
ACED-BENCH: Evaluating How LLM Agents Acquire and Act on Causal Evidence

Yiwei Chen*
UC San Diego

Amila Weerasinghe
Amazon

Siddharth Biswal
Amazon

Utkrisht Rajkumar
Amazon

Olcay Boz
Amazon

Taylor Berg-Kirkpatrick
UC San Diego

Rose Yu
UC San Diego

Abstract

Large language models are increasingly used as scientific assistants, yet we still lack reliable ways to evaluate whether they can investigate a system rather than merely answer a prompt. A useful evaluation must let agents choose experiments, interpret finite evidence, and make causal decisions, while keeping the environment coherent across repeated observations and interventions and preserving objective ground truth. We introduce ACED-BENCH (Active Causal Evidence and Decision Benchmark), a neuro-symbolic benchmark built around this requirement. The key idea is to separate the readable scientific surface from the formal causal world: LLMs generate natural-language scenarios and task text, while hidden probabilistic graphical models define the causal graph, data-generating process, intervention semantics, validation checks, and oracle answers. This yields scientific worlds that are natural for agents to investigate but exact enough for scalable, programmatic evaluation. The main leaderboard ACED-CORE contains 180 tasks: 60 set-valued causal ancestry/descendancy questions and 120 constrained causal-decision tasks. We also report ACED-QUAL, a 150-question yes/no qualification check using the same hidden-PGM interface. We evaluate seven contemporary LLMs and two earlier-generation models across three active agent designs. ACED-QUAL checks whether agents can use the interface to draw conclusions from evidence, while ACED-CORE tests whether they can decide what evidence to collect and when to commit to an answer. Evidence-ledger analysis localizes failures within the investigation: agents test the wrong intervention, lose relevant evidence, or continue after sufficient evidence has appeared. These results suggest that reliable AI co-scientists need explicit support for experiment design, evidence retention, answer commitment, and stopping.

1 Introduction

Large language models (LLMs) are increasingly used as scientific assistants: they read papers, write code, analyze data, and propose hypotheses. If such systems are to become reliable AI co-scientists, we need to evaluate not only what they answer, but how they investigate. Scientific investigation requires deciding what would be informative to observe, choosing experiments, analyzing data, revising beliefs, and committing to a conclusion when the evidence is sufficient. This is especially important for causal questions, where patterns in passively observed data can differ from the effects of actively changing a system. A strong scientific agent must therefore know not only how to reason about causality, but also which experiment would make a causal claim testable.

*Email: vic010@ucsd.edu

Existing benchmarks capture parts of this problem, but rarely combine them. Causal-reasoning benchmarks test concepts such as intervention, confounding, and counterfactuals; statistical-reasoning benchmarks test inference from data; and scientific-agent benchmarks test tool use and multi-step investigation. What remains difficult is evaluating whether an agent can design useful experiments and make causal decisions in a world that is both readable in natural language and formally consistent. Such a benchmark must support repeated observation and intervention, provide coherent outcomes across experiments, and retain objective ground truth.

We introduce ACED-BENCH, a neuro-symbolic benchmark for evaluating whether LLM agents can investigate causal systems through experimental evidence. Each task is set in a natural-language scientific world backed by a hidden probabilistic graphical model (PGM) with categorical variables, a causal graph, and conditional probability tables (CPDs). The PGM defines a consistent data-generating process: agents may request observational data, run interventions, and write code to analyze the resulting finite samples under a fixed four-turn budget. Because the full PGM is known to the evaluator, gold answers are computed programmatically rather than manually labeled. Because the language layer is separated from the formal model, ACED-BENCH can scale through LLM-written scenarios combined with programmatic graph construction, validation, and oracle answer computation. Figure 1 illustrates the interaction loop for a hard ACED-CORE task.

The main leaderboard is ACED-CORE, a four-turn hard benchmark with 180 tasks: 60 set-valued causal ancestry/descendancy questions and 120 constrained causal-decision tasks, involving goals such as safe treatment choice, robust policy choice, mediation analysis, satisficing, and invalid-premise detection. These tasks require agents to identify the relevant causal contrast, collect appropriate evidence, and return a structured answer or decision. We separately report ACED-QUAL, a 150-question yes/no qualification check using the same hidden-PGM interface, which tests whether agents can recover more directly specified distributional and causal facts. This separation keeps the leaderboard focused on hard causal investigation while using ACED-QUAL to verify basic interface competence.

We evaluate seven contemporary LLMs and two earlier-generation reference models across three active agent designs. We also provide a guess-only comparison, which answers directly from the question without access to data, as a baseline for what can be inferred from the question text alone. Active evidence collection moves strong agents close to ceiling on ACED-QUAL, while ACED-CORE remains more sensitive to selecting the right causal contrast, spending the limited turn budget, and committing to a valid answer form.

Our contributions are:

- We introduce ACED-BENCH, an interactive PGM-based benchmark for evaluating LLM agents on active causal evidence acquisition in formally grounded scientific worlds. The main leaderboard is the four-turn ACED-CORE benchmark, containing 180 hard tasks: 60 set-valued causal ancestry/descendancy questions and 120 constrained causal-decision questions. ACED-QUAL provides a separate 150-question yes/no qualification check.
- We provide a scalable neuro-symbolic generation pipeline in which LLMs create semantic scenarios and task text, while programmatic components enforce graph construction, CPD sampling, intervention semantics, validation, and oracle answer computation.
- We evaluate seven contemporary LLMs, with two earlier-generation models as references, across active reasoning, single-prompt active-code, and modular active-code agents. Strong agents approach ceiling on ACED-QUAL, while ACED-CORE preserves substantial spread across both models and agent designs.
- We show that final accuracy hides actionable investigation bottlenecks: the evidence ledger separates agents that miss the right estimand from agents that find and lose it, while the SAMP5K ablation shows that changing the scarce resource from turns to sampled rows changes which agent designs improve.

2 Related work

Agentic scientific and data-analysis benchmarks. Recent benchmarks increasingly evaluate LLMs as agents that interact with tools, environments, or data sources. ScienceWorld and DiscoveryWorld test scientific exploration in interactive environments requiring hypothesis formation,

