10	0/10

at or below 25.0% for every model. Reaching the first target is therefore much more common than using the information acquired during interaction to control a new target under a tight action budget.

Relevant factual evidence does not guarantee the right historical revision. Grok-4.6 acquires all annotated repair-relevant evidence, yet reaches only 60.6% Partial Edit Accuracy, a 50.0% Counterfactual Impact Rate, and 40.0% official success. The bottleneck is therefore not only whether the agent observes the relevant factual traces, but whether it turns those traces into a historical revision that changes the replay in the required way.

Task-level identifiability does not imply agent-level success. Across the four regimes, the benchmark controls whether the evidence available through the permitted interaction is sufficient to determine the target solution. The results show that this is only the starting point for an agent: it must still select evidence efficiently, integrate it into reliable predictions, carry information forward to a new control objective, or map factual traces to the correct alternative history. A single success rate collapses these failures into the same endpoint; the diagnostics reveal where the path from evidence to action breaks.

J Family-Level Diagnostic Results

We report family-level diagnostic results for all generator families. These results provide a finer-grained breakdown of the regime-level summaries in Figure 3. Metric definitions are given in Appendix H.

J.1 Association

Table 13 reports the family-level diagnostic results for Association.

J.2 Observation

Table 14 reports the family-level diagnostic results for Observation.

Table 14: Family-level diagnostic results for Observation. Test-Level Accuracy is computed over individual Test predictions. E0–E3+ report episode-level joint prediction errors. Invalid denotes episodes without a valid final joint submission.

Model	Family	Test-Level Acc.	E0	E1	E2	E3+	Invalid
Gemini-3.8-Flash	O1	95.0%	9/10	0/10	1/10	0/10	0/10
Gemini-3.8-Flash	O2	96.7%	9/10	1/10	0/10	0/10	0/10
Gemini-3.8-Flash	O3	83.3%	2/10	6/10	2/10	0/10	0/10
Kimi-K3	O1	95.0%	9/10	0/10	1/10	0/10	0/10
Kimi-K3	O2	93.3%	8/10	2/10	0/10	0/10	0/10
Kimi-K3	O3	75.0%	1/10	4/10	4/10	1/10	0/10
GPT-5.6-Sol	O1	82.5%	6/10	1/10	3/10	0/10	0/10
GPT-5.6-Sol	O2	96.7%	9/10	1/10	0/10	0/10	0/10
GPT-5.6-Sol	O3	73.3%	1/10	4/10	3/10	2/10	0/10
Qwen3.8-2.4T-A95B	O1	82.5%	6/10	1/10	3/10	0/10	0/10
Qwen3.8-2.4T-A95B	O2	86.7%	8/10	1/10	0/10	1/10	0/10
Qwen3.8-2.4T-A95B	O3	65.0%	0/10	3/10	3/10	4/10	0/10
Grok-4.6	O1	85.0%	6/10	2/10	2/10	0/10	0/10
Grok-4.6	O2	93.3%	8/10	2/10	0/10	0/10	0/10
Grok-4.6	O3	71.7%	0/10	6/10	1/10	3/10	0/10
Claude-Opus-5	O1	62.5%	1/10	3/10	6/10	0/10	0/10
Claude-Opus-5	O2	53.3%	2/10	3/10	4/10	1/10	0/10
Claude-Opus-5	O3	83.3%	2/10	6/10	2/10	0/10	0/10
DeepSeek-V4.1-Flash	O1	55.0%	1/10	1/10	7/10	1/10	0/10
DeepSeek-V4.1-Flash	O2	43.3%	0/10	5/10	3/10	2/10	0/10
DeepSeek-V4.1-Flash	O3	76.7%	2/10	5/10	1/10	2/10	0/10

J.3 Intervention

Table 15 reports the family-level diagnostic results for Intervention.

J.4 Counterfactual

Table 16 reports the family-level diagnostic results for Counterfactual.

Table 15: Family-level diagnostic results for Intervention. Lock 1 Success reports the fraction of episodes reaching the first target. Lock 2 Success | Lock 1 reports conditional second-target success among episodes completing Lock 1. L2/L1 reports the corresponding success count and eligible count. Outcome columns report episode counts.