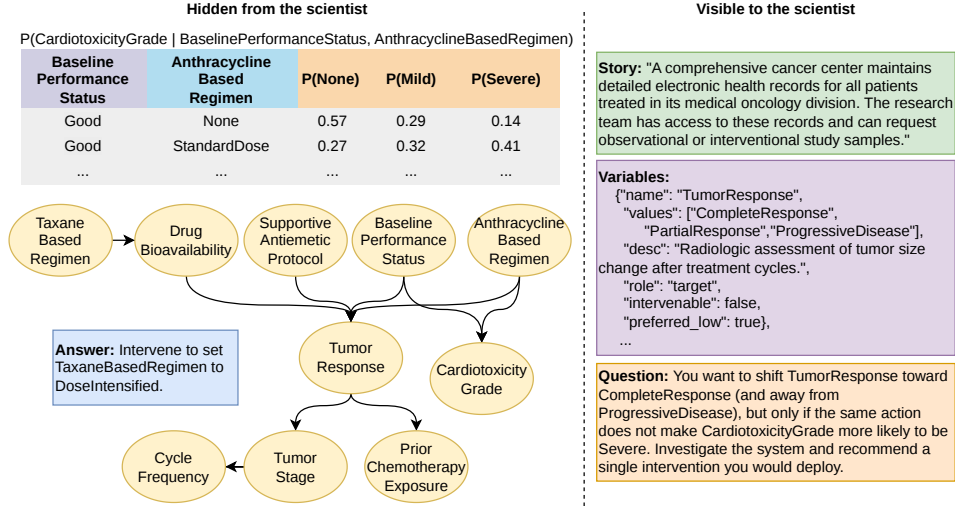


Figure 1: Overview of ACED-BENCH on a hard causal-decision task. Each task is generated from a hidden PGM. The benchmark oracle has access to the causal graph, CPDs, and exact interventional answers. The LLM scientist agent only observes the scenario, variable metadata, non-intervenability annotations, and the question, and must answer under a four-turn budget.

experimentation, and explanation [1, 2]. DiscoveryBench, BLADE, DA-Bench, and KramaBench evaluate data-driven investigation, code execution, and data-to-insight workflows [3–6]. These benchmarks establish that scientific-agent evaluation should include planning, tool use, and iterative interaction. ACED-BENCH asks a complementary question: whether an agent can gather and use evidence in a controlled causal system. Agents still interact, query data, and write code, but every world has exact observational and interventional semantics, allowing oracle-checkable evaluation of causal evidence, causal decisions, and the trajectory that produced them.

Causal and statistical reasoning benchmarks. Causal-reasoning benchmarks test whether LLMs understand concepts such as association, intervention, confounding, and counterfactuals [7–10]. Statistical-reasoning benchmarks evaluate probability, uncertainty, calibration, and data analysis [11–16]. CauSciBench is especially close in its focus on causal-scientific reasoning, but the relevant evidence in these benchmarks is usually given directly as text, a graph, or a fixed dataset [17]. In ACED-BENCH, evidence is endogenous: the agent must decide which observational or interventional samples to request, analyze finite data, and determine when the evidence is sufficient.

LLMs for causal discovery and experimental design. Several works use LLMs to assist causal discovery, for example by supplying background knowledge, orienting edges, reducing equivalence classes, or guiding intervention selection [18–21]. This line of work is close to our causal motivation, but often treats the LLM as one component in a causal-discovery algorithm. Interactive causal benchmarks such as Auto-Bench also evaluate iterative discovery, but focus primarily on hidden-graph or adjacency-matrix reconstruction [22]. Our benchmark instead evaluates sample-based causal investigation: agents answer statistical and causal queries, choose constrained interventions, and are scored by exact PGM ground truth.

Active causal learning and intervention design. Our setting is grounded in causal inference and PGMs [23, 24]. It is also related to active causal learning, where interventions are selected to identify structure or reduce uncertainty [25–28]. Classical methods often assume an explicit statistical model, posterior, or acquisition objective. In contrast, ACED-BENCH evaluates general-purpose LLM agents that interact through natural language, sampling APIs, and optional code execution. The goal is not to propose a new intervention-selection algorithm, but to test whether agents can choose informative evidence and use it correctly.

Trajectory-level evaluation and evidence management. Final-answer accuracy can hide important differences between agents that make the same mistake for different reasons. Process-level evaluation

and language-agent trajectory analysis therefore study intermediate reasoning, actions, observations, and tool use [29–33]. ACED-BENCH supports this kind of analysis by logging actions, queries, code executions, and final answers, enabling secondary trajectory-level analyses.

3 ACED-BENCH

3.1 Hidden-PGM interaction

ACED-BENCH evaluates LLM agents as sample-based causal investigators in hidden PGM worlds. Each world is a tuple

$$W = (G, \mathcal{V}, \mathcal{P}, s, \mathcal{Q}, \mathcal{N}),$$

where $G = (V, E)$ is a DAG over categorical variables, $\mathcal{V}$ contains variable names, descriptions, and value sets, $\mathcal{P}$ defines the Bayesian-network CPDs, s is a natural-language scientific scenario, $\mathcal{Q}$ is a set of questions, and $\mathcal{N} \subseteq V$ is the set of non-intervenable variables. The evaluator has access to the full PGM. The agent observes only the scenario, variable metadata, non-intervenability annotations, the question, and its interaction history. At each turn, the agent may request observational samples from $P_W(V)$, request interventional samples from $P_W(V \mid \text{do}(X = x))$, run code when permitted, or submit a final answer. The primary ACED-CORE score is evaluated under a four-turn budget: query turns, failed-query turns, final answers, and give-up actions each count as one outer scientist turn. Figure 1 illustrates this hidden-world interface. Appendix A shows the analogous interaction loop for structural questions.

3.2 World generation

ACED-BENCH makes the scientific story readable without letting language determine the answer. An LLM drafts variable names, value descriptions, scenarios, and task text; a programmatic backend constructs the graph, samples CPDs, defines interventions, validates the world, and computes the oracle answer. The agent therefore encounters a plausible scientific scenario, while the evaluator retains exact causal ground truth.

This separation also makes the benchmark scalable. New domains, graph sizes, and archetype instances can be generated and filtered without manual answer labeling; the limiting costs are generation, validation rejection, simulation, and LLM evaluation. For causal-decision tasks, Figure 2 summarizes how a decision archetype turns a causal signature into a scientific task. The archetype fixes semantic roles, graph constraints, validation checks, and the oracle answer, while the LLM supplies the surface wording. Exact interventional inference must then confirm the intended causal signature. Worlds with broken graph constraints, small effect margins, or ambiguous oracle answers are rejected. Appendix B gives the structural and archetype-specific validation criteria used to accept generated worlds.

3.3 Benchmark split and gold answers

ACED-CORE. ACED-CORE allows a four-turn budget, combining 60 causal ancestry/descendancy list queries with 120 causal-decision questions built around safety, robustness, mediation, satisficing, and invalid-premise archetypes. The ancestry/descendancy queries ask for all causes or all effects of a target variable, so the agent must discover and aggregate multiple causal relations rather than answer a single binary contrast. The decision questions require recommending a valid intervention, mediator, or rejection. Together, they test whether agents can move from evidence access to causal commitment under a tight interaction budget.

ACED-QUAL. ACED-QUAL is a separate qualification check using the same hidden-PGM interface and four-turn budget. It contains 150 yes/no structural questions over the same style of worlds: causal-effect questions plus marginal and conditional queries. Because the question specifies the target contrast, ACED-QUAL asks whether an agent can obtain and interpret the relevant evidence once the scientific question has already been posed.

The split separates three increasingly demanding skills. The 150 yes/no qualification questions test whether agents can use evidence for a stated contrast. The 60 causal list questions test whether they can search over many possible relations and synthesize a complete set. The 120 decision questions

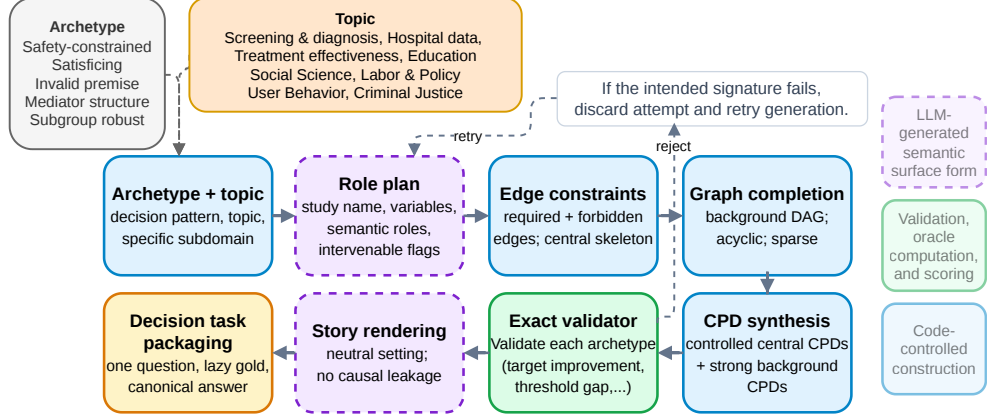


Figure 2: Generation pipeline for the causal-decision portion of ACED-CORE. A decision pattern and topic determine semantic roles, graph constraints, CPD construction, validation checks, and the final question. LLMs provide readable scientific surface forms, while code enforces causal validity and computes oracle answers.

Table 1: Evaluation regimes in ACED-BENCH. ACED-CORE is the main benchmark, while ACED-QUAL is a qualification check for basic evidence use.

Regime	Answer format	Tasks
ACED-CORE	Lists / decisions	60 causal list + 120 decision (180 total)
ACED-QUAL	Yes/no	150 causal-effect, marginal, conditional

test whether they can convert causal evidence into a constrained scientific action. Only the latter two groups define the main ACED-CORE leaderboard.

Table 1 summarizes the main benchmark and the qualification check. Both use worlds spanning eight topical domains: Criminal Justice, Education, Hospital Data, Labor & Policy, Screening & Diagnosis, Social Science, Treatment Effectiveness, and User Behavior. The split is methodological rather than merely taxonomic: ACED-QUAL asks whether the agent can use the interface for specified yes/no contrasts, while ACED-CORE asks whether it can decide which causal comparisons matter and return a structured causal answer under budget.

All gold answers are computed from the underlying PGM. Generated worlds are filtered using effect-size margins, threshold gaps, and tie tolerances to avoid unstable labels. Appendix C discusses the fine-grained structural question breakdown.

4 Agent and evaluation protocol

Our protocol isolates two questions: whether LLM agents can choose informative experiments under a tight interaction budget, and which stages of the investigation fail when they do not. All agents receive the same scenario, variable metadata, non-intervenability annotations, question, and interaction history; the hidden-PGM interface is fixed across runs. We vary the LLM and agent design, from no-data guessing to active sampling, code-assisted analysis, and modular planning. The protocol supports both final-answer scoring and error-mode analyses of evidence selection, causal-contrast estimation, answer commitment, and stopping.

Agent variants. We consider four agent designs. The main four-turn ACED-CORE leaderboard contains the three designs that actively request evidence; guess-only is a separate prior-only sanity check. The **guess-only** agent answers immediately without data access. The **active reasoning** agent requests observational or interventional samples and reasons from simulator summaries, but cannot execute code. The **single-prompt active-code** agent can additionally write Python code to analyze sampled CSV files. The **modular active-code** agent separates planning, code analysis, interpretation,

and experiment design into explicit stages. The three active variants isolate the effects of active sampling, code execution, and modular decomposition under the same benchmark interface.

Models. We evaluate seven contemporary LLMs: Claude Opus-4.6, Claude Sonnet-4.6, Gemini-3-Pro, DeepSeek-V4-Pro, Qwen3.6-27B, GPT-OSS-120B, and Llama-4-Maverick. We also include GPT-4o and Llama-3.3-70B as earlier-generation reference models. For each model, we run the agent variants supported by our inference stack on ACED-CORE and ACED-QUAL. The main text reports the four-turn budgeted setting. A sampled-row budget ablation is summarized in Appendix G.

Interaction budget and parser. The main ACED-CORE setting caps each active agent at four outer scientist turns. The underlying interface also enforces a ceiling of 10 successful data queries per task, but the four-turn budget is the binding resource constraint in the main evaluation. Validation and parser failures do not spend this successful-query budget. A fixed parser LLM maps natural-language data requests to typed simulator queries; in our runs this parser is Claude Opus-4.6 via AWS Bedrock (`us.anthropic.claude-opus-4-6-v1`), temperature 0.1, max output 512 tokens. The parser is not evaluated as a scientist agent: it performs no causal reasoning, and deterministic checks enforce query validity and non-intervenability constraints. Holding this interface fixed ensures that model comparisons reflect evidence selection and analysis rather than differences in data access.

Scoring. ACED-QUAL binary questions are scored by accuracy. ACED-CORE uses task-specific scoring: causal ancestry/descendancy list queries are scored by exact match and F_1 , categorical mediator questions are scored by exact label match, intervention-choice decision questions are parsed into a variable-value intervention and scored against the oracle effect table, and invalid-premise questions are scored against the oracle rejection or alternative recommendation. For intervention-choice decision tasks, effects are compared using the expected-state index induced by the ordered categorical outcome states; lower index is treated as better unless the world explicitly specifies otherwise. We report tie-accepted accuracy unless otherwise stated, with decision ties and safety comparisons accepted within 0.05 expected-state-index units. For satisficing tasks, the predicted intervention must also satisfy the stored improvement threshold. For subgroup-robust tasks, the predicted intervention must improve every subgroup by at least 0.15 expected-state-index units at scoring time.

Uncertainty estimates. For structural questions, including the 150 ACED-QUAL questions and the 60 causal-ancestry/descendancy questions in ACED-CORE, we report world-clustered bootstrap intervals, resampling worlds rather than individual questions to preserve dependence among questions generated from the same graph. For causal-decision questions, each world contributes one question, so world-level and question-level bootstraps coincide. For within-world comparisons between agent variants, we use paired bootstrap intervals. Appendix D gives the corresponding bootstrap details for the plotted intervals.

Behavioral metrics. In addition to final-answer accuracy, we log query budget use, failed queries, code executions, code failures, give-up rate, and binary label bias. These metrics help distinguish failures of causal reasoning from failures of tool use, parsing, or budget management.

4.1 Evidence-ledger diagnostic

Final-answer accuracy does not reveal why an interactive agent failed. A trajectory may fail because the agent never collected the right evidence, because it collected evidence but did not use it in the final answer, or because it kept querying after sufficient evidence had already appeared. We therefore build an evidence ledger for modular active-code trajectories. The ledger asks whether a trajectory ever reaches evidence that would justify the gold answer. For each turn, an LLM annotator judges whether the turn computed the correct estimand, whether the resulting evidence supports the gold answer, and whether that evidence is adequate. A turn is marked as *strong evidence* only if all three conditions hold. Four derived diagnostics then separate evidence acquisition, evidence retention, ignored evidence, and extra turns after evidence. The benchmark score says whether the agent answered correctly; the ledger explains where the investigation broke.