Model	Fam.	L1 Succ.	L2 | L1	L2/L1	Both	L1 Fail	L2 Fail	Other
Gemini-3.8-Flash	I1	20.0%	50.0%	1/2	1	8	1	0
Gemini-3.8-Flash	I2	10.0%	0.0%	0/1	0	9	1	0
Gemini-3.8-Flash	I3	100.0%	0.0%	0/10	0	0	10	0
Kimi-K3	I1	50.0%	80.0%	4/5	4	5	1	0
Kimi-K3	I2	30.0%	0.0%	0/3	0	7	3	0
Kimi-K3	I3	100.0%	0.0%	0/10	0	0	10	0
GPT-5.6-Sol	I1	50.0%	60.0%	3/5	3	5	2	0
GPT-5.6-Sol	I2	70.0%	14.3%	1/7	1	3	6	0
GPT-5.6-Sol	I3	100.0%	10.0%	1/10	1	0	9	0
Qwen3.8-2.4T-A95B	I1	40.0%	75.0%	3/4	3	6	1	0
Qwen3.8-2.4T-A95B	I2	60.0%	33.3%	2/6	2	4	4	0
Qwen3.8-2.4T-A95B	I3	100.0%	0.0%	0/10	0	0	10	0
Grok-4.6	I1	20.0%	50.0%	1/2	1	8	1	0
Grok-4.6	I2	40.0%	0.0%	0/4	0	6	4	0
Grok-4.6	I3	100.0%	0.0%	0/10	0	0	10	0
Claude-Opus-5	I1	80.0%	25.0%	2/8	2	2	6	0
Claude-Opus-5	I2	0.0%	N/A	0/0	0	10	0	0
Claude-Opus-5	I3	100.0%	0.0%	0/10	0	0	10	0
DeepSeek-V4.1-Flash	I1	0.0%	N/A	0/0	0	10	0	0
DeepSeek-V4.1-Flash	I2	0.0%	N/A	0/0	0	10	0	0
DeepSeek-V4.1-Flash	I3	100.0%	0.0%	0/10	0	0	10	0

Table 16: Family-level diagnostic results for Counterfactual. Diagnostic metrics include Relevant Evidence Coverage, Partial Edit Accuracy, and Counterfactual Impact Rate. Official Success is included as a reference column. Claude Opus 5 Counterfactual results are unavailable.

Model	Family	Evidence Cov.	Partial Edit Acc.	Impact Rate	Official Succ.
Gemini-3.8-Flash	C1	76.2%	60.0%	40.0%	40.0%
Gemini-3.8-Flash	C2	74.9%	80.0%	90.0%	70.0%
Gemini-3.8-Flash	C3	71.9%	76.7%	70.0%	70.0%
Kimi-K3	C1	44.4%	46.7%	20.0%	20.0%
Kimi-K3	C2	46.9%	35.0%	40.0%	10.0%
Kimi-K3	C3	54.1%	63.3%	50.0%	50.0%
GPT-5.6-Sol	C1	59.3%	53.3%	30.0%	30.0%
GPT-5.6-Sol	C2	45.7%	40.0%	60.0%	20.0%
GPT-5.6-Sol	C3	55.6%	50.0%	30.0%	30.0%
Qwen3.8-2.4T-A95B	C1	60.0%	53.3%	30.0%	30.0%
Qwen3.8-2.4T-A95B	C2	57.6%	45.0%	50.0%	0.0%
Qwen3.8-2.4T-A95B	C3	59.5%	56.7%	40.0%	40.0%
Grok-4.6	C1	100.0%	53.3%	30.0%	30.0%
Grok-4.6	C2	100.0%	65.0%	70.0%	40.0%
Grok-4.6	C3	100.0%	63.3%	50.0%	50.0%
Claude-Opus-5	C1	N/A	N/A	N/A	N/A
Claude-Opus-5	C2	N/A	N/A	N/A	N/A
Claude-Opus-5	C3	N/A	N/A	N/A	N/A
DeepSeek-V4.1-Flash	C1	88.8%	46.7%	20.0%	20.0%
DeepSeek-V4.1-Flash	C2	84.7%	60.0%	70.0%	30.0%
DeepSeek-V4.1-Flash	C3	95.0%	63.3%	50.0%	50.0%