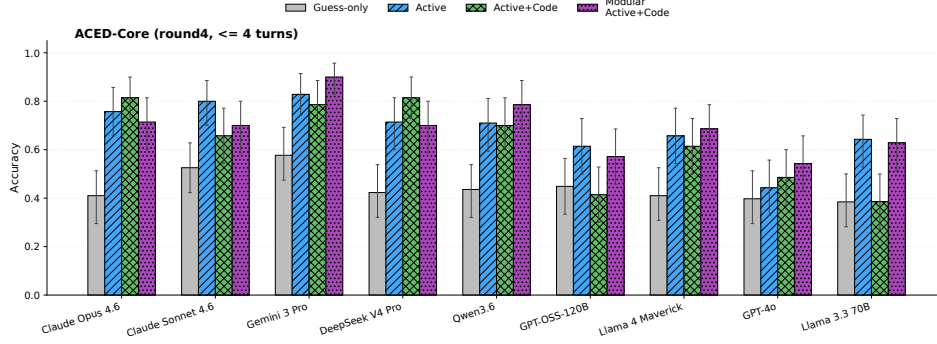


Figure 3: Accuracy on ACED-CORE, the main hard benchmark. Results use the four-turn budgeted setting and combine causal ancestry/descendancy list queries with constrained causal-decision tasks. The budget applies to active agents; guess-only cells are omitted because they do not use interaction turns. Error bars are 95% world-clustered bootstrap intervals.

5 Results

All main results use the four-turn budgeted setting. We present the hard ACED-CORE leaderboard, the ACED-QUAL qualification check, and the ledger diagnostic explaining failure mechanisms. Appendix G reports a sampled-row ablation showing how the conclusions change when rows, rather than turns, are the constrained resource.

5.1 ACED-CORE results

ACED-CORE makes the scientifically hard step precede the statistic: an agent must decide which causal contrast matters before data analysis can help. Figure 3 reports accuracy on this four-turn benchmark, combining causal ancestry/descendancy list queries with constrained causal-decision questions. The results preserve substantial spread across models and agent designs, indicating that current active agents still differ sharply in experiment selection and causal commitment.

The four-turn budget makes code access useful but not automatically beneficial. Some models benefit from code or modular decomposition, while others lose accuracy when the code loop consumes scarce turns or fails to produce a clean causal comparison. Single-prompt active-code can underperform active reasoning, while modular decomposition recovers some but not all of these failures. Thus, ACED-CORE stresses the same pressure that matters in scientific work: the scarce resource is often not computing a statistic once data are in hand, but choosing which evidence would answer the question and recognizing when it is strong enough to act on.

5.2 ACED-QUAL results

Figure 4 reports the same four-turn protocol on ACED-QUAL, the 150-question yes/no qualification check. ACED-QUAL is easier because the question specifies the target causal-effect, marginal, or conditional contrast; the agent mainly needs to request and interpret the relevant samples. Several active cells are at or near ceiling, showing that strong agents can use the sampling interface when the relevant contrast is specified.

ACED-QUAL serves as a qualification check for evidence use, rather than as the main scientific challenge. Failures remain informative, especially when a model’s code loop is brittle. For example, GPT-OSS-120B with the single-prompt coder often fails by violating the action protocol rather than by making an incorrect causal judgment. For strong active agents, however, ACED-QUAL mostly confirms that the hidden-PGM interface is usable before the evaluation turns to ACED-CORE.

Together, the two result panels separate basic evidence use from the harder benchmark problem. ACED-QUAL shows that many current agents can use the interface when the relevant contrast is specified. ACED-CORE remains harder because the agent must decide what evidence matters, gather it within four turns, and express the final answer in a constrained structured form. The contrast between panels identifies where the difficulty enters, rather than merely showing that one

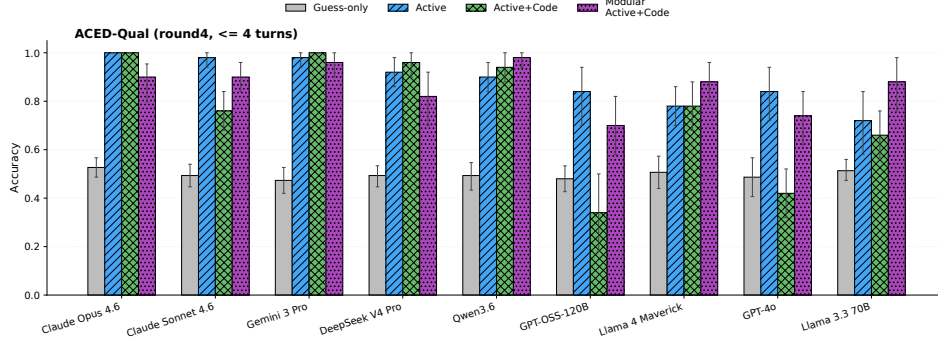


Figure 4: Accuracy on ACED-QUAL, the 150-question yes/no qualification check. Results use the same four-turn budgeted setting as ACED-CORE. These questions test causal-effect, marginal, and conditional judgments from actively requested samples. The budget applies to active agents; guess-only cells are omitted because they do not use interaction turns. Error bars are 95% world-clustered bootstrap intervals.

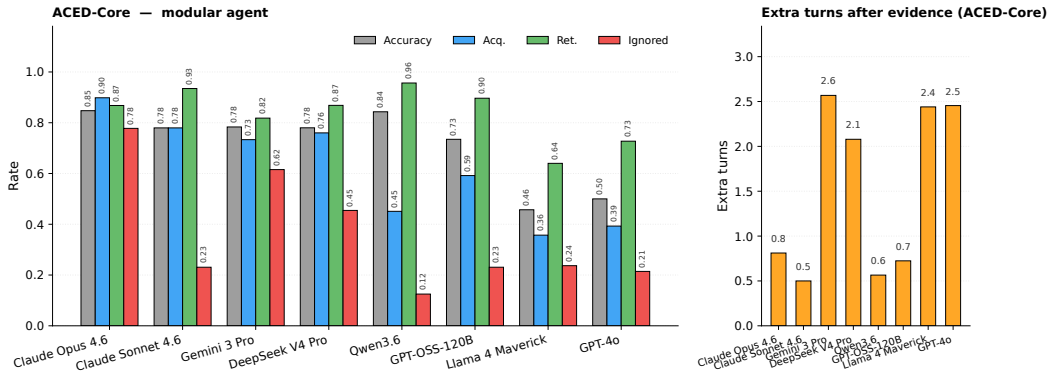


Figure 5: Evidence-ledger diagnostics for modular active-code trajectories on ACED-CORE. The left panel reports final-answer accuracy, evidence acquisition, evidence retention, and ignored evidence. The right panel reports extra turns after the first strong-evidence turn.

set of questions is easier. Many agents can estimate a named contrast, but fewer can formulate the right contrast, search over a set of possible causal relations, or convert evidence into a constrained intervention decision.

5.3 Failures often start before the final answer

Across 3,579 evaluated modular active-code trajectories, many final-answer errors are already visible before the final response. Failed ACED-CORE runs often never produce a strong-evidence turn for the target causal contrast. When strong evidence does appear, agents do not always preserve it: some trajectories compute evidence that supports the gold answer, then answer incorrectly or incompletely. A third failure mode is stopping: agents sometimes continue querying after the first strong-evidence turn instead of committing. These patterns identify three scientific bottlenecks: experiment selection, evidence retention, and stopping. Figure 5 summarizes these error modes. The next subsection shows how the same labels identify the moment when a trajectory becomes actionable. Appendix E shows the corresponding acquisition-retention scatter across ACED-QUAL and ACED-CORE.

5.4 A trace shows when evidence becomes actionable

Figure 6 shows a safety-constrained ACED-CORE decision task where the agent must improve DisorderDetectionTimeliness without increasing FalsePositiveBurden. The early turns are insufficient because the agent has not yet compared candidate interventions on both the target and

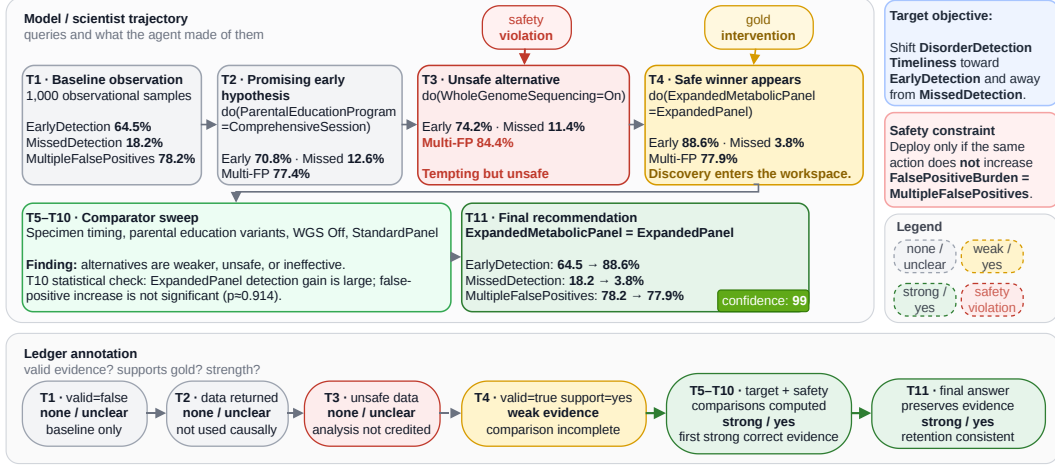


Figure 6: Example evidence-ledger trace for a safety-constrained ACED-CORE decision task. Strong evidence appears only after the agent compares candidate interventions on both the target and safety outcomes.

safety outcomes. The decisive comparison supports `ExpandedMetabolicPanel=ExpandedPanel`: early detection rises from 64.5% to 88.6%, while the safety outcome remains essentially unchanged. The final answer retains this supported recommendation. Additional trajectory examples are provided in Appendix F.

6 Limitations and discussion

The scientific control that makes ACED-BENCH evaluable also narrows its scope. The current benchmark uses fully observed categorical PGMs with known intervention semantics, so it does not yet cover continuous variables, latent confounding, temporal dynamics, measurement error, selection effects, model misspecification, or open-ended experiment design. The scenarios are synthetic, but the labels are not language-model judgments: code constructs graphs, samples CPDs, enforces intervention semantics, validates margins, and computes oracle answers. The evaluated instance contains 180 ACED-CORE tasks and 150 ACED-QUAL questions; larger releases could support finer analysis by domain, graph structure, decision archetype, and failure mode. The evidence ledger remains a diagnostic analysis, while final-answer accuracy under the fixed interaction budget is the benchmark score.

The intended positive impact is sharper evaluation of AI systems proposed as scientific assistants. The main risk is overinterpretation: success in these synthetic hidden-PGM worlds should not be read as evidence of deployment readiness in real scientific or policy settings, especially where data quality, fairness, privacy, and intervention semantics are contested.

7 Conclusion

We introduced ACED-BENCH, a neuro-symbolic benchmark for evaluating LLM agents as active causal investigators in hidden PGM worlds. ACED-QUAL checks basic evidence use under the same interface, while ACED-CORE tests whether agents can identify the relevant causal contrast, collect evidence, and return structured causal-reachability answers or constrained causal decisions.

Across seven contemporary LLMs, two earlier-generation references, and three active agent designs, ACED-CORE preserves substantial spread. The results point to concrete investigation skills: selecting informative experiments, estimating the right causal quantities, retaining evidence across turns, and stopping when the answer is supported. By combining natural-language scientific scenarios with oracle-scored causal simulators, ACED-BENCH provides a controlled way to measure how future agents acquire and act on causal evidence.

References

- [1] Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Science-World: Is your Agent Smarter than a 5th Grader? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.775.
- [2] 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, December 2024. doi: 10.52202/079017-0324.
- [3] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards Data-Driven Discovery with Large Language Models, July 2024.
- [4] Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A Merrill, Jeffrey Heer, and Tim Althoff. BLADE: Benchmarking Language Model Agents for Data-Driven Science. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13936–13971, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.815.
- [5] Xueyu Hu, Ziyu Zhao, Shuang Wei, et al. Infiagent-dabench: Evaluating agents on data analysis tasks. In *Proceedings of the 41st International Conference on Machine Learning*, pages 19544–19572. PMLR, 2024.
- [6] Eugenie Lai, Gerardo Vitagliano, Ziyu Zhang, Om Chabra, Sivaprasad Sudhir, Anna Zeng, Anton A. Zabreyko, Chenning Li, Ferdi Kossmann, Jialin Ding, Jun Chen, Markos Markakis, Matthew Russo, Weiyang Wang, Ziniu Wu, Michael J. Cafarella, Lei Cao, Samuel Madden, and Tim Kraska. KramaBench: A Benchmark for AI Systems on Data-to-Insight Pipelines over Data Lakes, March 2026.
- [7] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. CLadder: Assessing Causal Reasoning in Language Models. *Advances in Neural Information Processing Systems*, 36:31038–31065, December 2023.
- [8] Zeyu Wang. CausalBench: A Comprehensive Benchmark for Evaluating Causal Reasoning Capabilities of Large Language Models. In Kam-Fai Wong, Min Zhang, Ruifeng Xu, Jing Li, Zhongyu Wei, Lin Gui, Bin Liang, and Runcong Zhao, editors, *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 143–151, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [9] Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can Large Language Models Infer Causation from Correlation?, April 2024.
- [10] Jin Du, Li Chen, Xun Xian, An Luo, Fangqiao Tian, Ganghua Wang, Charles Doss, Xiaotong Shen, and Jie Ding. Ice Cream Doesn’t Cause Drowning: Benchmarking LLMs Against Statistical Pitfalls in Causal Inference, March 2026.
- [11] Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. Are LLMs Capable of Data-based Statistical and Causal Reasoning? Benchmarking Advanced Quantitative Reasoning with Data. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9215–9235, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.548.

- [12] Akshay Paruchuri, Jake Garrison, Shun Liao, John B Hernandez, Jacob Sunshine, Tim Althoff, Xin Liu, and Daniel McDuff. What Are the Odds? Language Models Are Capable of Probabilistic Reasoning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11712–11733, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.654.
- [13] Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. Reasoning over Uncertain Text by Generative Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24911–24920, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i23.34674.
- [14] Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. Are Large Language Models Good Statisticians? *Advances in Neural Information Processing Systems*, 37:62697–62731, December 2024. doi: 10.52202/079017-2005.
- [15] Alana Renda, Jillian Ross, Michael Cafarella, and Jacob Andreas. OpenEstimate: Evaluating LLMs on Reasoning Under Uncertainty with Real-World Data, April 2026.
- [16] Yuchen Lu, Run Yang, Yichen Zhang, Shuguang Yu, Runpeng Dai, Ziwei Wang, Jiayi Xiang, Wenxin E, Siran Gao, Xinyao Ruan, Yirui Huang, Chenjing Xi, Haibo Hu, Yueming Fu, Qinglan Yu, Xiaobing Wei, Jiani Gu, Rui Sun, Jiaxuan Jia, and Fan Zhou. StatEval: A Comprehensive Benchmark for Large Language Models in Statistics, October 2025.
- [17] Sawal Acharya, Terry Jingchen Zhang, Andrew Kim, Anahita Haghighat, Xianlin Sun, Rahul Babu Shrestha, Maximilian Mordig, Furkan Danisman, Clijo Jose, Yahang Qi, Pepijn Cobben, Bernhard Schölkopf, Mrinmaya Sachan, and Zhijing Jin. CauSciBench: Assessing LLM Causal Reasoning for Scientific Research. In *NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science*, October 2025.
- [18] Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal Discovery with Language Models as Imperfect Experts, July 2023.
- [19] Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. Integrating Large Language Models in Causal Discovery: A Statistical Causal Approach, May 2025.
- [20] Ahmed Abdulaal, Adamos Hadjivasiliou, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C Castro, and Daniel C Alexander. CAUSAL MODELLING AGENTS: CAUSAL GRAPH DISCOVERY THROUGH SYNERGISING METADATA- AND DATA-DRIVEN REASONING. 2024.
- [21] Junyi Li, Yongqiang Chen, Chenxi Liu, Qianyi Cai, Tongliang Liu, Bo Han, Kun Zhang, and Hui Xiong. Can Large Language Models Help Experimental Design for Causal Discovery?, March 2025.
- [22] Tingting Chen, Srinivas Anumasa, Beibei Lin, Vedant Shah, Anirudh Goyal, and Dianbo Liu. Auto-Bench: An Automated Benchmark for Scientific Discovery in LLMs, February 2025.
- [23] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.
- [24] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.
- [25] Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(11), 2008.
- [26] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

- [27] Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius von Kügelgen. Active Bayesian Causal Inference. *Advances in Neural Information Processing Systems*, 35:16261–16275, December 2022.
- [28] Jiaqi Zhang, Louis Cammarata, Chandler Squires, Themistoklis P. Sapsis, and Caroline Uhler. Active Learning for Optimal Intervention Design in Causal Models, August 2023.
- [29] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s Verify Step by Step, May 2023.
- [30] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models, March 2023.
- [31] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, December 2023.
- [32] Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. Beyond the Final Answer: Evaluating the Reasoning Trajectories of Tool-Augmented Agents, October 2025.
- [33] Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. AgentProcessBench: Diagnosing Step-Level Process Quality in Tool-Using Agents, March 2026.

A Additional benchmark diagrams

Figure 7 shows how structural questions use the same hidden-world interface as the causal-decision tasks. The distinction matters because ACED-QUAL and the causal-ancestry/descendancy portion of ACED-CORE use the same hidden-PGM simulator but place different demands on the agent. For ACED-QUAL, the question specifies a single yes/no causal-effect, marginal, or conditional contrast, so the main challenge is requesting and interpreting the relevant observational or interventional samples. For causal ancestry/descendancy queries, the agent must repeat this process over multiple candidate variables and aggregate the evidence into a set/list answer. This shared interface lets the structural generator support both a qualification check and a harder set/list component of the main benchmark.

The figure also highlights a design choice shared by all ACED-BENCH tasks: the agent never sees the graph or CPDs directly. It sees only natural-language metadata and the sampling interface, while the oracle computes labels from the hidden PGM. This separation lets the benchmark evaluate active evidence acquisition rather than graph reading or answer extraction from the prompt.

B World validation details

Structural validation. For structural worlds, question labels are determined by graph ancestry and d-separation. D-separation gives the conditional-independence guarantee; CPD resampling is used only to avoid accidental near-independences in graph configurations that should be dependent and to sanity-check the simulator. For each tested pair of variables (A, B) and conditioning variable Z , we compute

$$\text{TV}(P(A, B \mid Z = z), P(A \mid Z = z)P(B \mid Z = z))$$

by exact inference for each value z of Z . When the graph implies that A and B should be conditionally dependent given Z , we require the distance to exceed $\varepsilon_{\text{faith}} = 0.02$. When the graph d-separates the variables, exact inference is used as an implementation check rather than as the source of the independence label. CPDs are resampled up to 64 times until the dependency checks pass or the attempt budget is exhausted. Worlds for which no CPD draw passes validation are discarded. These checks filter accidental independences and unstable dependencies that would make structural questions ambiguous.

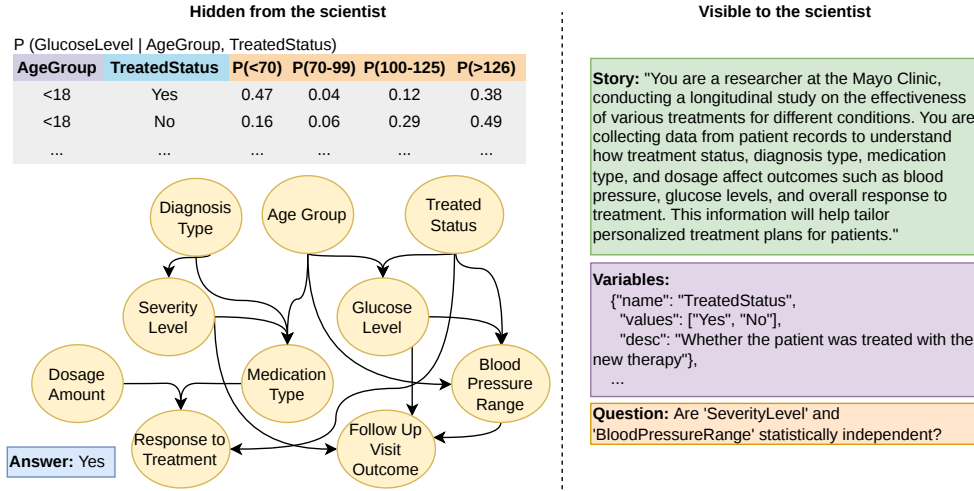


Figure 7: Interaction loop for structural questions, including the ACED-QUAL qualification regime. The benchmark oracle samples from a hidden PGM, while the LLM scientist agent observes only the scenario, variable metadata, non-intervenability annotations, and question. The agent requests observational or interventional samples and answers structural statistical or causal queries, such as marginal, conditional, causal-effect, or causal-ancestry/descendancy questions.

Causal-decision archetypes. The causal-decision generator contributes 120 questions to the main hard benchmark. Each world is built around one of five decision archetypes. **Safety** asks the agent to improve a target without causing unacceptable harm to a side outcome. **Satisficing** asks the agent to find any intervention that improves the target by a meaningful amount, rather than the single best intervention. **Subgroup robustness** asks for an intervention that improves outcomes across all subgroup values, not only on average. **Mediation** asks whether an effect operates through a proposed mediator or which candidate mediator explains the effect. **Invalid premise** asks the agent to reject a proposed intervention that is not causally valid, such as intervening on a non-intervenable variable, and to recommend a valid alternative when one exists. These archetypes require agents to identify the relevant causal contrast before analyzing data.

Causal-decision validation. For causal-decision worlds, each archetype has a dedicated exact-inference validator. Expected-state-index effects are computed from the explicit ordered state list for each outcome variable, using zero-based state indices; lower index is better for outcome variables unless the world metadata states otherwise. Safety-constrained worlds are rejected unless a safe and effective action exists: at least one intervention improves the target by at least 0.20 expected-state-index units while changing the safety outcome by at most 0.08 units. They are also rejected unless the tempting unsafe action genuinely worsens the safety outcome.

Mediator-structure worlds are rejected unless the total causal effect of treatment on outcome exceeds 0.20 units and the subvariant-specific signature holds. For mediated-only worlds, the residual treatment effect after intervening on the mediator must be at most 0.10 units. For direct-and-mediated worlds, the residual effect must exceed 0.18 units. For not-mediator worlds, fixing the proposed mediator must leave a residual treatment effect of at least 0.18 units, and intervening on the proposed mediator must move the outcome by at most 0.10 units. For which-mediator worlds, the true mediator's interventional effect must exceed the runner-up by at least 0.20 units.

Satisficing worlds are rejected unless the improvement distribution has a clean gap of at least 0.10 expected-state-index units that allows the threshold to be placed without ambiguity. Subgroup-robust worlds are rejected unless the robust intervention improves every subgroup by at least 0.10 units, the average-best intervention is at least 0.05 worse for some subgroup, and the bad intervention harms at least one subgroup. Invalid-premise worlds are rejected unless the proposed invalid variable has an observed association of at least 0.20 with the target and the recommended alternative has an interventional effect of at least 0.20. Across all archetypes, worlds are also rejected when required or

forbidden edges cannot be placed within the generated graph, when effect margins are too small, or when the oracle answer is ambiguous.

C Structural question-type breakdown

The structural question generator contains two evaluation groups that differ in how much experiment selection they require. The ACED-QUAL group contains 150 yes/no questions: causal-effect questions, marginal questions, and conditional questions. These questions provide the agent with a relatively clear estimand, so active sampling can often estimate the relevant relationship directly. The ACED-CORE structural component contains 60 set/list structural questions: `all_causes_of` and `all_effects_of`. These causal-ancestry/descendancy queries are harder because the agent must discover and aggregate a set of causal relations across many candidate variables. This distinction explains why strong active agents can nearly solve ACED-QUAL while still making substantial errors on the causal-ancestry/descendancy portion of ACED-CORE.

The fine-grained breakdown should therefore be read as a diagnostic decomposition of the structural generator, not merely as a ranking of models. Yes/no questions mainly test whether the agent can request and interpret the relevant distributional or causal comparison. Causal ancestry/descendancy queries additionally test whether the agent can plan a set of interventions or comparisons, track multiple candidate variables, and synthesize the resulting evidence into a set/list answer.

D Regime accuracy with 95% confidence intervals

Figures 4 and 3 report 95% percentile bootstrap CIs with $B=2000$ resamples. For the 150 ACED-QUAL yes/no questions and the 60 causal-ancestry/descendancy questions in ACED-CORE, we resample structural worlds rather than individual questions to preserve within-world dependence from shared graphs and CPDs. For the causal-decision portion of ACED-CORE, each world contributes one question, so world-level and question-level bootstraps coincide. For within-world comparisons between agent variants, we use paired bootstrap intervals.

E Evidence-ledger scatter diagnostic

Figure 8 compares evidence acquisition and evidence retention for modular active-code trajectories on ACED-QUAL and ACED-CORE. This view complements the ACED-CORE bar chart in Figure 5. ACED-QUAL and ACED-CORE are shown together because the contrast is diagnostic: ACED-QUAL tests whether agents can use the interface when the relevant estimand is specified, while ACED-CORE requires the agent to identify and preserve the relevant causal contrast. Llama-3.3-70B is omitted because eval-backed modular active-code ledger trajectories were not available.

F Illustrative evidence-ledger trajectories

This appendix gives additional qualitative examples from high-performing modular active-code trajectories on ACED-CORE decision tasks. As in Section 5.4, these examples are not used as benchmark scores. They are included to make the evidence-ledger annotations easier to interpret and to show how similar final-answer outcomes can arise from different investigative paths.

Safety-constrained discovery trace. The main-text trace in Figure 6 comes from a Gemini-3-Pro ACED-CORE decision task. The agent must improve `DisorderDetectionTimeliness` toward `EarlyDetection` without making `FalsePositiveBurden` more likely to be `MultipleFalsePositives`. The observational baseline has `EarlyDetection`=64.5%, `MissedDetection`=18.2%, and `MultipleFalsePositives`=78.2%. The ledger marks the baseline turn as insufficient because the decision requires interventional evidence. The agent then tests several plausible actions. `ParentalEducationProgram=ComprehensiveSession` improves early detection to 70.8% without increasing multiple false positives, but it is weaker than the gold action. `WholeGenomeSequencing=On` improves early detection to 74.2%, but raises multiple false positives to 84.4%, violating the safety constraint. The pivotal query is `do(ExpandedMetabolicPanel=ExpandedPanel)`: early detection rises to 88.6%, missed

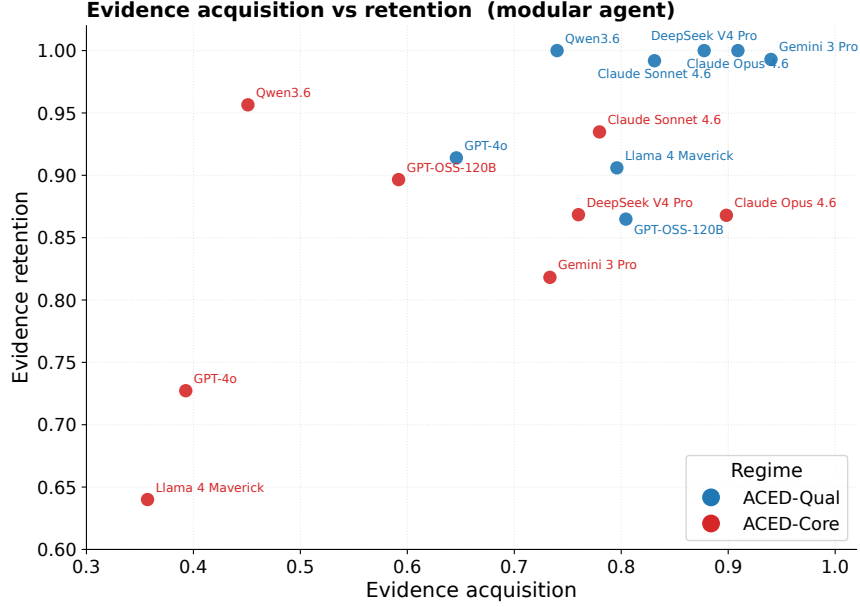


Figure 8: Evidence acquisition versus evidence retention for modular active-code trajectories. The scatter compares ACED-QUAL with ACED-CORE; bubble size indicates ignored evidence. The ledger is a diagnostic trajectory analysis, not a separate benchmark score.

detection falls to 3.8%, and multiple false positives remain stable at 77.9%. The ledger first marks this as weak correct evidence when the decisive data has arrived but the cross-candidate comparison is incomplete, then as strong correct evidence once the model compares target and safety outcomes across interventions. Later queries test specimen-collection timing, parental-education levels, sequencing off, and the standard metabolic panel; these extra turns mainly rule out weaker or unsafe alternatives. The final answer retains the supported recommendation, `ExpandedMetabolicPanel=ExpandedPanel`.

Subgroup-robust deployment. A second Gemini-3-Pro trajectory asks for an intervention that shifts `DailyWearAdherence` toward `Consistent` and away from `Abandoned`, but the intervention must work for both `FitnessTrackerExperience=FirstTime` and `FitnessTrackerExperience=Experienced` users. The gold answer is `AdaptiveMicroReminders=On`. The first turns collect observational data and request interventions, but the ledger marks them as insufficient because the agent has not yet computed subgroup-specific causal contrasts. Once the agent compares interventions by subgroup, the ledger marks strong correct evidence: `AdaptiveMicroReminders=On` raises consistent adherence for both groups, from about 29.4% to 54.8% for first-time users and from about 16.0% to 40.9% for experienced users. The example is useful because the best-looking action is not simply the one with the largest effect in one subgroup. `GamifiedChallenges=Daily` performs especially well for experienced users, but is weaker for first-time users; `SocialLeaderboard=Public` is even harmful for first-time users. The final answer retains the robust recommendation, showing how the ledger can distinguish average improvement from deployable subgroup robustness.

Mediator-structure discovery. A DeepSeek-V4-Pro trajectory asks whether the effect of `OutreachDosage` on `ShootingIncidence` flows entirely through `ConflictMediationEvents`, or whether there is also a pathway that does not pass through that mediator. The gold answer is `also_direct_or_other`. The first two turns request interventional samples but do not yet compare treatment arms or evaluate the mediator-structure estimand, so the ledger marks them as insufficient. After the third interventional query, the agent begins testing residual effects after accounting for `ConflictMediationEvents`; the ledger marks this as weak correct evidence. In the final answer, the agent fits an ordinal model over 1,500 interventional observations from three `OutreachDosage` arms, controlling for the mediator and pre-treatment covariates. The treatment coefficients remain

large and significant after conditioning on the mediator (Standard: $\beta = -0.62$, $p < 0.0001$; Intensive: $\beta = -2.32$, $p < 0.0001$; likelihood-ratio $\chi^2 = 160.35$). The ledger therefore marks the final turn as strong correct evidence for a non-mediated pathway. This case shows a different kind of scientific discovery: the agent is not just choosing an action, but identifying the mechanism by which an intervention affects an outcome.

G Sample-budget ablation

The main benchmark uses a four-turn budget because it stresses planning efficiency: an agent must choose high-value experiments early rather than recover through a long sequence of exploratory queries. We also ran a sample-budget ablation, SAMP5K, that caps each question at 5,000 total sample rows. This ablation changes the scarce resource: an agent may still iterate, but it must allocate a limited number of observational and interventional samples across candidate contrasts.

Changing the constraint also changes which agent designs benefit. Figure 9 shows ACED-CORE accuracy under the sampled-row budget. On paired causal-decision cells, SAMP5K most consistently improves the single-prompt active-code agent: Active+Code improves in 8 of 9 model cells, with a mean gain of 7.3 accuracy points over the four-turn setting. Active reasoning improves in 6 of 9 cells, with a mean gain of 4.2 points, while modular active-code improves in 4 of 9 cells, is unchanged in 2, and drops in 3, for a mean gain of 2.6 points. The largest Active+Code gains occur for Qwen3.6 (+16.7 points), Gemini-3-Pro (+13.3 points), and Claude Opus-4.6 (+10.0 points). SAMP5K therefore complements the four-turn score by separating failures of early experiment selection from failures of sample allocation. We do not use the sampled-row setting to define the ACED-CORE leaderboard.

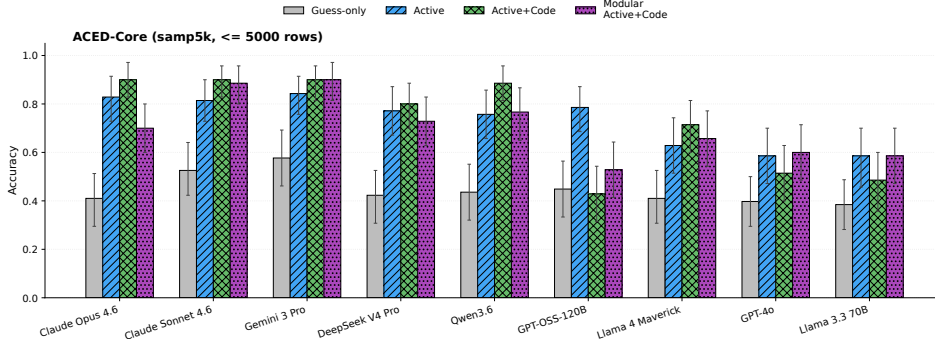


Figure 9: Accuracy on ACED-CORE under the SAMP5K sample-budget ablation. Each task permits at most 5,000 sampled rows, changing the constrained resource from interaction turns to sample allocation.

H Compute resources

All evaluated model calls used hosted APIs; the experiments did not require local training, fine-tuning, model serving, or GPU workers. Local reproduction uses CPU-only Python orchestration for simulation, evaluation, bootstrap resampling, evidence-ledger analysis, and plotting in a standard Python environment. Wall-clock time is dominated by hosted-model latency and rate limits. Provider-side hardware, memory allocation, and per-call execution time are not exposed through standard API access.